ON THE REQUIREMENTS AND ADVANTAGES OF COOPERATIVE COLLISION AVOIDANCE SYSTEMS FOR VULNERABLE ROAD USERS

A dissertation submitted in fulfilment of the requirements for the degree of

Doktor der Naturwissenschaften (Dr. rer. nat.)

to the
Chair for Communication Technology
Faculty of Electrical Engineering and Computer Science
University of Kassel

by

Marek Benjamin Bachmann, Master of Science (M.Sc.)

Date of the disputation: 21. December 2021

Examining committee:

Prof. Dr.-Ing.	Klaus David
Prof. Dr.-Ing.	Delphine Reinhardt
Prof. Dr. habil.	Josef Börcsök
Prof. Dr. rer. nat.	Albert Zündorf

Kassel, October 2021

ABSTRACT

Every year, approximately 350,000 vulnerable road users (VRU), like pedestrians and bicyclists, die due to collisions with vehicles. One approach to reduce this number of fatalities is to use of VRU collision avoidance systems in vehicles. Those car-based VRU collision avoidance systems use sensors mounted in the vehicles. Car-based collision avoidance systems are already helpful to prevent collisions but only work reliable in scenarios where there is a line-of-sight between the vehicle and the VRU. *Cooperative* VRU collision avoidance systems promise to be a solution for the line-of-sight limitation. In cooperative VRU collision systems, the VRU is equipped with a portable device like a smartphone or a smartwatch. This allows to exchange the movement information between the vehicle and the VRU over a wireless network, like WiFi or 5G. Compared to car-based VRU collision avoidance systems, which can determine the position of a VRU relative to the vehicle with millimetre accuracy, e.g. using LIDAR but only in line-of-sight conditions, currently smartphones do not yet achieve a positioning accuracy for detecting collisions reliable. The exact relation between how accurate mobile devices recognize the movement of the VRU and the collision detection performance of a cooperative VRU collision avoidance system is unknown. This dissertation is therefore dedicated to answer the following research question: "What is the minimum required accuracy for VRU movement recognition on mobiles devices for a certain collision detection performance of a cooperative VRU collision avoidance system?" To answer the research question, this dissertation provides the following contributions: First, to precisely detect collisions, an algorithm for precise time-to-collision based collision detection is proposed. Next, for quantifying the collision detection performance, a performance metric for VRU collision avoidance systems is proposed. This performance metric consists of two Key Performance Indicators (KPIs) which represent a system's probability to correctly indicate impending and non impending collisions. Using the two KPIs it is then investigated which accuracy is required for the recognition of VRU's movement, i.e., which sensor accuracy is necessary to achieve specific KPI values of a cooperative VRU collision avoidance system. For the investigation to be representative, it is based on scenarios from the European New Car Assessment Programme (Euro NCAP). Finally, this dissertation provides a simulation based comparison between a car-based VRU collision avoidance system and a cooperative VRU collision avoidance system.

ZUSAMMENFASSUNG

Jedes Jahr sterben etwa 350.000 ungeschützte Verkehrsteilnehmer (engl. vulnerable road users (VRUs)), wie Fußgänger und Fahrradfahrer, durch Kollisionen mit Fahrzeugen. Ein Ansatz zur Verringerung dieser Zahl von Todesopfern ist der Einsatz von Kollisionsvermeidungssystemen für VRUs in Fahrzeugen. Diese fahrzeugbasierten VRU-Kollisionsvermeidungssysteme verwenden in den Fahrzeugen installierte Sensoren. Fahrzeugbasierte Kollisionsvermeidungssysteme sind bereits hilfreich, um Kollisionen zu vermeiden, funktionieren aber nur dann zuverlässig, wenn eine Sichtverbindung zwischen dem Fahrzeug und dem VRU besteht. *Kooperative* VRU-Kollisionsvermeidungssysteme können eine Lösung für diese Einschränkung zu sein. Bei kooperativen VRU-Kollisionsschutzsystemen ist der VRU mit einem tragbaren Gerät wie einem Smartphone ausgestattet. Dies ermöglicht den Austausch von Bewegungsinformationen zwischen dem Fahrzeug und dem VRU über eine Funkverbindung wie WLAN oder 5G. Im Vergleich zu fahrzeugbasierten VRU-Kollisionsvermeidungssystemen, haben Smartphones derzeit noch keine ausreichende Positionierungsgenauigkeit um Kollisionen zuverlässig zu erkennen. Der genaue Zusammenhang zwischen der Genauigkeit, mit der mobile Geräte die Bewegung der VRUs erkennen, und der Kollisionserkennungsleistung eines kooperativen VRU-Kollisionsvermeidungssystems ist unbekannt. Diese Dissertation widmet sich daher der folgenden Forschungsfrage: "Was ist die erforderliche Mindestgenauigkeit der VRU-Bewegungserkennung auf mobilen Geräten für eine bestimmte Kollisionserkennungsleistung eines kooperativen VRU-Kollisionsvermeidungssystemes?" Zur Beantwortung der Forschungsfrage liefert diese Dissertation die folgenden Beiträge: Erstens wird zur Erkennung von Kollisionen ein Algorithmus zur präzisen “time-to-collision” basierten Kollisionserkennung vorgeschlagen. Anschließend wird zur Quantifizierung der Kollisionserkennungsleistung eine Leistungskennzahl für VRU-Kollisionsvermeidungssysteme eingeführt. Diese Leistungsmetrik besteht aus zwei Key Performance Indicators (KPIs), die die Wahrscheinlichkeit eines Systems quantifizieren, drohende und nicht drohende Kollisionen korrekt anzuzeigen. Unter Verwendung der beiden KPIs wird dann untersucht, welche Genauigkeit für die Erkennung der Bewegung von VRUs erforderlich ist, um einen bestimmten KPI-Wert zu erreichen. Damit die Untersuchung repräsentativ ist, stützt sie sich auf Szenarien aus dem European New Car Assessment Programme (Euro NCAP). Schließlich bietet diese Arbeit einen simulationsbasierten Vergleich zwischen einem fahrzeugbasierten und einem kooperativen VRU-Kollisionsvermeidungssystem.

CONTENTS

1 INTRODUCTION

According to the Global Status Report on Road Safety 2018 from the World Health Organisation, in 2016, 351.000 Vulnerable Road Users (VRUs), like pedestrians and bicyclists, were killed due to collisions with vehicles [1.1]. Compared to the 325,000 VRU fatalities in 2013, reported in the Global Status Report on Road Safety 2015, this is an increase of 26,000 fatalities (8.2 %) [1.2].

To improve the safety of VRUs, more and more vehicles are equipped with a VRU Collision Avoidance System (VRU CAS). A VRU CAS recognizes VRUs and their movement and intervenes when there is a risk of a collision.

The fact that in 2015 the European New Car Assessment Programme (Euro NCAP) had introduced the first version of the obligatory "Test Protocol Autonomous Emergency Braking (AEB) VRU systems"[1.3], shows that there is an urgent need and a significant interest for better VRU protection. This assessment protocol influences the overall safety rating of a car in respect to "how well vehicles autonomously detect and prevent collisions with pedestrians"[1].

Current VRU CASes available rely on sensor systems installed in the vehicles. Such *car-based* VRU CASes, as they will be called in the scope of this dissertation, use Radio detection and ranging (Radar), infrared, camera, or Light Detection and Ranging (LIDAR) systems to detect a VRU [1.4]. Car-based VRU CASes first have to detect a VRU in the environment and then track its movement to calculate a collision risk between the car and the VRU.

Even tough car-based VRU CASes are already helpful to prevent collisions, or at least reduce the severity of injuries, such systems have considerable limitations. First, since the sensor systems are based on the *reflection* of electric-magnetic waves at the VRU, those systems require a Line-of-Sight (LOS) and therefore cannot address all Euro NCAP scenarios [P1]. Because of the LOS restriction, car-based VRU CASes might fail to detect a VRU early enough in situations in which the VRU is covered by an obstacle, like a parked car. Second, for car-based VRU CASes, it is hard to infer any additional information about VRUs, like being a pedestrian or a bicyclist, movement dynamics or other contextual information, like their current level of distraction. Therefore car-based VRU CASes are also limited in respect to predict the VRUs movement reliably.

1 Euro NCAP. (Nov. 2015). 'Euro NCAP Puts Autonomous Pedestrian Detection to the Test' [Online]. https://www.euroncap.com/de/presse/pressemitteilungen/euro-ncap-puts-autonomous-pedestrian-detection-to-the-test/, accessed 13.03.2021

For accident hot spots, like busy intersections, there are some ongoing field tests to improve VRU safety by adding additional surveillance systems to the infrastructure. The drawback of this approach is that it is expensive and that it does not scale well.

1.1 OPTIMAL VRU COLLISION AVOIDANCE SYSTEM?

A promising enhancement for car-based VRU CASes are *cooperative* VRU Collision Avoidance Systems (cooperative VRU CASes). The idea of a cooperative VRU CAS is to include VRUs actively in road safety by using information about the VRU from the VRU's mobile devices, especially - but not limited to - the VRU's movement information. The term cooperative VRU CAS is - not yet - a standard term, nevertheless, within the scope of this dissertation the term *cooperative* Collision Avoidance System (CAS) is used for any system that actively include information from the VRU side, because vehicles and VRUs *cooperate to avoid collisions* by providing their information to the CAS.

So far, several publications suggest different architectures for a cooperative VRU CAS in which VRUs exchange movement information via radio with vehicles to prevent collisions [P1], [1.5]–[1.16], [P2]–[P5]. Besides some specific systems [1.17]–[1.20] that use special hardware tags on the VRU side, most approaches propose to use smartphones on the VRU side.

A cooperative VRU CAS could address most limitations of the existing car-based VRU CASes. Note that cooperative VRU CASes are still under research and no commercial solutions are available yet. A cooperative VRU CAS basically requires three components, see Fig. 1.1:

1. At the VRU: A mobile device (e.g. a smartphone) for movement recognition and transmission of movement and warning data (1),
2. In the vehicle: A smartphone and/or an on-board unit also used for movement recognition and transmission of movement and warning data (2),
3. A wireless network connection between VRUs and vehicles. The communication can be direct (3a), e.g. via Wireless Local Area Network (WLAN) or Device-to-Device (D2D) in 5G, or cellular-based (3b) with an optional server (e.g., a Multi-Access Edge Computing server).

Thus, algorithms for detecting collision can be run directly on the devices, when direct or cellular communication is used, or at a server when cellular communication is used. Also a combination of direct and cellular communication is possible allowing to dynamically switch between the server or the devices for collision detection calculation, e.g.

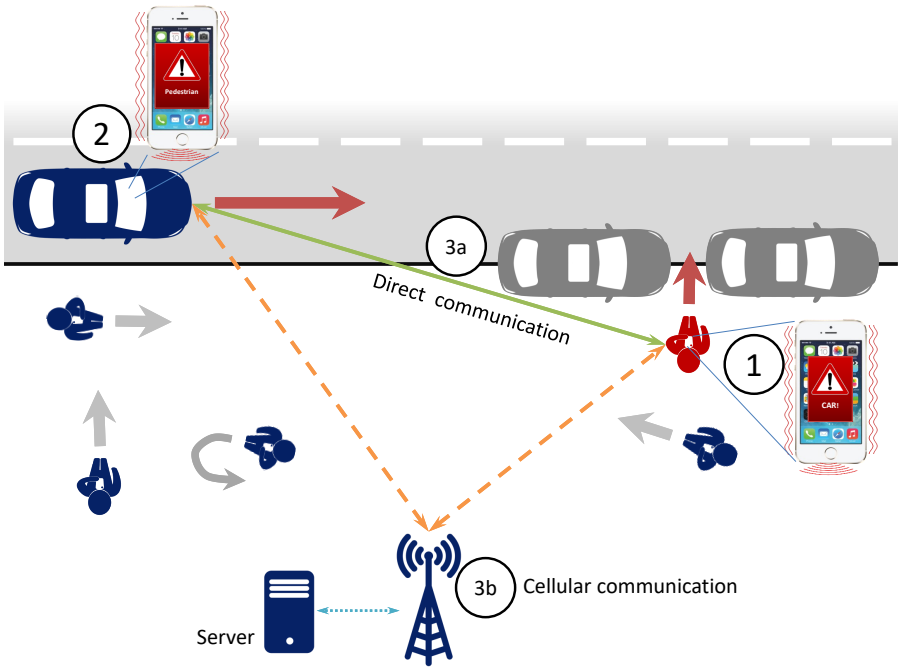


Fig. 1.1: Concept sketch of an *cooperative* VRU Collision Avoidance System [1.12]. In the sketch a pedestrian (1) and a car (2) exchange their movement information either via a direct communication link (3a) or via cellular communication (3b).

use the server to pre-filter potential collisions in crowded situations and the devices for real time precise collision detection.

Since VRUs and vehicles can transmit their movement vectors via a wireless network connection, cooperative VRU CASes are independent of the LOS condition; a key limitation of the aforementioned car-based VRU CASes. Another major unprecedented advantage of a cooperative VRU CAS is that not only the driver, but also the VRU can be warned. Furthermore, a cooperative VRU CAS does not only allow the exchange of movement information between a VRU and the vehicle but also the transmission of any other vital information, like being a pedestrian or bicyclist, which helps to determine the risk of a collision more precisely.

Essentially, mobile devices, like smartphones or smartwatches, provide everything that is necessary for a cooperative VRU CAS: First, to determine the movement of a VRU, Global Navigation Satellite Systems (GNSS), digital compass information, and Inertial Measurement Units (IMUs), i.e. an accelerometer and a gyroscope, can be used. Second, to exchange movement information with vehicles, direct and cellular radio access methods are available. The technical possibilities and the fact that a rising number of people permanently carry smartphones with them, makes such devices a promising platform for cooperative VRU CASes.

Moreover, collision detection algorithms can easily be deployed on smartphones at large-scale and at lower individual costs than infrastructure-based solutions, thus enable a fast penetration of a cooperative VRU CAS in vehicles and for VRUs.

1.2 PROBLEM STATEMENT

A VRU CAS needs especially to work reliable in urban scenarios since, e.g. in 2019 in Germany, 94.6% of accidents between vehicles and pedestrians occurred in urban areas[2]. The challenge for a VRU CAS in an urban area is that different road users, like vehicles and pedestrians, frequently move past each at small distances. Accordingly, the margin between a correct and false collision detection is small. Additionally, in urban areas Non-Line-of-Sight (NLOS) scenarios are frequent. Therefore, a VRU CAS needs not only to detect all impending collisions even in NLOS scenarios, but also has to avoid false collision detections even when road users move past each other at a small distance. Avoiding frequent false collision detections is crucial since otherwise the system will not be accepted by the user. The ability of a VRU CAS to reliable detect impending collisions and at the same time avoid false detections is called *collision detection performance* within the scope of this dissertation. Thus, a VRU CAS which frequently does not detect impending collisions or falsely detect a collision when no collision is imminent has a low collision detection performance. On the contrary, a VRU CAS which detects most of impending collisions and also rarely falsely detect collisions is considered to have a high collision detection performance.

To achieve a high collision detection performance, the accuracy with which the movement of a VRU can be recognized is crucial. Compared to car-based VRU CASes, which can determine the position of a VRU relative to the vehicle with millimetre accuracy, e.g. using LIDAR, but only in LOS conditions, currently smartphones do not yet achieve a GNSS positioning accuracy for detecting collisions reliable [P5]. But there there is a continuous improvement in smartphone GNSS positioning accuracy, so that it can be expected that the accuracy will be much higher in the near future [1.21]. Nevertheless. the collision detection performance does not solely depend on position accuracy. A VRU CAS needs to have the current movement information consisting of, at least, its current position, speed, and direction. As it will be detailed in the following Chapter 2 the relation between how accurate mobile devices recognize the movement of the VRU and the collision detection performance of a cooperative VRU CAS is unknown. This

2 Statistisches Bundesamt Destasis. (Jul. 2020). 'Verkehrsunfaelle 2019.' [Online]. Available: https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Verkehrsunfaelle/Publikationen/Downloads-Verkehrsunfaelle/verkehrsunfaelle-jahr-2080700197004.pdf, accessed 07.12.2020

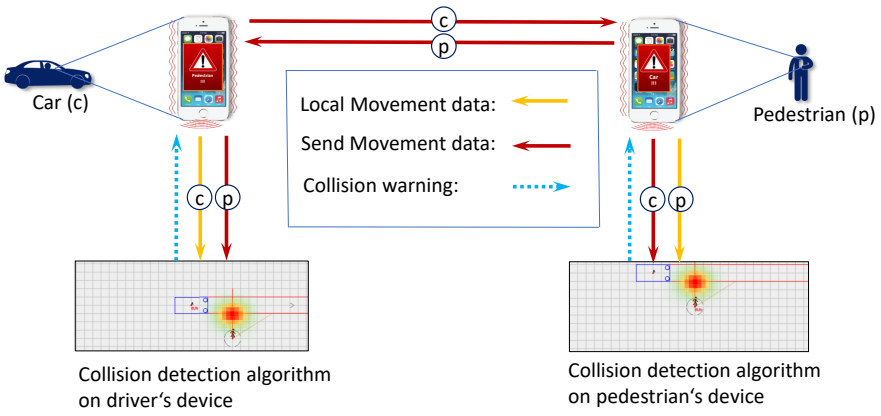


Fig. 1.2: Peer-to-Peer (P2P) architecture

dissertation is therefore dedicated to answer the following research question:

> What is the minimum required accuracy for VRU movement recognition on mobiles devices for a certain collision detection performance of a cooperative VRU CAS?

To answer this research questions two challenges need to be solved:

CHALLENGE I (C I): How to precisely detect collisions in a cooperative VRU CAS?

CHALLENGE II (C II): How to quantify the collisions detection performance of a VRU CAS independent to a certain architecture?

The first challenge (C I) arises from the problem statement that in urban areas the margin between a correct and false collision detection is small. Therefore, it is crucial to use collision detection algorithms that are precise, i.e. take into account the "geometry" of the road users: Considering the current state-of-the-art regarding collision detection algorithms the realistic representation of a VRU has been neglected. For the majority of the proposed cooperative VRU CASes the authors use a point representation for VRUs, others just use collisions detection algorithms dedicated for car-to-car collision detection based on rectangle geometries. Using a non appropriate representation for the road users in collision detection algorithms also may cause false collision detection. Thus, to answer the research question a precise collision detection algorithm is necessary, which would achieve a high collision detection performance provided that no errors in recognizing the VRU movement occur.

The second challenge (C II) is due to the fact that, besides a first qualitative investigation conducted in [1.12], there exists no method to systematically investigate possible architectures of cooperative VRU CASes in terms of collision detection performance. Even though the basic concept of a cooperative VRU CAS is quite simple, there are many

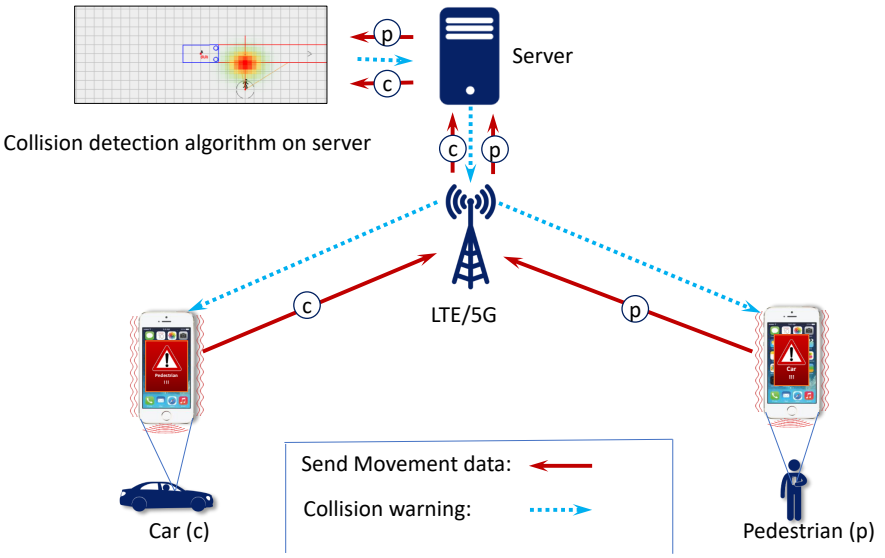


Fig. 1.3: Client/server architecture

different ways to implement a cooperative VRU CAS. For example, two vastly different approaches are: P2P architectures with D2D [1.9], [1.14], [1.22] communication and local processing, see Fig. 1.2, or a client/server architecture [1.5], [1.23]–[1.25] with collision detection running on the server, see Fig. 1.3. In addition also architectures based on hybrid communication with processing distributed over the vehicles, the VRUs, and a server were proposed [1.26]. Note that those different approaches have been proposed but had only been evaluated on an individual basis. The key difference between these approaches is where the collision detection is performed. Most approaches use a server located either close to the roadside [1.23], [1.24], [1.26] or further apart in a data centre [1.24], [1.25], while some use the mobile device itself [1.14], [1.22]. Thus, to get the minimum required movement recognition accuracy for an arbitrary cooperative VRU CAS a general indicator is is needed that maps the influence of movement recognition errors on the collision detection performance.

1.3 CONTRIBUTION

To answer the research question "What is the minimum required accuracy for VRU movement recognition on mobiles devices for a certain collision detection performance of a cooperative VRU CAS?", this dissertation provides the following contributions: First, in accordance to C I, to precisely detect collisions, an algorithm for precise Time to Collision (TTC) based collision detection is proposed. The algorithm models real world scenarios in a two-dimensional metric coordinate system and supports point, rectangle, and circle geometries to represent the road users. Within the coordinate system the algorithm is able

to determine both, the TTC and the impact position at the road users' geometries.

Next, for quantifying the collision detection performance, in accordance to C II, a performance metric for VRU CASes is proposed. This performance metric consists of two Key Performance Indicators (KPIs) which represent a system's probability to correctly indicate impending and non impending collisions. To precisely determine the KPIs, an extension of the precise TTC based collision detection algorithm is used. The idea of the adaptation is to equate the road users' movement equations to the direction instead of time. In this way it is possible to determine "direction ranges" for which collisions would occur. This is particular helpful for calculating the KPIs since this approach reduces the complexity for calculation and increases the precision because it allows to consider a continuous range of directions instead iterating over discrete direction values. Even though it is not in scope of the dissertation the adapted collision detection algorithms for "direction ranges" can also be used as an probabilistic collision detection approach instead of TTC based collision detection.

Using the two KPIs it is then investigated which accuracy is required for the recognition of VRU's movement, i.e., which accuracy in terms of position, speed, and direction is necessary to achieve a specific KPI value of a cooperative VRU CAS. For this investigation to be representative, it is based on scenarios from the Euro NCAP.

Finally, this dissertation provides a simulation based comparison between a car-camera based VRU CAS and a cooperative VRU CAS. This comparison is intended to answer the legitimate question: Even if a cooperative VRU CAS is feasible using mobile devices on the VRU side: Can a cooperative VRU CAS provide a significant higher collision detection performance in relevant collision scenarios compared to car-based VRU CAS? For this, the two KPIs are simulative compared for a cooperative VRU CAS and car-based VRU CAS in the Euro NCAP NLOS collision scenario. The simulations are conducted for two VRU speeds, first a walking pedestrian and second a running pedestrian. Besides the determination of the twoKPIs values within the time domain of the Euro NCAP NLOS collision scenario, it is analysed when the VRU is in the Field of View (FOV) of the car-based VRU CAS, and based on this time, if a collision can still be prevented according to the Latest Time to Avoid Collision (LTAC).

1.4 KEY ASSUMPTIONS

Within the whole scope of the dissertation a simplified cooperative VRU CAS system will be used. The simplified cooperative VRU CAS detects collision by extrapolation of the linear movement of two road users and determines if there is a tangential point for the road user's geometries. If there is a tangential point an impending collision is

detected, otherwise no impending collision is detected. The collision detection is based on the following four key assumptions:

1. Collision detection is based on linear extrapolation of the non-accelerated movements of a VRU and a vehicle
2. Precise binary collision detection, i.e. tangential point detection between the road users' geometries for any time in the future
3. Error free movement recognition for the vehicle
4. No communication delay.

The reasons for these assumptions for finding the *minimum* required movement accuracy on the mobile device of a VRU in a cooperative VRU CAS are: Using linear movement extrapolation (1.) simplifies the prediction of the VRU movement by assuming no change of the VRU's current movement. Using more complex predictions of the road user's movement would increase the required movement accuracy. Using tangential point detection (2.) between the road users' geometries guarantees that a collision is reliable detected as long as the assumptions of non-accelerated linear movement for both road users holds and the geometries are an appropriate representation of the real vehicle and VRU. The errors in recognizing the movement of the vehicle (3.) are neglected since it can be assumed, without loss of any generality, that the errors in recognizing the movement of the VRU are relative to the vehicles frame of reference. Communication delays are neglected (4.) since, in terms of the accuracy for movement recognition, a communication delay would be an error source independent to the sensor accuracies on the mobile device. In addition, communication delays can be corrected in case of linear movement extrapolation using synchronised sender timestamps.

1.5 OUTLINE

This dissertation consists of 9 chapters. In the following Chapter 2, an overview of the state of the research regarding VRU CASes and the state-of-the-art regarding collision detection algorithms is given.

In Chapter 3 an algorithm for precise collision detection is presented. The precise collision detection algorithm is required to determine the performance metric introduced in the succeeding Chapter 4. Since the precise collision detection algorithm supports the representation of the VRU as a circle or point geometry it can also be used as an collision detection algorithm which represents VRUs more realistic in comparison the the state-of-the-art algorithms. Afterwards in Chapter 4 the general performance metric for VRU CAS is introduced and detailed. The formal definition of the influence of movement recognition accuracy on the general performance metric is then provided in Section 4.5.

An algorithm for calculating the performance metric regrading the movement recognition accuracy is then presented in Chapter 5. It is an adaption of the algorithm for precise collision detection from Chapter 3. The idea for this algorithm is to equate linear movement not by time but by directions which reduces the complexity for calculating the performance metric.

In order to answer the research question for most of the statistical relevant collision scenarios between pedestrians and vehicles, in Chapter 6 an investigation is conducted to find the smallest number of scenarios which, by slightly adjustment of the scenario parameters, cover most of the statistical relevant situations between pedestrians and vehicles. Thereafter, the required accuracies for VRU movement recognition in cooperative VRU CAS regarding the performance metric from Section 4.5 are investigated for the identified scenarios using the algorithm from Chapter 5. The results for the minimum required movement recognition accuracy for a certain collision detection performance are then used in Chapter 7 for a direct comparison between a car-based VRU CAS and a cooperative VRU CAS . Finally, in Chapter 8 the dissertation is summarized and the conclusion is given.

1.6 OWN PUBLICATIONS

Parts of the work conducted within the scope of this dissertation have been published in the Institute of Electrical and Electronics Engineers (IEEE) Transactions on Intelligent Transportation Systems (T-ITS) journal, or were published in the proceedings of conferences. These publications are as follows:

[P1] M. Bachmann, M. Morold, S. Engel, J. Götz and K. David, 'Camera vs. Cooperative VRU Collision Avoidance', in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, Antwerp, Belgium, May 2020, pp. 1–5. DOI: 10.1109/VTC2020-Spring48590.2020.9128854 (cit. on pp. 1, 2).

[P2] M. Bachmann, M. Morold and K. David, 'Improving Smartphone Based Collision Avoidance by Using Pedestrian Context Information', in *2017 IEEE International Conference on Pervasive Computing and Communications Work in Progress (PerCom WiP 2017)*, Kona, USA, Mar. 2017, pp. 2–5. DOI: 10.1109/PERCOMW.2017.7917507 (cit. on p. 2).

[P3] M. Bachmann, M. Morold and K. David, 'On the Required Movement Recognition Accuracy in Cooperative VRU Collision Avoidance Systems', *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1708–1717, Mar. 2021. DOI: 10.1109/TITS.2020.2976593 (cit. on p. 2).

[P4] M. Morold, M. Bachmann, L. Mathuseck and K. David, 'Automated Learning of Pedestrian Walking Speed Profiles for Improved Movement Prediction', in *INFORMATIK 2019: 50 Jahre Gesellschaft für Informatik – Informatik für Gesellschaft (Workshop-Beiträge)*, C. Draude, M. Lange and B. Sick, Eds., Kassel, Germany: Gesellschaft für Informatik e.V., Sep. 2019, pp. 209–218. DOI: 10.18420/inf2019_ws24 (cit. on p. 2).

[P5] M. Bachmann, M. Morold, K. David and P. Henkel, 'The Wireless Seat Belt Requirements, Experiments, and Solutions for Pedestrian Safety', in *14th Workshop on Context and Activity Modeling and Recognition*, Athens, Greece, Mar. 2018, pp. 250–255. DOI: 10.1109/PERCOMW.2018.8480123 (cit. on pp. 2, 4).

REFERENCES

[1.1] World Health Organization, *Global status report on road safety 2018*, World Health Organization, Ed., accessed 14.12.2020, Geneva, Switzerland, 2018. [Online]. Available: https://apps.who.int/iris/rest/bitstreams/1164010/retrieve (cit. on p. 1).

[1.2] World Health Organization, *Global status report on road safety 2015*, accessed 14.12.2020, Geneva, Switzerland, 2015. [Online]. Available: http://apps.who.int/iris/bitstream/10665/189242/1/9789241565066_eng.pdf (cit. on p. 1).

[1.3] European New Car Assessment Programme (Euro NCAP). 'Test Protocol - AEB VRU systems (Version 1.0.1)'. checked 14.12.2020. (Jun. 2015), [Online]. Available: https://cdn.euroncap.com/media/21509/euro-ncap-aeb-vru-test-protocol-v101.pdf (cit. on p. 1).

[1.4] T. Gandhi and M. M. Trivedi, 'Pedestrian Protection Systems: Issues, Survey, and Challenges', *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 413–430, Sep. 2007. DOI: 10.1109/TITS.2007.903444 (cit. on p. 1).

[1.5] C. Sugimoto, Y. Nakamura and T. Hashimoto, 'Prototype of pedestrian-to-vehicle communication system for the prevention of pedestrian accidents using both 3G wireless and WLAN communication', in *Proc. 3rd International Symposium on Wireless Pervasive Computing (ISWPC)*, Santorini, Greece, May 2008, pp. 764–767. DOI: 10.1109/ISWPC.2008.4556313 (cit. on pp. 2, 6).

[1.6] C. Sugimoto, Y. Nakamura and T. Hashimoto, 'Development of Pedestrian-to-Vehicle Communication System Prototype for Pedestrian Safety Using both Wide-Area and Direct Communication', in *Proc. 22nd International Conference on Advanced*

Information Networking and Applications (AINA), Ginowan, Japan, Mar. 2008, pp. 64–69. DOI: 10.1109/AINA.2008.126 (cit. on p. 2).

[1.7] C. Sugimoto and Y. Nakamura, 'Provision of information support by pedestrian-to-vehicle communication system', in *Proc. 8th International Conference on ITS Telecommunications*, Phuket, Thailand, Oct. 2008, pp. 160–163. DOI: 10.1109/ITST.2008.4740248 (cit. on p. 2).

[1.8] X. Wu, R. Miucic, S. Yang, S. Al-Stouhi, J. Misener, S. Bai and W.-h. Chan, 'Cars Talk to Phones: A DSRC Based Vehicle-Pedestrian Safety System', in *Proc. 80th Vehicular Technology Conference (VTC2014-Fall)*, Vancouver, Canada, Sep. 2014, pp. 1–7. DOI: 10.1109/VTCFall.2014.6965898 (cit. on p. 2).

[1.9] K. Dhondge, S. Song, B.-Y. Choi and H. Park, 'WiFiHonk: Smartphone-Based Beacon Stuffed WiFi Car2X-Communication System for Vulnerable Road User Safety', in *Proc. Vehicular Technology Conference (VTC 2014-Spring)*, Seoul, South Korea, May 2014, pp. 1–5. DOI: 10.1109/VTCSpring.2014.7023146 (cit. on pp. 2, 6).

[1.10] S. Engel, C. Kratzsch and K. David, 'Car2Pedestrian–Communication: Protection of Vulnerable Road Users Using Smartphones', in *Proc. Advanced Microsystems for Automotive Applications 2013*, Heidelberg, 2013, pp. 31–41. DOI: 10.1007/978-3-319-00476-1_4 (cit. on p. 2).

[1.11] M. Liebner, F. Klanner and C. Stiller, 'Active safety for vulnerable road users based on smartphone position data', in *Proc. Intelligent Vehicles Symp. (IV)*, Gold Coast City, Australia, Jun. 2013, pp. 256–261. DOI: 10.1109/IVS.2013.6629479 (cit. on p. 2).

[1.12] K. David and A. Flach, 'CAR-2-X and Pedestrian Safety: Innovative Collision Avoidance System', *IEEE Vehicular Technology Magazine*, vol. 5, no. 1, pp. 70–76, 2010. DOI: 10.1109/MVT.2009.935536 (cit. on pp. 2, 3, 5).

[1.13] A. Flach and K. David, 'Car and Pedestrian Collisions: Causes and Avoidance Techniques', in *Wireless Vehicular Networks for Car Collision Avoidance*, R. Naja, Ed., New York: Springer, 2013, pp. 229–260. DOI: 10.1007/978-1-4419-9563-6_8 (cit. on p. 2).

[1.14] J. J. Anaya, P. Merdrignac, O. Shagdar, F. Nashashibi and J. E. Naranjo, 'Vehicle to pedestrian communications for protection of vulnerable road users', in *Proc. Intelligent Vehicles Symp. (IV)*, Dearborn, USA, Jun. 2014, pp. 1037–1042. DOI: 10.1109/IVS.2014.6856553 (cit. on pp. 2, 6).

[1.15] M. Bagheri, M. Siekkinen and J. K. Nurminen, 'Cellular-based vehicle to pedestrian (V2P) adaptive communication for collision avoidance', in *Proc. 3rd International Conference on Connected Vehicles and Expo (ICCVE)*, Vienna, Austria, Sep. 2014, pp. 450–456. DOI: 10.1109/ICCVE.2014.7297588 (cit. on p. 2).

[1.16] A. Tahmasbi-Sarvestani, H. Nourkhiz Mahjoub, Y. P. Fallah, E. Moradi-Pari and O. Abuchaar, 'Implementation and Evaluation of a Cooperative Vehicle-to-Pedestrian Safety Application', *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 4, pp. 62–75, Oct. 2017. DOI: 10.1109/MITS.2017.2743201 (cit. on p. 2).

[1.17] D. Westhofen, C. Grundler, K. Doll, U. Brunsmann and S. Zecha, 'Transponder- and Camera-based advanced driver assistance system', in *Proc. Intelligent Vehicles Symp. (IV)*, Madrid, Spain, Jun. 2012, pp. 293–298. DOI: 10.1109/IVS.2012.6232140 (cit. on p. 2).

[1.18] B. Schaffer, G. Kalverkamp, M. Chaabane and E. M. Biebl, 'A cooperative transponder system for improved traffic safety, localizing road users in the 5 GHz band', *Advances in Radio Science*, vol. 10, pp. 39–44, 2012. DOI: 10.5194/ars-10-39-2012 (cit. on p. 2).

[1.19] R. Raßhofer, D. Schwarz, C. Morhart and E. M. Biebl, 'Cooperative Sensor Technology for Preventive Vulnerable Road User Protection', in *Proc. 21st International Technical Conference of the Enhanced Savety of Vehicles*, Stuttgart, Germany, Jun. 2009, pp. 1–6 (cit. on p. 2).

[1.20] C. Morhart, E. Biebl, D. Schwarz and R. Rasshofer, 'Cooperative Multi-User Detection and Localization for Pedestrian Protection', in *Proc. German Microwave Conf. (GeMIC 2009)*, Munich, Germany, 2009, pp. 1–5. DOI: 10.1109/GEMIC.2009.4815863 (cit. on p. 2).

[1.21] Samuel K. Moore, 'Super-accurate GPS coming to smartphones in 2018', *IEEE Spectrum*, vol. 54, no. 11, pp. 10–11, 2017 (cit. on p. 4).

[1.22] R. Miller and Q. Huang, 'An adaptive peer-to-peer collision warning system', in *IEEE 55th Vehicular Technology Conference*, Birmingham, USA, May 2002, pp. 317–321. DOI: 10.1109/VTC.2002.1002718 (cit. on p. 6).

[1.23] M. Malinverno, G. Avino, C. Casetti, C. F. Chiasserini, F. Malandrino and S. Scarpina, 'Edge-Based Collision Avoidance for Vehicles and Vulnerable Users: An Architecture Based on MEC', *IEEE Vehicular Technology Magazine*, vol. 15, no. 1, pp. 27–35, 2020. DOI: 10.1109/MVT.2019.2953770 (cit. on p. 6).

[1.24] A. Napolitano, G. Cecchetti, F. Giannone, A. L. Ruscelli, F. Civerchia, K. Kondepu, L. Valcarenghi and P. Castoldi, 'Implementation of a MEC-based Vulnerable Road User Warning System', in *2019 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE)*, Turin, Itlay, Jul. 2019, pp. 1–6. DOI: 10.23919/EETA.2019.8804497 (cit. on p. 6).

[1.25] M. Bagheri, M. Siekkinen and J. K. Nurminen, 'Cloud-Based Pedestrian Road-Safety with Situation-Adaptive Energy–Efficient Communication', *IEEE Intelligent Transportation Systems Magazine*, vol. 8, no. 3, pp. 45–62, 2016. DOI: 10.1109/MITS.2016.2573338 (cit. on p. 6).

[1.26] H. Artail, K. Khalifeh and M. Yahfoufi, 'Avoiding car-pedestrian collisions using a VANET to cellular communication framework', in *13th International Wireless Communications and Mobile Computing Conference*, Rome, Italy, Jun. 2017, pp. 458–465. DOI: 10.1109/IWCMC.2017.7986329 (cit. on p. 6).

2

STATE-OF-THE-ART VRU COLLISION AVOIDANCE SYSTEMS

Protecting Vulnerable Road Users (VRUs) from accidents with vehicles can be categorized into two domains: First, passive VRU safety, which aim at lower the severity of injuries caused by an impact with a vehicle. Passive VRU safety is especially achieved by special front designs of the vehicles, often with absorbing chassis components, to lower the impact forces on crucial body parts [2.1]. The second domain, active VRU safety aims at preventing collisions with a VRU. Within the domain of active VRU safety, current products make use of radar, infrared, camera, or Light Detection and Ranging (LIDAR) systems and are often referred to as car-based VRU Collision Avoidance System (VRU CAS). Car-based systems have considerable limitations like very limited knowledge about the VRU and depending on Line-of-Sight (LOS) conditions. Car-based VRU CASes have been comprehensively surveyed in [2.1] and therefore will not be discussed in detail in this dissertation. To overcome the limitations of car-based VRU CASes, in recent years, considerable attention has been paid to develop *cooperative* VRU Collision Avoidance Systems (cooperative VRU CASes) based on Vehicle-to-everything (V2X) communication[2.2]–[2.9]. cooperative VRU CASes require that VRUs are equipped with some mobile device for recognizing a VRU's movements and exchanging information with vehicles. Besides some research on cooperative VRU CAS that use specialized hardware tags instead as a mobile device [2.10], [2.11], the majority of publications suggest the use of smartphones at the VRU side.

Considering the current state-of-the-research concerning cooperative VRU CAS one finds that most authors focussed on single aspects of cooperative VRU CASes, like communication, collision detection algorithms, and different implementation architectures. This chapter summarizes the current state-of-the-research within the next four sections in "General SOTA", "Communication", "System Architectures", and "Collision Detection Algorithms".

2.1 GENERAL

Some of the earliest publications that consider the usage of smartphones for VRU CAS were published by Sugimoto et al. [2.5]–[2.7] in 2008. In these publications, the authors proposed a pedestrian Collision Avoidance System (CAS) in which a collision risk is estimated by checking if the "vehicle's risk area" intersects with the "risk

area of the pedestrian" and by considering the remaining time to collision.

Flach and David [2.12], [2.13] also proposed a smartphone-based collision avoidance system in 2009. For the proposed system, they investigated how much time the system has available to avoid a collision considering the transmission time of data, calculations and warnings for different radio-based communication architectures. Since 2009 several other smartphone-based collision avoidance systems have been proposed [2.3], [2.4], [2.14], [2.15].

2.1.1 *GNSS Accuracy*

The ability to detect whether a collision between a VRU and the vehicle is possible also depends on the accuracy of movement information. Liebner et al. [2.16] investigated the feasibility of Global Navigation Satellite Systems (GNSS) in smartphones for the application in a bicycle warning system. The authors analysed the GNSS accuracy of a common smartphone in 2017 which was attached to a bicycle. They concluded that the lateral GNSS position accuracy was too low for lane level localization while the longitudinal accuracy was sufficient for the application in their bicycle warning system. Similarly, Theman et al. [2.9] also investigated the influence of position accuracy on "collision risk determination" between a VRU and vehicles for developing approaches for evasive and breaking manoeuvrers of vehicles. The authors conclude that the standard deviation of the predicted VRU position error must not exceed 0.55 m in order to maintain a false activation rate of an avoidance manoeuvrer under 10%. Rünz et al. [2.17] proposed a smartphone-based dead reckoning localization solution. The authors showed that in order to detect impending collisions between VRUs and vehicles with a "true-positive event" probability of over 50%, the standard deviation for position error has to be smaller than 0.4 m. Moreover, the authors proposed a method to improve positional accuracy by using smartphone sensors for dead reckoning localization.

2.1.2 *Context Information*

In [2.18], it was investigated how to reduce the probability of a Missed Alarm by detecting when a pedestrian steps over a curb. In [2.19], it was additionally shown that increasing the position accuracy using a GNSS/Inertial Navigation System (INS) solution helps to reduce the Missed Alarm and False Alarm rate. Bieshaar et al. [2.20] used the detection of cyclist starting movement (waiting / starting) to select an appropriate weighting between two forecasting models in order to obtain an improved movement position prediction accuracy.

2.2 COMMUNICATION

Considering the current state of the research most attention was paid to the networking issue. Networking can be direct, e.g., via Device-to-Device (D2D) in 5G, via cellular links, or using a hybrid combination of both, D2D and cellular.

2.2.1 *Direct (WLAN / DSRC / Bluetooth)*

Dedicated Short Range Communication (DSRC) is a set of protocols and standards designated for Vehicle-to-Vehicle (V2V) communication based on Wireless Local Area Network (WLAN). In detail, DSRC uses the Institute of Electrical and Electronics Engineers (IEEE) 802.11p WLAN standard for short to medium range communication between vehicles which is an extension to the existing IEEE-802.11 consumer WLAN standards. Since DSRC is designated for the use in vehicles, consumer devices like smartphones do not support DSRC communication out of the box.

Wu et al. [2.4] developed a DSRC based VRU protection prototype for smartphones by implementing appropriate software drivers to enable an Android based smartphone to use DSRC communication. The authors evaluate their DSRC prototype regarding Inter Packet Gap (IPG) and Received Signal Strength Indicator (RSSI). In Dhondge et al. [2.3] consumer WLAN was used for communication in a VRU protection prototype. They evaluated the number of packages received depending on distance and speed, the VRU's Time Available to Stop (TAS) after a received warning depending on vehicle speed, and the resulting collision risk based on TAS and the vehicle speed. Another consumer WLAN based VRU protection prototype was evaluated by Anaya et al. [2.8]. In this publication the authors examined the probability that a message is received as well as the Packet Delivery Ratio (PDR), both depending on distance. In addition the probability of being informed depending on Packet Inter-Reception Time (PIR)was evaluated. The use of a combination between DSRC and Bluetooth has been proposed by Miller et al. [2.21].

2.2.2 *Cellular (2-5G)*

An alternative to direct WLAN/DSRC based communication is the use of cellular networks. Bagheri [2.2] developed a client/server VRU protection prototype based on smartphones which transmits the information over the cellular network with an adaptive transmission rate. Pedestrians use their smartphone to run a mobile app and connect to a cloud application (e.g. over Long Term Evolution (LTE)/5G Internet) and vehicles utilize either a built-in module (with a dedicated Subscriber Identity Module (SIM)-card) or driver's smartphone.

The evaluation focused on the required transmission frequency for different car speeds and the battery life of the smartphone depending on transmission frequency. In contrast to the client/server architecture in [2.2], Liebner et al. [2.16] proposed a peer-to-peer 3G based bicycle protection prototype. The authors evaluated the High Speed Downlink Packet Access (HSDPA) latency between moving smartphone and the warning probability of the system based on "time difference between bicycle and car reaching the collision point"

2.2.3 *Hybrid*

A third option for communication is the combination of direct and cellular communication, also called hybrid communication. Sugimoto et al. [2.6], which was one of the first group that proposed the use of smartphones for a VRU CAS, used a combination LTE/WLAN communication depending on the distance between the road users. They evaluated their system throughout three publications. In [2.6] the evaluation was focused on WLAN transmission speed and transmission distance. The impact of transmission speed on their risk calculation was evaluated in [2.5]. A questionnaire based evaluation if their system could rise the attention of the car driver was finally given in [2.7]. A hybrid communication was also presented by Artail et al. [2.22]. In this publication the pedestrian's device uses a LTE cellular network while the vehicles use DSRC communication. It was evaluated how the number of pedestrians within a specific area influences the transmission time of message. An Mobile Edge Computing (MEC) based hybrid communication approach was presented by Malinverno et al. [2.23]. The basic idea here is that a server, the Edge Server, is located close to the road side and therefore can be accessed either via direct WLAN connection or using the cellular network. The authors evaluated their approach regarding network load and compared it with a distributed system where each road user runs its own collision detection algorithm.

2.3 SYSTEM ARCHITECTURES

The calculations for collision detection can be done on the devices of the road users or on a dedicated server.

2.3.1 *On Server*

The majority of publications considered collision detection on a server. This server might be located close to the road side like in [2.22]–[2.24] or further apart in a data centre [2.2], [2.24]. Both, Napolitano et al. [2.24] and Malinverno et al. [2.23], investigated the impact on collision detection delays for a cloud servers (remote) and close to the road side

located servers, also called MEC. Malinverno et al. [2.23] compared collision detection performance between centralized (cloud server) and decentralized (edge server and on device) approaches in LOS and Non-Line-of-Sight (NLOS) regarding true positives/false positives defections. A cloud based collision detection algorithm was proposed by Bagheri et al. [2.2]. The authors used location, speed, and direction of the vehicle and the pedestrian. The authors do not give information how this movement data is actually used to detect collisions. Instead the authors focussed on the evaluation of their "situation-adaptive beaconing". For this, a pedestrian's current risk level is classified as either risk free, low or high, which is based on the current distance to nearby roads and vehicles. Based on these risk levels, the transmission frequency of the pedestrian smartphone is adjusted accordingly, which allows to reduce unnecessary network communication. If a pedestrian is classified as "high risk", the server executes a collision detection algorithm with an update rate of 10 Hz. As soon as a high probability of collision is predicted, the server informs both the vehicle's and pedestrian's smartphone with a critical alert. An approach that uses servers directly located at road intersections was proposed by Artail et al. [2.22]. On this servers, called Road Side Units (RSU) by the authors, the collision detection is performed. Vehicles connect to a RSU using Vehicular Ad Hoc Network (VANet) while pedestrians connect via LTE to the RSU. Collision detection is performed on the RSUs using position, speed, direction, and some car specific physical information like road friction. Using the location from the pedestrian's smartphones it was then evaluated if a pedestrian is inside a collision zone with a vehicle.

2.3.2 *On Device*

Anaya et al. [2.8] proposed a VRU protection system in which the vehicles calculate a prediction of their future position in form of a Geographical Destination Area (GDA). The GDA is created from the vehicle's current GNSS position, its speed, yaw rate, length, and width. This GDA is then packaged in a Cooperative Awareness Message (CAM) and transmitted to the pedestrian's smartphone. The collision detection on the pedestrian's smartphone is performed if the pedestrian is inside the GDA. If the pedestrian is inside the GDA the collision detection algorithm evaluates if the pedestrian can reach the intersection point between the vehicle's and its own trajectory based on the pedestrian's maximum speed. Also, for car-to-car collision detection, different groups [2.21], [2.25], [2.26] proposed collision detection algorithms that run a device or a on-board unit in the cars. All these algorithms require at least current position, speed, and direction.

2.4 COLLISION DETECTION

Acknowledgement: This section is based on a literature research from Johann Götz which he conducted during his students project supervised by the author.

The state-of-the-art regarding collision detection between a vehicle and a VRU can be categorized in deterministic and non-deterministic approaches. Deterministic approaches, can be vision based (and thus require LOS conditions), like in [2.27], [2.28], [2.29], [2.30] or cooperative, like in [2.21], [2.31], [2.23]. Probabilistic approaches, such as [2.32] and [2.33], focus on collision detection between sets of multiple predicted future movement trajectories of the road users in order and derive a collision probability from them.

2.4.1 *Deterministic*

Deterministic approaches assume that the road users' movement is constant. Here, constant means, that the current movement vectors do not change, i.e. in case of non linear movement, rotation and acceleration are not time dependent. The current movement trajectories of the road users are extrapolated to determine if both road users would collide for any future time, which is the called the Time to Collision (TTC), or just analyse if the distance between two the two road falls below a certain distance threshold.

Deterministic LOS-dependent collision detection algorithms are vehicle-centric and thus, the collision detection always takes place from the perspective of the vehicle. Road users are only detected when they in LOS. Deterministic cooperative approaches, aim to detect road users also in NLOS scenarios. They utilize movement information from every participating road user and thus they are not necessarily vehicle-centric.

Due to the consideration of only the current movement trajectories, the deterministic approaches are often more efficient in terms of runtime compared to non-deterministic approaches. For deterministic approaches, it is required that the current position, speed, and direction of the road users are known.

2.4.1.1 *Vision-Based*

Hernandez et al. [2.28], proposed an approach to detect pedestrians in which a collision zone is generated to estimate the collision risk using the Field of View (FOV) of the car's LIDAR and detected lane cluster information. Within the collision zone, pedestrians are detected using the Closest Point of Approach (CPA).

Castro et al. [2.30] proposed a collision detection approach, that allows vehicles, pedestrian and bicycle collision detection. The ap-

proach is based on distance calculation between linear moving objects in a three-dimensional space. It utilizes video data from surveillance cameras that are placed on the streets. The camera calibration process determines the real positioning of objects in the scene. Moving 3D objects are detected using the camera calibration process and 2D-tracking. The collision risk between the objects is graded using fuzzy logic.

2.4.1.2 *Cooperative*

While relative movement information from nearby vehicles and VRUs in vision-based algorithms is processed from the perspective of the host vehicle, cooperative collision detected algorithms can use absolute coordinate systems for the road user positions in a 2- or 3-dimensional coordinate system.

Cooperative collision detected algorithms that depend on road map information, i.e., on what type of road structure, like road intersections, the road users are located and what kind of geometry the road has, i.e., how many lanes there are currently and how wide each lane is are called road map dependent.

On the contrary to road map dependent collision detection, route intersection does not rely on road map information and instead uses only the road users' absolute movement information. The deterministic road map collision detection and route intersection are reviewed in the following.

ROAD MAP DEPENDENT APPROACHES Road map dependent collision detection use a simple distance calculation between road users in combination with current road map information, e.g., the type of the road at which the host vehicle is located. Road map dependent collision detection considers typical road user intersection scenarios, e.g., in which a pedestrian crosses the street perpendicularly to the lanes. Additional information about road structure, such as the number and geometry, i.e., width, of the lanes and the sideways distance to the curb, is therefore required at any time. As a result, the deterministic, road map dependent collision detection tend to be more limited compared to other approaches, such as route intersection that is described in the following subsection.

Xiog et al. [2.34] and Varnavsky et al. [2.35] used the vehicle's longitudinal and pedestrian's lateral trajectory information to calculate an intersection point and thus the longitudinal distance of the vehicle to the intersection point. Additionally, [2.35] considered the width of the car and expanded the distance calculation by the opposite lane. On the other hand, [2.34] add an emergency and warning area to improve the performance.

Rather than calculating the collision between two individual vehicles, Artail et al. [2.22] developed a clustering algorithm that groups together cars that drive in the same direction in order to define "collision

zones" (C-Zones) which cover the entire width of a lane. A group is formed by comparing overlapping breaking distances from neighbouring cars. Among the group, a leader is chosen as a midpoint reference from the closest neighbours in order to identify the highest forward breaking distance. A collision is detected when a pedestrian, that carries a mobile device, enters a C-Zone. This C-Zone were then expanded by a "Look-Ahead Zone" (L-Zone) to detect pedestrians, that might collide with the vehicle, during a time period.

A different approach by Oda et al. [2.11] relies on distance calculation using Radio Frequency Identification (RFID) technology that can be utilized at road intersections. At an intersection, a magnetic field is generated by at least two 125kHz LF-signal generators. Active 950MHz RFID tags need to be carried by pedestrians or installed on bicycles, which send the tag ID information to an on-board device of a vehicle to alert the driver after the tag has entered a magnetic field. Though this approach is not limited by road structure itself, it still depends on RFIDs that have to be installed at every road intersection.

ROUTE INTERSECTION Route intersection approaches use the movement information of two road users for modelling their movement in a two- or three-dimensional coordinate system. Within this coordinate system it is then evaluated if there is a time t for which both road users would be at the same position which would indicate a collision. Using a route intersection approach, it is therefore possible to calculate the future collision point and TTC between two road users. Compared to vision-based and road map dependent collision detection, route intersection is not limited by road structure or LOS.

An often used route intersection approach intended for vehicle-to-vehicle collision detection was given by Miller et al. [2.21] (also Yang et al. [2.36]) who use linear trajectory information of each vehicle in order to calculate an intersection point. A collision is detected whenever the time to the intersection point of both road users' trajectories are equal, i.e., they will be at the same future position. For collision detection in which the road users are not represented as points, the difference between both TTCs is used to compare whether it is greater than a given threshold. Jimenez et al. [2.31] improved the approach of Miller et al. [2.21] by calculating an intersection area between the trajectories of both vehicles based on their rectangle geometry in order to determine intersection points which represent the corners of the intersection area.

Zadeh et al. [2.37], detect VRUs only when they are inside a predefined activation area. The authors suggest a "Three Phases Warning System" that is based on fuzzy neural network. In the activation phase, a trapezoidal area is calculated based on the location, speed, acceleration and direction of the vehicle. When a pedestrian is detected within the trapezoidal area, the prediction phase uses route intersec-

tion calculation for both road users and their time difference to to the intersection point. The system proposed by [2.8], already mentioned in Section 2.2.1, uses route intersection when one or multiple pedestrians are inside a GDA. The TTCs and intersection points are determined between the vehicle and each intersection point that is calculated from the shape of the GDA and pedestrians' position.

2.4.2 *Probabilistic Collision Detection*

Probabilistic collision detection approaches take into account multiple different trajectories for the road users. In this way probabilistic consider uncertainty due to sensor inaccuracy and changes in the road users' current movements. This results in a set of multiple possible current positions (due to positioning errors) and future trajectories (due to uncertain movement dynamics in the future, like rotation and acceleration) with corresponding probabilities. A drawback for probabilistic collision detection is that due to the higher amount of resulting calculations for different movement trajectories, this type of collision detection requires more computational time.

Among the probabilistic collision detection approaches, there are also some that are depending on the current road map and other that do not. Both are described in the following sections.

2.4.2.1 *Road Map dependent*

Because of the demanding computational power of probabilistic approaches, the applicability are sometimes restricted to certain scenarios to limit the amount of possible movement trajectories. For instance, the information about the current road map can be used for determining the course of a vehicles along the street.

Althoff et al. [2.32] described a probabilistic collision detection approach that uses "stochastic reachable sets", i.e, predicted movement trajectories, of road users, depending on the road structure. The reachable sets are discretized into segments. The overlapping between the segments is determined in order to calculate the collision probability.

Similarly to [2.32], Greene et al. [2.38] introduce an approach that consists of two parts: "preliminary collision assessment" and "specialized collision assessment". The preliminary assessment is a fast, imprecise pre-computations of possible collision scenarios using the intersection of segmented cones that are determined from extrapolated trajectories of the road user, its worst-case parameters and time in a three-dimensional space. Additional information about the road structure is required in order to restrict the rectangular shape of collision cones. The specialized assessment focuses on close examination of each possible collision scenario that is determined by preliminary assessment. Using statistic interference, the overlapping ratio between

two segmented cones is determined which also represents the collision probability.

2.4.2.2 *Road Map independent*

Besides the high computational demand, many approaches renounce the use of road map information but rely on other ways, like using a Kalman filter for position prediction, to restrict possible future trajectories of the road user.

Using their "Sector Overlap Detection Algorithm", Lin et al.[2.39] calculate sectors of overlapping arcs, which are determined with the deviation of road user movement, in order to determine the overlapping rate and thus the collision probability.

By utilizing stochastic variables for other vehicles as control inputs for collision prediction, Eidehall et al. [2.33] create an approximation of vehicles' future trajectories with Monte Carlo sampling in order to reduce computationally demanding sample sets. In order to support circular movement of the vehicle in curved streets, they use a curved two-dimensional coordinate system.

Campos et al. [2.40] focus on frontal collision detection. Movement prediction, that is based on unscented Kalman filter, determines collision zones using a-priori and a-posteriori estimations. The a-priori estimation uses a motion model to generate states at future time steps of a vehicle. The a-posteriori estimation filters out measurement noises of each step in order to refine the calculated states. If the collision zones intersect, it means that potential collisions exist. A probabilistic threat assessment is used to calculate a collision zone that contains future collisions between both rectangular shaped vehicles. From the collision zone, the collision probability is derived.

2.5 CONCLUSION

As shown in the above sections, a comprehensive, in-depth performance analysis on how a collision avoidance system might work in real-world usage is missing. The publications regarding cooperative VRU CAS basically focused on system concepts, improving position accuracy, or solutions for reliable transmission of movement information. Besides the publications [2.9], [2.16], [2.17] which analysed at least the required GNSS accuracy but in regard to their very specific system concepts, like the activation of an avoidance manoeuvrer in [2.9] to reduce impact severity, none of these publications delivered a comprehensive analysis of the general required accuracies for position, direction, and speed and their influence on VRU collision avoidance performance. Accordingly, there neither exits an objective metric to assess the performance of a cooperative VRU CAS, nor are the required sensor accuracies known to achieve a "good" collision avoidance performance. In addition, regarding the collision detection algorithms

used in VRU CASes, the geometry of the VRU is neglected. Considering the state-of-the-art research for VRU collision detection, is was shown that current algorithms for VRU collision detection are either "thought from the perspective of the vehicle"; and are therefore LOS-dependent, or a point was generally used to model the VRU [2.23], [2.35], [2.37] in case of TTC or distance based collision detection.

The remainder of this dissertation closes this gap according to the outlined contributions from Section 1.3. In detail, a precise VRU collision detection algorithm supporting a circle geometry for pedestrians is and a performance metric for cooperative VRU CASes depending on movement sensor accuracies on the VRU side is introduced. The precise collision detection algorithm is then used to investigate the required movement sensor accuracies on the VRU side regarding the performance metric. In addition the performance of a camera-based VRU CAS with a cooperative VRU CAS in NLOS scenarios is compared.

REFERENCES

[2.1] T. Gandhi and M. M. Trivedi, 'Pedestrian Protection Systems: Issues, Survey, and Challenges', *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 413–430, Sep. 2007. DOI: 10.1109/TITS.2007.903444 (cit. on p. 15).

[2.2] M. Bagheri, M. Siekkinen and J. K. Nurminen, 'Cellular-based vehicle to pedestrian (V2P) adaptive communication for collision avoidance', in *Proc. 3rd International Conference on Connected Vehicles and Expo (ICCVE)*, Vienna, Austria, Sep. 2014, pp. 450–456. DOI: 10.1109/ICCVE.2014.7297588 (cit. on pp. 15, 17–19).

[2.3] K. Dhondge, S. Song, B.-Y. Choi and H. Park, 'WiFiHonk: Smartphone-Based Beacon Stuffed WiFi Car2X-Communication System for Vulnerable Road User Safety', in *Proc. Vehicular Technology Conference (VTC 2014-Spring)*, Seoul, South Korea, May 2014, pp. 1–5. DOI: 10.1109/VTCSpring.2014.7023146 (cit. on pp. 15–17).

[2.4] X. Wu, R. Miucic, S. Yang, S. Al-Stouhi, J. Misener, S. Bai and W.-h. Chan, 'Cars Talk to Phones: A DSRC Based Vehicle-Pedestrian Safety System', in *Proc. 80th Vehicular Technology Conference (VTC2014-Fall)*, Vancouver, Canada, Sep. 2014, pp. 1–7. DOI: 10.1109/VTCFall.2014.6965898 (cit. on pp. 15–17).

[2.5] C. Sugimoto, Y. Nakamura and T. Hashimoto, 'Prototype of pedestrian-to-vehicle communication system for the prevention of pedestrian accidents using both 3G wireless and WLAN communication', in *Proc. 3rd International Symposium on Wireless Pervasive Computing (ISWPC)*, Santorini, Greece, May 2008,

pp. 764–767. DOI: 10.1109/ISWPC.2008.4556313 (cit. on pp. 15, 18).

[2.6] C. Sugimoto, Y. Nakamura and T. Hashimoto, 'Development of Pedestrian-to-Vehicle Communication System Prototype for Pedestrian Safety Using both Wide-Area and Direct Communication', in *Proc. 22nd International Conference on Advanced Information Networking and Applications (AINA)*, Ginowan, Japan, Mar. 2008, pp. 64–69. DOI: 10.1109/AINA.2008.126 (cit. on pp. 15, 18).

[2.7] C. Sugimoto and Y. Nakamura, 'Provision of information support by pedestrian-to-vehicle communication system', in *Proc. 8th International Conference on ITS Telecommunications*, Phuket, Thailand, Oct. 2008, pp. 160–163. DOI: 10.1109/ITST.2008.4740248 (cit. on pp. 15, 18).

[2.8] J. J. Anaya, P. Merdrignac, O. Shagdar, F. Nashashibi and J. E. Naranjo, 'Vehicle to pedestrian communications for protection of vulnerable road users', in *Proc. Intelligent Vehicles Symp. (IV)*, Dearborn, USA, Jun. 2014, pp. 1037–1042. DOI: 10.1109/IVS.2014.6856553 (cit. on pp. 15, 17, 19, 23).

[2.9] P. Themann, J. Kotte, D. Raudszus and L. Eckstein, 'Impact of positioning uncertainty of vulnerable road users on risk minimization in collision avoidance systems', in *Proc. Intelligent Vehicles Symposium (IV)*, Seoul, South Korea, Jun. 2015, pp. 1201–1206. DOI: 10.1109/IVS.2015.7225846 (cit. on pp. 15, 16, 24).

[2.10] R. Raßhofer, D. Schwarz, E. Biebl, C. Morhart, O. Scherf, S. Zecha, R. Grünert, and H. Frühauf, 'Pedestrian Protection Systems using Cooperative Sensor Technology', in *Advanced Microsystems for Automotive Applications 2007*, Valldorf, Jürgen and Gessner, Wolfgang, Ed., Berlin, Heidelberg: Springer, 2007, pp. 135–145. DOI: 10.1007/978-3-540-71325-8_11 (cit. on p. 15).

[2.11] H. Oda, S. Kubota and Y. Okamoto, 'Research on technology for reducing sudden pedestrian or cyclist accidents with vehicles', in *Proc. IEEE Intelligent Transportation Systems Conference*, Sep. 2007, pp. 1032–1036. DOI: 10.1109/ITSC.2007.4357641 (cit. on pp. 15, 22).

[2.12] A. Flach and K. David, 'A Physical Analysis of an Accident Scenario between Cars and Pedestrians', in *Proc. 70th Vehicular Technology Conference (VTC 2009-Fall)*, Anchorage, USA, Sep. 2009, pp. 1–5. DOI: 10.1109/VETECF.2009.5378805 (cit. on p. 16).

[2.13] K. David and A. Flach, 'CAR-2-X and Pedestrian Safety: Innovative Collision Avoidance System', *IEEE Vehicular Technology Magazine*, vol. 5, no. 1, pp. 70–76, 2010. DOI: 10.1109/MVT.2009.935536 (cit. on p. 16).

[2.14] S. Engel, K. David, D. Warkow and M. Holzknecht, 'Car2-Pedestrian Positioning: Methods for Improving GPS Positioning in Radio-Based VRU Protection Systems', in *6. Tagung Fahrerassistenzsysteme*, Munich, Germany, Nov. 2013, pp. 1–7 (cit. on p. 16).

[2.15] S. Engel, C. Kratzsch and K. David, 'Car2Pedestrian–Communication: Protection of Vulnerable Road Users Using Smartphones', in *Proc. Advanced Microsystems for Automotive Applications 2013*, Heidelberg, 2013, pp. 31–41. DOI: 10.1007/978-3-319-00476-1_4 (cit. on p. 16).

[2.16] M. Liebner, F. Klanner and C. Stiller, 'Active safety for vulnerable road users based on smartphone position data', in *Proc. Intelligent Vehicles Symp. (IV)*, Gold Coast City, Australia, Jun. 2013, pp. 256–261. DOI: 10.1109/IVS.2013.6629479 (cit. on pp. 16, 18, 24).

[2.17] J. Rünz, F. Flehmig, W. Rosenstiel and M. Knoop, 'Requirements and Evaluation of a Smartphone Based Dead Reckoning Pedestrian Localization for Vehicle Safety Applications', in *Advanced Microsystems for Automotive Applications 2016*, ser. Lecture Notes in Mobility, T. Schulze, B. Müller and G. Meyer, Eds., Cham: Springer, 2016, pp. 3–17. DOI: 10.1007/978-3-319-44766-7_1 (cit. on pp. 16, 24).

[2.18] M. Bachmann, M. Morold and K. David, 'Improving Smartphone Based Collision Avoidance by Using Pedestrian Context Information', in *2017 IEEE International Conference on Pervasive Computing and Communications Work in Progress (PerCom WiP 2017)*, Kona, USA, Mar. 2017, pp. 2–5. DOI: 10.1109/PERCOMW.2017.7917507 (cit. on p. 16).

[2.19] M. Bachmann, M. Morold, K. David and P. Henkel, 'The Wireless Seat Belt Requirements, Experiments, and Solutions for Pedestrian Safety', in *14th Workshop on Context and Activity Modeling and Recognition*, Athens, Greece, Mar. 2018, pp. 250–255. DOI: 10.1109/PERCOMW.2018.8480123 (cit. on p. 16).

[2.20] M. Bieshaar, S. Zernetsch, A. Hubert, B. Sick and K. Doll, 'Cooperative Starting Movement Detection of Cyclists Using Convolutional Neural Networks and a Boosted Stacking Ensemble', *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 4, pp. 534–544, 2018. DOI: 10.1109/TIV.2018.2873900 (cit. on p. 16).

[2.21] R. Miller and Q. Huang, 'An adaptive peer-to-peer collision warning system', in *IEEE 55th Vehicular Technology Conference*, Birmingham, USA, May 2002, pp. 317–321. DOI: 10.1109/VTC.2002.1002718 (cit. on pp. 17, 19, 20, 22).

[2.22] H. Artail, K. Khalifeh and M. Yahfoufi, 'Avoiding car-pedestrian collisions using a VANET to cellular communication framework', in *13th International Wireless Communications and Mobile Computing Conference*, Rome, Italy, Jun. 2017, pp. 458–465. DOI: 10.1109/IWCMC.2017.7986329 (cit. on pp. 18, 19, 21).

[2.23] M. Malinverno, G. Avino, C. Casetti, C. F. Chiasserini, F. Malandrino and S. Scarpina, 'Edge-Based Collision Avoidance for Vehicles and Vulnerable Users: An Architecture Based on MEC', *IEEE Vehicular Technology Magazine*, vol. 15, no. 1, pp. 27–35, 2020. DOI: 10.1109/MVT.2019.2953770 (cit. on pp. 18–20, 25).

[2.24] A. Napolitano, G. Cecchetti, F. Giannone, A. L. Ruscelli, F. Civerchia, K. Kondepu, L. Valcarenghi and P. Castoldi, 'Implementation of a MEC-based Vulnerable Road User Warning System', in *2019 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE)*, Turin, Itlay, Jul. 2019, pp. 1–6. DOI: 10.23919/EETA.2019.8804497 (cit. on p. 18).

[2.25] C.-M. Huang, S.-Y. Lin, C.-C. Yang and C.-H. Chou, 'A Collision Pre-Warning Algorithm Based on V2V Communication', in *Proc. 4th International Conference on Ubiquitous Information Technologies & Applications*, Fukuoka, Japan, Dec. 2009, pp. 1–6. DOI: 10.1109/ICUT.2009.5405740 (cit. on p. 19).

[2.26] S. Joerer, M. Segata, B. Bloessl, R. Lo Cigno, C. Sommer and F. Dressler, 'A Vehicular Networking Perspective on Estimating Vehicle Collision Probability at Intersections', *IEEE Transactions on Vehicular Technology*, vol. 63, no. 4, pp. 1802–1812, 2014. DOI: 10.1109/TVT.2013.2287343 (cit. on p. 19).

[2.27] J. Wang, C. Yu, S. E. Li and L. Wang, 'A forward collision warning algorithm with adaptation to driver behaviors', *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1157–1167, 2016. DOI: 10.1109/TITS.2015.2499838 (cit. on p. 20).

[2.28] D. C. Hernandez, A. Filonenko, J. Hariyono, A. Shahbaz and K.-H. Jo, 'Laser based collision warning system for high conflict vehicle-pedestrian zones', in *Proc. IEEE 25th International Symposium on Industrial Electronics (ISIE)*, Jun. 2016, pp. 935–939. DOI: 10.1109/ISIE.2016.7745016 (cit. on p. 20).

[2.29] E. Dagan, O. Mano, G. P. Stein and A. Shashua, 'Forward collision warning with a single camera', in *Proc. IEEE Intelligent Vehicles Symposium*, Jun. 2004, pp. 37–42. DOI: 10.1109/IVS.2004.1336352 (cit. on p. 20).

[2.30] J. L. Castro, M. Delgado, J. Medina and M. D. Ruiz-Lozano, 'An expert fuzzy system for predicting object collisions. its application for avoiding pedestrian accidents', *Expert Systems with Applications*, vol. 38, no. 1, pp. 486–494, 2011. DOI: 10.1016/j.eswa.2010.06.088 (cit. on p. 20).

[2.31] F. Jiménez, J. E. Naranjo and F. García, 'An improved method to calculate the time-to-collision of two vehicles', *International Journal of Intelligent Transportation Systems Research*, vol. 11, no. 1, pp. 34–42, 2013. DOI: 10.1007/s13177-012-0054-4 (cit. on pp. 20, 22).

[2.32] M. Althoff, O. Stursberg and M. Buss, 'Model-based probabilistic collision detection in autonomous driving', *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 299–310, 2009. DOI: 10.1109/TITS.2009.2018966 (cit. on pp. 20, 23).

[2.33] A. Eidehall and L. Petersson, 'Statistical threat assessment for general road scenes using monte carlo sampling', *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 137–147, 2008. DOI: 10.1109/TITS.2007.909241 (cit. on pp. 20, 24).

[2.34] G. Xiog, T. Yang, M. Li, Y. Zhang, W. Song and J. Gong, 'A Novel V2X-based Pedestrian Collision Avoidance System and the Effects Analysis of Communication Delay and Packet Loss on Its application', in *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, Sep. 2018, pp. 1–6. DOI: 10.1109/ICVES.2018.8519600 (cit. on p. 21).

[2.35] A. N. Varnavsky and N. V. Sinitsina, 'Statistical modeling the probability of some types of automobile-pedestrian accidents', in *Proc. 3th International Scientific-Technical Conference on Actual Problems of Electronic Instrument Engineering (APEIE)*, Oct. 2016, pp. 216–220. DOI: 10.1109/APEIE.2016.7806452 (cit. on pp. 21, 25).

[2.36] J. Yang, J. Wang and B. Liu, 'An intersection collision warning system using wi-fi smartphones in vanet', in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, Dec. 2011, pp. 1–5. DOI: 10.1109/GLOCOM.2011.6134294 (cit. on p. 22).

[2.37] R. B. Zadeh, M. Ghatee and H. R. Eftekhari, 'Three-phases smartphone-based warning system to protect vulnerable road users under fuzzy conditions', *IEEE Transactions on Intelligent*

Transportation Systems, vol. 19, no. 7, pp. 2086–2098, July 2018. DOI: 10.1109/TITS.2017.2743709 (cit. on pp. 22, 25).

[2.38] D. Greene, J. Liu, J. Reich, Y. Hirokawa, A. Shinagawa, H. Ito and T. Mikami, 'An efficient computational architecture for a collision early-warning system for vehicles, pedestrians, and bicyclists', *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 942–953, 2011. DOI: 10.1109/TITS.2010.2097594 (cit. on p. 23).

[2.39] C.-H. Lin, Y.-T. Chen, J.-J. Chen, W.-C. Shih and W.-T. Chen, 'Psafety: A collision prevention system for pedestrians using smartphone', in *Proc. IEEE 84th Vehicular Technology Conference (VTC Fall)*, Sep. 2016, pp. 1–5. DOI: 10.1109/VTCFall.2016.7881183 (cit. on p. 24).

[2.40] G. R. de Campos, A. H. Runarsson, F. Granum, P. Falcone and K. Alenljung, 'Collision avoidance at intersections: A probabilistic threat-assessment and decision-making system for safety interventions', in *Proc. IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)*, Oct. 2014, pp. 649–654. DOI: 10.1109/ITSC.2014.6957763 (cit. on p. 24).

3

PRECISE COLLISION DETECTION

To address challenge I "How to precisely detect collisions in *cooperative* VRU Collision Avoidance Systems (cooperative VRU CASes)", this chapter provides an algorithm to precisely detect collisions between a vehicle and a Vulnerable Road User (VRU). The algorithm is based on the assumption that both road users, i.e. the vehicle and the VRU, can be suitable modelled either as a rectangle or a circle geometry. Considering this assumption, precise means that the proposed algorithm is able to determine the exact impact position, i.e. the tangential point of the used vehicle's and VRU's geometries. The algorithm is based on non-accelerated linear movement extrapolation. Using non-accelerated linear movement extrapolation, a collision is detected if there is a time t for which there is a tangential point between the two road users' geometries. If there is such a t, this time is the Time to Collision (TTC). In other words, the algorithms determines if there will be a collision between two road users if both do not change their movement until the TTC. Other algorithms published so far are not able to precisely detect collisions under the assumption that the vehicle should be modelled as a rectangle and the pedestrian as a circle. This is because they either use distance thresholds which do not consider the impact position or use a point geometry for the pedestrian.

This chapter is structured as follows: In the next Section 3.1, it is detailed how real world scenarios are modelled in a so called Road Space Model (RSM) for collision detection between a vehicle and a pedestrian. Next, in Section 3.2 the linear movement equations are introduced. In Section 3.3, the general idea of the collision detection algorithm is described and in Section 3.4 the most basic formula for TTC calculation is derived using point geometries for the road users. Next, in Section 3.5 two transformation approaches for the RSM, called "normalized transformation" and "relative transformation" are proposed. The transformations are the basis for a "divide and conquer" approach, called "subdivision and reconstruction", introduced in Section 3.6, to calculate the TTC for rectangle and circle geometries.

In Section 3.7 the calculation of the TTC for rectangle-to-point/point-to-rectangle subdivision components and in Section 3.8 the calculation of the TTC for the circle-to-point/point-to-circle subdivision components are detailed.

Finally, in Section 3.9, it is explained how to reconstruct the TTC for a rectangle-to-circle/circle-to-rectangle representation of the road users from the TTC values of the less complex components rectangle-to-point and circle-to-point .

Acknowledgement: Parts of this Chapter, especially the "subdivision and reconstruction" part and most of the figures, were initially created by Johann Götz during his Master-thesis supervised by the author.

3.1 MODELLING THE ROAD SPACE

The road space is modelled in a special 2-dimensional metric coordinate system, called the Road Space Model (RSM). An instance of a RSM always evaluates a potential vehicle-VRU collision in a 1:1 relation. In detail, in situations involving more than one vehicle-VRU pair, each pair is evaluated in an own, separate instance of a RSM. The RSM maps a small geographic area of the earth ellipsoid around two road users to a flat 2-dimensional metric coordinate system. A RSM is created by setting the World Geodetic System 1984 (WGS84) position $(\mathrm{long}_c, \mathrm{lat}_c)$ of road user c to the origin of the two-dimensional RSM, i.e. $(x_c, y_c)^T = (0,0)^T$. To map the WGS84 position $(\mathrm{long}_p, \mathrm{lat}_p)$ of road user p to the corresponding position (x_p, y_p) in the RSM, the inverted Vincenty's formulae [3.1] is evaluated. The inverted Vincenty's formulae gives the metric distance d and azimuth φ_{WGS84} between the two WGS84 coordinates $(\mathrm{long}_c, \mathrm{lat}_c)$ and $(\mathrm{long}_p, \mathrm{lat}_p)$. Using d and φ_{WGS84} the position of p is set to

$$\begin{pmatrix} x_p \\ y_p \end{pmatrix} = d \cdot \begin{pmatrix} \sin \varphi_{WGS84} \\ \cos \varphi_{WGS84} \end{pmatrix} \tag{3.1}$$

Be aware that the azimuth φ_{WGS84} from WGS84 coordinates is defined to be 0° in north direction (180°in south direction) and 90°in east direction (270° in west direction). Thus the sine function occurs for the x-component and the cosine function for y-component in the RSM, effecting that an angle of 0° is parallel to the y-axis in the RSM and not to parallel to the x-axis like in a Cartesian coordinate system.

The azimuth φ_{WGS84} for the road users $i \in \{p, c\}$ can be taken from Global Navigation Satellite Systems (GNSS) bearing and, if available on the device, a digital compass. Likewise the speed v_i for both road users $i \in \{p, c\}$, can be taken from GNSS speed measurements.

Within the RSM, the vehicle and the VRU can be represented either as rectangle, circle, or point geometry. The geometric centres of the used geometries represents the current position of the road users in the RSM. To simplify the terminology and used symbols, without loss of any generality, in the following a RSM between a car (c) and a pedestrian (p) as shown in Fig. 3.1 is assumed. Thus, the subscript $_c$ will refer to the involved vehicle. i.e. a car (c), and the subscript $_p$ to the involved VRU, i.e. a pedestrian (p). Unless expressly stated otherwise, the pedestrian is modelled by a circle with the radius $r_p = 0.5\,\mathrm{m}$ and the car by a rectangle with the length $l_c = 4\,\mathrm{m}$ and the width $w_c = 2\,\mathrm{m}$. Depending on the architecture of a cooperative

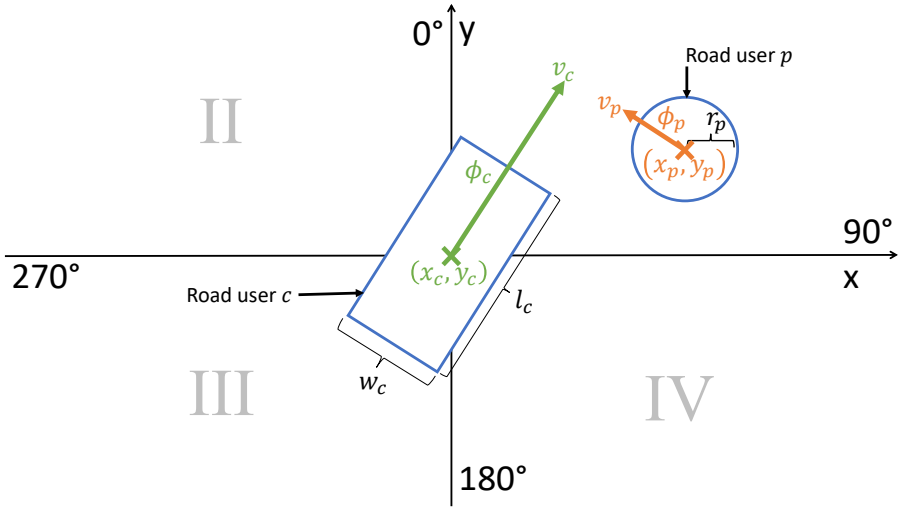


Fig. 3.1: Road Space Model of a real-world collision scenario between a pedestrian (p) and a car (c) based on the two-dimensional Road Space Model (RSM) for which the longitude (east-west) is aligned to x-axis and the latitude (north-south) is aligned to the y-axis.

VRU CAS, there are multiple ways of creating a RSM instance from real world sensor measurements for both road users' movements. For the scope of this dissertation is is assumed that a RSM is instantiated using WGS84 coordinates from smartphone GNSS.

3.2 ROAD USER MOVEMENT

The future position $\mathbf{r}_i(t) = (x_{t_i}.y_{t_i})^T$ for the road users $i \in \{p, c\}$, depending on the prediction time t is given by the non-accelerated linear movement equation

$$\mathbf{r}_i(t) = v_i \cdot t \cdot \begin{pmatrix} \sin\phi_i \\ \cos\phi_i \end{pmatrix} + \mathbf{r}_{0_i} \tag{3.2}$$

where v_i is the speed, ϕ_i is the direction (corresponding to the azimuth φ_{WGS84} from WGS84 coordinates), and $\mathbf{r}_{0_i} = (x_i, y_i)^T$ is the current position, e.g. from converted GNSS measurements. Since (3.2) is completely defined by the parameters x_i, y_i, ϕ_i, v_i, a 3-tuple

$$\mathbf{m}_i = ((x_i, y_i)^T, \phi_i, v_i) = (\mathbf{r}_{0_i}, \phi_i, v_i) \tag{3.3}$$

is used for referring to a specific linear movement equation in the form of (3.2). $\mathbf{m}_i$ is called the movement vector of a road user i.

3.3 BASIC APPROACH FOR COLLISION DETECTION

The most basic case for detecting a collision between two non-accelerated linear moving road user within the above introduced RSM is to neglect any geometry and evaluate if both road users will be at the some

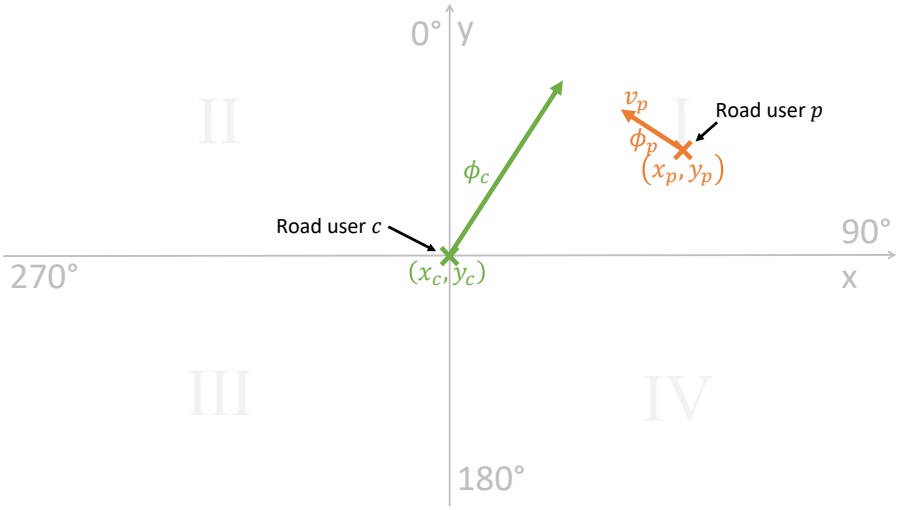


Fig. 3.2: Point-to-point Road Space Model (RSM)

position for any time t in the future, see Fig. 3.2. A RSM using points for the road users is called point-to-point model.

Using eq. (3.2), an impending collision is detected if there is a time t for which the road user p, whose future positions are defined by $\mathbf{r}_p(t)$ with the movement vector $\mathbf{m}_p$, is at the same position as the road user c, whose future positions is defined by $\mathbf{r}_c(t)$ with the movement vector $\mathbf{m}_c$. If there is such a t, then TTC $= t$. Accordingly, a binary function $\mathrm{col}(\mathbf{m}_p, \mathbf{m}_c)$ is used that evaluates if there is an impeding collision as follows:

$$\mathrm{col}(\mathbf{m}_p, \mathbf{m}_c) = \begin{cases} 1 & \text{, if TTC} \geqslant 0 \\ 0 & \text{, else} \end{cases} \tag{3.4}$$

The general idea to determine the TTC is to equate the linear movement equations $\mathbf{r}_p(t)$ of the road user p and $\mathbf{r}_c(t)$ of the road user c, so that $\mathbf{r}_p(t) = \mathbf{r}_c(t)$. The formula is rearranged according to (3.5), to calculate t_x and t_y which represent the times at which c and p are at same x- and y-position, separately.

$$\begin{aligned} \mathbf{r}_p(t) &= \mathbf{r}_c(t) \\ v_p \cdot t \cdot \begin{pmatrix} \sin\phi_p \\ \cos\phi_p \end{pmatrix} + \begin{pmatrix} x_p \\ y_p \end{pmatrix} &= v_c \cdot t \cdot \begin{pmatrix} \sin\phi_c \\ \cos\phi_c \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{aligned} \tag{3.5}$$

$$\begin{aligned} \Rightarrow t_x &= \frac{x_p - x_c}{v_c \sin\phi_c - v_p \sin\phi_p}, \\ t_y &= \frac{y_p - y_c}{v_c \cos\phi_c - v_p \cos\phi_p} \end{aligned} \tag{3.6}$$

3.4 BASIC POINT-TO-POINT TTC ALGORITHM

Using t_x and t_y the TTC for two road users represented as points can be calculated using algorithm 1.

Algorithm 1: TTC for point-to-point RSM

```
   input  : t_x, t_y
   output: TTC
1  TTC:= null;
2  if t_x ⩾ 0 ∧ t_y ⩾ 0 then
3  |  if t_x = t_y then
4  |  |  TTC:= t_x;
5  |  else
6  |  |  if t_x = 0 then
7  |  |  |  TTC:= t_y;
8  |  |  end
9  |  |  if t_y = 0 then
10 |  |  |  TTC:= t_x;
11 |  |  end
12 |  end
13 end
```

If TTC is null, then there is no collision for the current point-to-point scenario.

Using two points to model the road users is unrealistic, because they only collide when the positions of the points overlap. Since this situation almost never occurs, this calculation is not suited for collision detection in real life situations. However, both equations for t_x and t_y from this section can be reused to determine collision between more complex geometry combinations, such as rectangle-to-point or circle-to-point. By including the properties of a rectangle or a circle geometry in the calculation, it allows to focus on the geometry of either road user c or p. Before geometries are introduced in Section 3.6, in the next Section 3.5 two transformations for a RSM are proposed that allow to shorten the terms for t_x and t_y in (3.6), to reduce the complexity for determining t_x and t_y. This complexity reduction will become useful for determining the TTC when geometries for road users are considered.

3.5 ROAD SPACE MODEL TRANSFORMATIONS

The eqs. (3.5) and (3.6) can be simplified by transforming the RSM to an other "perspective", i.e. change the frame of reference. Within this dissertation two transformations are used, see Fig. 3.3. One transformation option is a "Normalized RSM". Transforming a RSM in a normalized a RSM consists of two steps. Fist the translation of the road

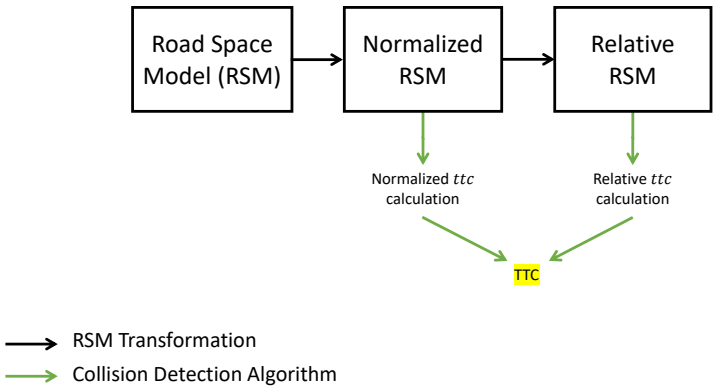


Fig. 3.3: Road Space Model Transformation Steps

users coordinates so that the road user c is at the origin $(0, 0)$, keeping the relative relation between the position of c and p. Second, rotating the translated positions of c and p so that the movement direction of c is aligned to one of the axes. The translation and rotation steps are shown in Fig. 3.4 and formally detailed in 3.5.1.

The second option is a relative RSM transformation which based on an already normalized RSM, see the flowchart in Fig. 3.3. The idea is to change the frame of reference from an absolute coordinate system to a relative coordinate system, as detailed in Section 3.5.2. In a relative RSM the current position of one of the road users is the reference point for the origin of the RSM coordinate system. The main advantage for the relative RSM transformation is that all movement dynamics are projected to one road user while the other road users remains in the origin of the RSM coordinate system. A drawback of the relative transformation is that it requires some additional trigonometric operations compared to the normalized transformation. Nevertheless, as it will be discussed in Chapter 5 there are some operations for which the relative RSM transformation is worthwhile to reduced the overall computational complexity. Which transformation should be uses in a real VRU Collision Avoidance System (VRU CAS) mainly depends on the kind of movement measurements, i.e. either given as absolute WGS84 coordinates or relative coordinates from a road user's perspective.

3.5.1 *Normalized Transformation*

To conduct the transformation of a non-transformed RSM, directly created from real world measurements as seen Fig. 3.1, two steps are performed: Translation and Rotation. Let $(x_c, y_c)^T$ be the original position of road user c and $(x_p, y_p)^T$ the original position of road user p in the non-transformed RSM.

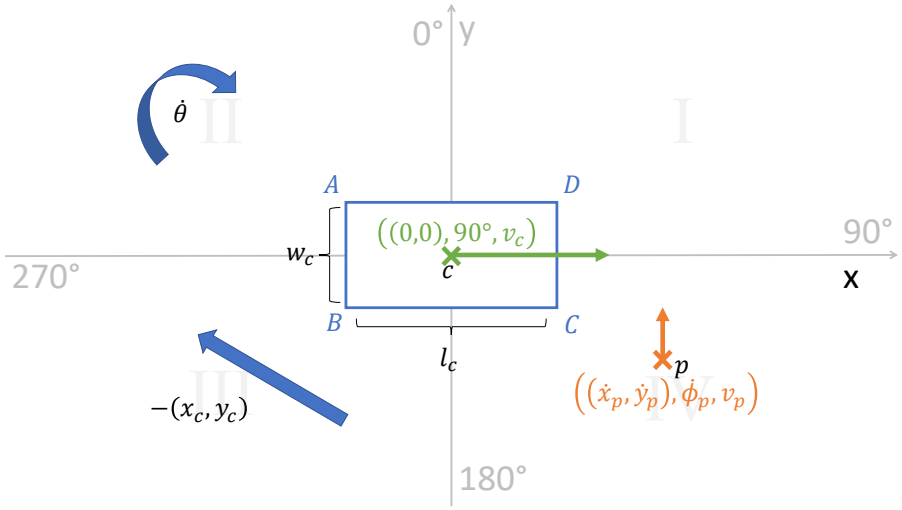


Fig. 3.4: Normalized RSM

First, for the translation, if not already done in the mapping process from real world sensor measurements to the 2-dimensional coordinate system of the RSM during initialization, the road user model c is translated by $(-x_c, -y_c)^T$ to the origin $(0,0)^T$.

Second, for the rotation, the direction of road user c is set to $\dot{\phi}_c = 90°$, i.e. along to the x-axis of the RSM. This effects that the y-coordinate becomes constant $\dot{y}_c = 0$ for the linear movement equation eq. (3.2) and thus it can be simplified for road user c by

$$\dot{\mathbf{r}}_c(t) = \begin{pmatrix} \dot{x}_c \\ 0 \end{pmatrix} = \begin{pmatrix} v_c \cdot t \\ 0 \end{pmatrix} \tag{3.7}$$

Given that the original direction ϕ_c of road user c the rotation angle $\dot{\theta}$ for a normalized RSM transformation is obtained by

$$\dot{\theta} = 90° - \phi_c \tag{3.8}$$

To keep the original relation between both road users, the position of road user p needs to be translated and rotated by $\dot{\theta}$ as well. The normalized direction of road user p given the original direction ϕ_p is $\dot{\phi}_p = \phi_p + \dot{\theta}$.

The translated and rotated position for road user p is then obtained in one step using eq. (3.9).

$$\begin{aligned} \dot{x}_p &= (x_p - x_c)\cos\dot{\theta} - (y_p - y_c)\sin\dot{\theta}, \\ \dot{y}_p &= (y_p - y_c)\cos\dot{\theta} + (x_p - x_c)\sin\dot{\theta} \end{aligned} \tag{3.9}$$

Both steps, translation and rotation, and the resulting normalized RSM are sketched up in Fig. 3.4.

Using the normalized RSM, the calculation of t_x and t_y can be simplified due to $\dot{x}_c = 0$, $\dot{y}_c = 0$, $\sin\dot{\phi}_c = 1$ and $\cos\dot{\phi}_c = 0$, see (3.10) and (3.11). Since $\cos\dot{\phi}_p = 0$ for $\dot{\phi}_p = 90°$ and $\dot{\phi}_p = 270°$, i.e

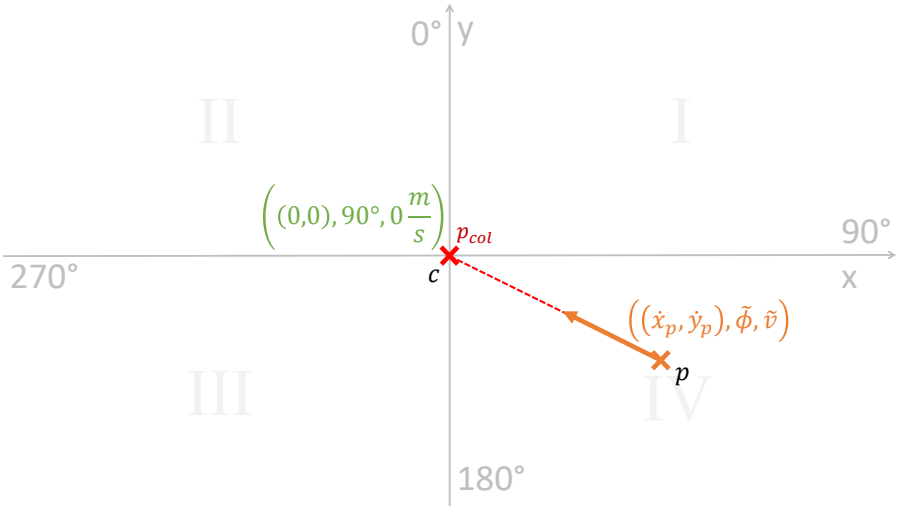


Fig. 3.5: Relative point-to-point RSM

moving parallel to the x-axis, t_y get undefined (division by zero) for $\cos \dot{\phi}_p = 0$. To handle $\cos \dot{\phi}_p = 0$ to two cases need to be considered: First, $\dot{y}_p = 0 \Rightarrow t_y = 0$, i.e road user p moves at the x-axis as road user c. Second, $\dot{y}_p \neq 0 \Rightarrow t_y = \infty$, since road user p moves parallel either above or beneath to the x-axis.

$$\Rightarrow t_x = \frac{\dot{x}_p}{\dot{v}_c - \dot{v}_p \sin \dot{\phi}_p} \tag{3.10}$$

$$\Rightarrow t_y = \begin{cases} 0 & , \text{if } \dot{\phi}_p \in \{90°, 270°\} \wedge \dot{y}_p = 0 \\ \infty & , \text{if } \dot{\phi}_p \in \{90°, 270°\} \wedge \dot{y}_p \neq 0 \\ -\frac{\dot{y}_p}{\dot{v}_p \cos \dot{\phi}_p} & , \text{else} \end{cases} \tag{3.11}$$

3.5.2 *Relative Transformation*

The normalized RSM from 3.5.1 can be transformed into a relative RSM where the the current position of road user c is at the origin of the RSM coordinate system. In other words, a relative RSM models a scenario from the frame of reference of road user c. The idea of using a relative RSM is to further reduce the amount of variables in eq. (3.4) for collision detection. This is achieved by setting the relative (sign:˜) speed $\tilde{v}_c$ of road user c to $\tilde{v}_c = 0$, see Fig. 3.5. Note that since $\tilde{v}_c = 0$, there is no green arrow for road user c in Fig. 3.5, i.e. the road user c "stays" at the origin and the whole movement is projected to road user p. Accordingly, the actual speed v_p and direction ϕ_p of road user p now has to be converted for the relative movement vector $\tilde{\mathbf{m}}_p = (\dot{\mathbf{r}}_p, \tilde{\phi}, \tilde{v})$.

Since a speed v is in fact the magnitude of a velocity vector $\mathbf{v} = (v_x, v_y)$, i.e. $v = \|\mathbf{v}\| = \sqrt{v_x^2 + v_y^2}$, this relation can be used to derive the relative speed of p as follows.

Let $\mathbf{v}_p = (v_{p_x}, v_{p_y})^T$ be the original velocity of road user p and $\mathbf{v}_c = (v_{c_x}, v_{c_y})^T$ be the original velocity of road user c in a normalized RSM. The components of $\mathbf{v}_i$ ($i \in \{p, c\}$) can be determined as given in eq. (3.12) and eq. (3.13)

$$v_{i_x} = v_i \cdot \sin \dot{\phi}_i \tag{3.12}$$

$$v_{i_y} = v_i \cdot \cos \dot{\phi}_i \tag{3.13}$$

To obtain the relative velocity $\tilde{\mathbf{v}}$ of p, eq. (3.14) is used

$$\begin{aligned} \tilde{\mathbf{v}} &= \mathbf{v}_p - \mathbf{v}_c \\ &= \left(v_{p_x} - v_{c_x},\ v_{p_y} - v_{c_y}\right)^T \\ &= \left(v_p \cdot \sin \dot{\phi}_p - v_c \cdot \sin \dot{\phi}_c,\ v_p \cdot \cos \dot{\phi}_p - v_c \cdot \cos \dot{\phi}_c\right)^T \\ &= (\tilde{v}_x,\ \tilde{v}_y)^T \end{aligned} \tag{3.14}$$

Inserting eq. (3.14) in eq. (3.15) then gives the relative speed $\tilde{v}$ of road user p in a relative RSM.

$$\tilde{v} = \sqrt{\tilde{v}_x^2 + \tilde{v}_y^2} \tag{3.15}$$

To derive the relative direction $\tilde{\phi}$ of p, the angle between $\tilde{\mathbf{v}}$ and the normalized y-axis vector $\mathbf{e}_2 = (0, 1)$ can be determined, as seen in (3.16).

$$\begin{aligned} \tilde{\phi} &= \operatorname{atan2}\left(\mathbf{e}_{2_x}, \mathbf{e}_{2_y}\right) - \operatorname{atan2}\left(\tilde{v}_y, \tilde{v}_x\right) \\ &= \pm \arccos\left(\frac{\tilde{\mathbf{v}}\mathbf{e}_2}{\|\tilde{\mathbf{v}}\|\,\|\mathbf{e}_2\|}\right) \\ &= \pm \arccos\left(\frac{\tilde{v}_x}{\sqrt{\tilde{v}_x^2 + \tilde{v}_y^2}}\right) \end{aligned} \tag{3.16}$$

Compared to the normalized RSM transformation, the relative RSM transformation has the advantage of overall less terms in the equations for TTC calculation due to $\dot{x}_c = 0$, $\dot{y}_c = 0$ but also $\tilde{v}_c = 0$.

$$\begin{aligned} \tilde{\mathbf{r}}_c(t) &= \tilde{v} \cdot t \begin{pmatrix} \sin \tilde{\phi}_p \\ \cos \tilde{\phi}_p \end{pmatrix} + \begin{pmatrix} \dot{x}_p \\ \dot{y}_p \end{pmatrix} \\ \iff \begin{pmatrix} 0 \\ 0 \end{pmatrix} &= \tilde{v} \cdot t \begin{pmatrix} \sin \tilde{\phi}_p \\ \cos \tilde{\phi}_p \end{pmatrix} + \begin{pmatrix} \dot{x}_p \\ \dot{y}_p \end{pmatrix} \end{aligned} \tag{3.17}$$

After rearranging (3.17), t_x and t_y can be obtained and shortened using $\tilde{v}_c = 0$ and $\dot{x}_c = 0$, $\dot{x}_y = 0$, like seen in (3.18) and (3.19). Since for $\tilde{\phi} = 0°$ and $\tilde{\phi} = 180°$, $\sin\tilde{\phi} = 0$ $t_x$needs to be evaluated according to (3.18):

$$\Rightarrow t_x = \begin{cases} 0 & , \text{if } \tilde{\phi} \in \{0°, 180°\} \wedge \dot{x}_p = 0 \\ \infty & , \text{if } \tilde{\phi} \in \{0°, 180°\} \wedge \dot{x}_p \neq 0 \\ -\frac{\dot{x}_p}{\tilde{v}\sin\tilde{\phi}} & , else \end{cases} \tag{3.18}$$

Similar, since for $\tilde{\phi} = 90°$ and $\tilde{\phi} = 270°$, $\cos\tilde{\phi} = 0$ t_y needs to be evaluated according to (3.19):

$$\Rightarrow t_y = \begin{cases} 0 & , \text{if } \tilde{\phi} \in \{90°, 270°\} \wedge \dot{y}_p = 0 \\ \infty & , \text{if } \tilde{\phi} \in \{90°, 270°\} \wedge \dot{y}_p \neq 0 \\ -\frac{\dot{y}_p}{\tilde{v}\cos\tilde{\phi}} & , else \end{cases} \tag{3.19}$$

For the basic point-to-point RSM, the advantage of the relative approach compared to the normalized approach is barely noticeable since the difference is only the term v_c in t_x. The difference starts to be noticeable the more complex the geometries for the road users become. When geometries for the road users are considered in collision detection, the complexity and ultimately the amount of terms in an equation is higher. In order to reduce complexity of equations, an approach called "subdivision and reconstruction" is introduced in the next section.

3.6 SUBDIVISION AND RECONSTRUCTION

In this section, an approach called Subdivision and Reconstruction (SubRec) is introduced which is based on the "divide and conquer" paradigm in computer science for implementing efficient algorithms. Similarly to "divide and conquer", the SubRec approach can be used on a RSM with road users, that have geometries, so that the RSM and the geometries are subdivided it into multiple partial RSMs with simpler geometries.

Using SubRec allows parallelization for the precise collision detection algorithms when geometries are considered for the road users and thus can reduce the total time for collision detection. The SubRec approach can be applied for both, normalized RSM and relative RSM transformations.

In Fig. 3.6, the SubRec approach in form of a graph for an abstract example is shown: Let there be a RSM in which the road users c and p are represented by geometries G_c and G_p. In the example in Fig. 3.6, G_c consists of geometry parts g_1, g_2 and g_3 whereas G_p consists of

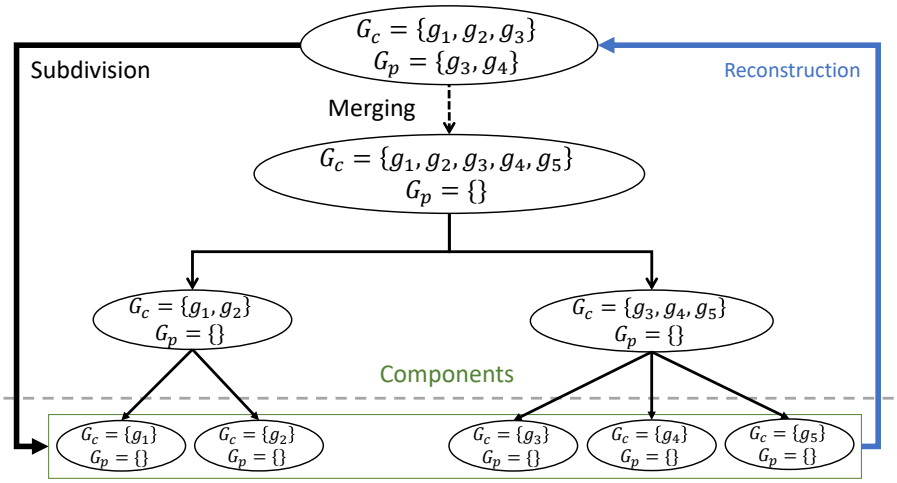


Fig. 3.6: Subdivision and Reconstruction

geometry parts g_4 and g_5. SubRec consists of the following two steps: "Subdivision" and "Reconstruction". In the "Subdivision" step, in case that each road user has a geometry, the geometry parts of G_p can be merged into G_c in order to construct a new expanded geometry for c. The road user p becomes a point, i.e. $G_p = \emptyset$. Then, the merged geometries set G_c is used to subdivide the "components" in G_c-to-point RSMs, $G_c \in \{P, R, C\}$ ($P = Point$, $R = Rectangle$, $C = Circle$). in which c has a geometry such as a rectangle or a circle, or a point. In the example in Fig. 3.6, the merged geometries set G_c is subdivided into five components, i.e. g_1-to-point to g_5-to-point RSMs. The collision detection algorithms for G_c-to-point RSMs require a lower amount of parameters and variables and thus the calculation is less complex than in the original RSM. In the "Reconstruction" step, the results of each G_c-to-point RSMs are merged to obtain the TTC.

3.6.1 *General Subdivision and Reconstruction Algorithm*

The general SubRec algorithm for precise collision detection has two road users c and p as input whereby c's geometry can be subdivided into four points or circles g_i, $i \in \{1, 2, 3, 4\}$. The parameters speed and direction for g_i are taken from the road user c. Once the i normalized or relative RSMs for (g_i, p) are created, the precise collision detection algorithms for the g_i-to-point RSMs, detailed in the following Sections 3.7 to 3.9, are used to calculate the g_i-to-point ttc_{g_i}. The TTC for the original G_c-to-G_p RSM is then calculated by determining the minimum of ttc_{g_i}, see algorithm 2.

Algorithm 2: SubRec for TTC calculation using normalized or relative RSMs

input : (c, p, G_c)
output: TTC

```
foreach g_i ∈ G_c do
    RSM ← create {normalized | relative} RSM for (g_i, p);
    ttc_{g_i} ← RSM.getTTC(g_i, p);
end
TTC ← min(ttc_{g_i});
```

3.7 RECTANGLE-TO-POINT/POINT-TO-RECTANGLE RSM

A rectangle-to-point RSM consists of the road user c, modelled as a rectangle, and p, modelled as a point. In this section, it is detailed how the TTC in a rectangle-to-point RSM can be calculated using the normalized or relative RSM transformation. The rectangle-to-point RSM is the first of two essential basic components to for the SubRec algorithm to handle rectangle-to-rectangle and rectangle-to-circle/circle-to-rectangle RSMs.

3.7.1 *Using the Normalized Transformation*

Consider a normalized rectangle-to-point RSM, like shown in Fig. 3.4. The rectangle geometry of c consists of four edges that are defined by two given vertices. Since the RSM is normalized, c's rectangle is rotated in direction 90° which has the advantage that the rectangle's edges are parallel to the x-axis and the y-axis, respectively, see Fig. 3.4. Let w_c be the width and l_c the length of the rectangle representing road user c . Using w_c and l_c, the positions of the vertices A to D can be determined, according to eqs. (3.20) to (3.23).

$$A = \left(\dot{x}_c - \frac{l_c}{2}, \dot{y}_c - \frac{w_c}{2}\right) = \left(-\frac{l_c}{2}, -\frac{w_c}{2}\right) \tag{3.20}$$

$$B = \left(\dot{x}_c - \frac{l_c}{2}, \dot{y}_c + \frac{w_c}{2}\right) = \left(-\frac{l_c}{2}, \frac{w_c}{2}\right) \tag{3.21}$$

$$C = \left(\dot{x}_c + \frac{l_c}{2}, \dot{y}_c + \frac{w_c}{2}\right) = \left(\frac{l_c}{2}, \frac{w_c}{2}\right) \tag{3.22}$$

$$D = \left(\dot{x}_c + \frac{l_c}{2}, \dot{y}_c - \frac{w_c}{2}\right) = \left(\frac{l_c}{2}, -\frac{w_c}{2}\right) \tag{3.23}$$

To determine the TTC for a rectangle-to-point RSM, the normalized point-to-point collision detection formula from eq. (3.6) can be used

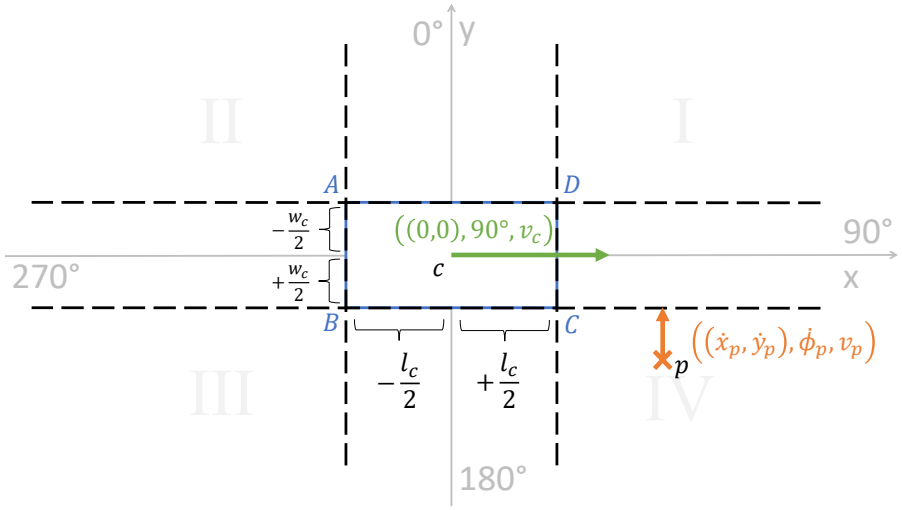


Fig. 3.7: Axis separation of the rectangle geometry in a normalized rectangle-to-point RSM

by replacing the position of c with the vertices A to D, see eqs. (3.24) and (3.25), to obtain the times $t_{x_{1,2}}$ and $t_{y_{1,2}}$.

$$\Rightarrow t_{x_{1,2}} = \frac{\dot{x}_p - \dot{x}_c \pm \frac{l_c}{2}}{v_c \sin\dot{\phi}_c - v_p \sin\dot{\phi}_p} = \frac{\dot{x}_p \pm \frac{l_c}{2}}{v_c - v_p \sin\dot{\phi}_p} \tag{3.24}$$

$$\Rightarrow t_{y_{1,2}} = -\frac{\dot{y}_p - \dot{y}_c \pm \frac{w_c}{2}}{v_c \cos\dot{\phi}_c v_p \cos\dot{\phi}_p} = -\frac{\dot{y}_p \pm \frac{w_c}{2}}{v_p \cos\dot{\phi}_p} \tag{3.25}$$

The $t_{x_{1,2}}$ indicate the time interval TX between t_{x_1} and t_{x_2} for which the rectangle area of the geometry of road user c covers the x-position of road user p, i.e. $x_c - \frac{l_c}{2} \leqslant x_p \leqslant x_c + \frac{l_c}{2}$. Likewise the time interval TY between t_{y_1} and t_{y_2} indicate the times for when the rectangle area of the geometry of road user c covers the y-position of road user p, i.e. $y_c - \frac{w_c}{2} \leqslant y_p \leqslant y_c + \frac{w_c}{2}$. Thus, the point of road user p is "inside" the area of the rectangle geometry of road user c when there is an intersection between TX and TY, i.e. $TX \cap TY \neq \emptyset$. In Section 3.7.3 an algorithm to get the TTC from TX and TY will be detailed.

3.7.2 *Using the Relative Transformation*

Consider a relative RSM with c, represented as a rectangle, and p, represented as a point, as shown in Fig. 3.8. Since $\dot{x}_c = 0$, $\dot{y}_c = 0$, $\tilde{v}_c = 0$, in a relative RSM, $t_{x_{1,2}}$ can be further shortened, as shown shown in eq. (3.26) and eq. (3.27).

$$\Rightarrow t_{x_{1,2}} = \frac{\dot{x}_p - \dot{x}_c \pm \frac{l_c}{2}}{(v_c \sin\phi_c - v_p \sin\phi_p)} = -\frac{\dot{x}_p \pm \frac{l_c}{2}}{\tilde{v}\sin\tilde{\phi}} \tag{3.26}$$

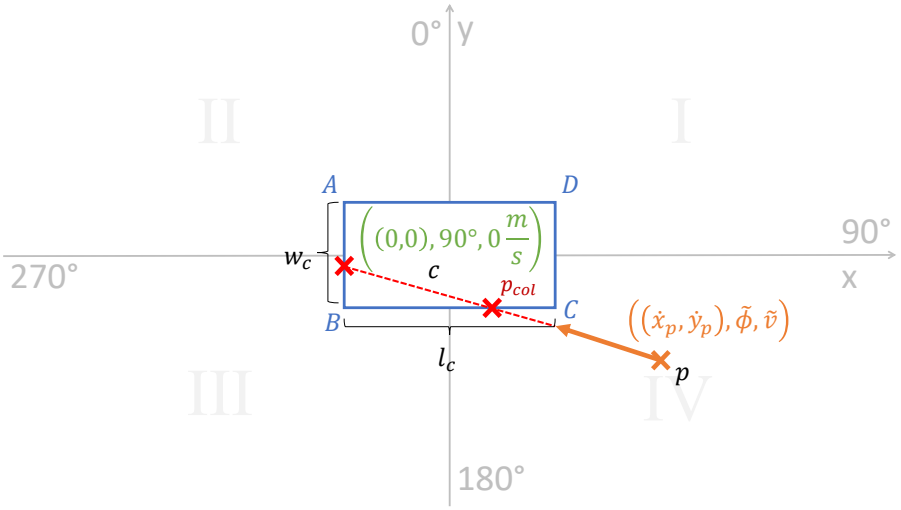


Fig. 3.8: Relative rectangle-to-point RSM

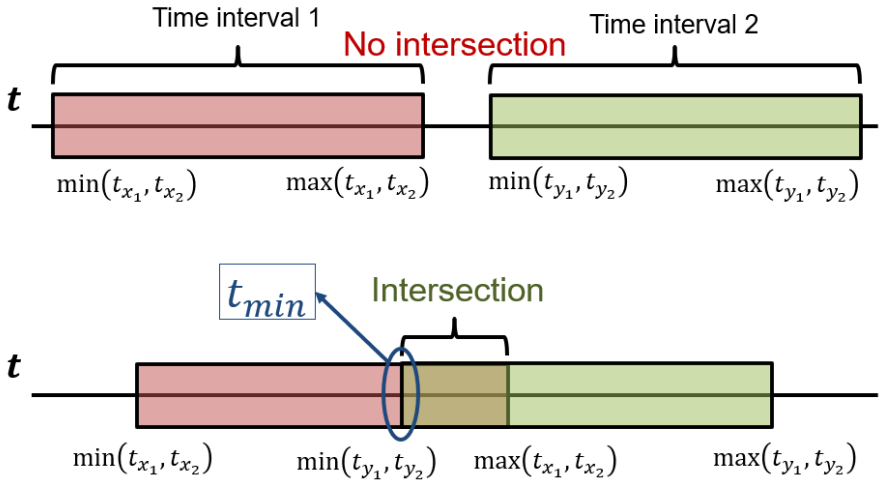


Fig. 3.9: Extract TTC from $t_{x_{1,2}}$ and $t_{y_{1,2}}$ time intervals

$$\Rightarrow t_{y_{1,2}} = \frac{\dot{y}_p - \dot{y}_c \pm \frac{w_c}{2}}{(v_c \cos\phi_c - v_p \cos\phi_p)} = -\frac{\dot{y}_p \pm \frac{w_c}{2}}{\tilde{v}\cos\tilde{\phi}} \tag{3.27}$$

Like for the normalized approach the $t_{x_{1,2}}$and $t_{y_{1,2}}$ are used to determine the TTC as detailed in the next Section 3.7.3.

3.7.3 *TTC Algorithm for rectangle-to-point components*

The both solutions $t_{x_{1,2}}$ and $t_{y_{1,2}}$ from eqs. (3.24) to (3.26), form time intervals which indicate the times at which both road users are at the same x-positions and when both road users are at the same y-position, respectively. Thus, neither $t_{x_{1,2}}$ nor $t_{y_{1,2}}$ alone indicate if there will be a tangential point between the road user's geometries, i.e. a collision between the road users is impending.

To obtain the TTC from $t_{x_{1,2}}$ and $t_{y_{1,2}}$, the algorithm 3 is used. The algorithm uses the two intervals TX and TY, given by $t_{x_{1,2}}$ and $t_{y_{1,2}}$, and determines if there is an intersection $TX \cap TY \neq \emptyset$. The intersection interval $TX \cap TY$ can be interpreted as the time for when the road user p "moves through" the geometry of road user c. For this reason, the TTC, i.e. the time at which road user p is at the edge of the geometry of road user c, is the beginning of the the interval $TX \cap TY$, compare Fig. 3.9.

Algorithm 3: Get TTC from intersection times $t_{x_{1,2}}$, $t_{y_{1,2}}$

input : $t_{x_{1,2}}$, $t_{y_{1,2}}$
output: TTC
1 $TTC \leftarrow \mathtt{null}$;
2 $TX \leftarrow [\min(t_{x_1}, t_{x_2}), \max(t_{x_1}, t_{x_2})]$;
3 $TY \leftarrow [\min(t_{y_1}, t_{y_2}), \max(t_{y_1}, t_{y_2})]$;
4 **if** $TX \cap TY \neq \emptyset$ **then**
5 $\quad TTC \leftarrow \min(TX \cap TY)$;
6 **end**

3.8 CIRCLE-TO-POINT/POINT-TO-CIRCLE RSM

A circle-to-point RSM consists of the road user c, modelled as a circle, and p, modelled as a point. In this section, it is detailed how the TTC in a circle-to-point RSM can be calculated using the normalized or relative RSM transformation. Like the previous rectangle-to-point RSM, circle-to-point RSM is the second essential basic component to for the SubRec algorithm to handle rectangle-to-rectangle and rectangle-to-circle/circle-to-rectangle RSMs.

3.8.1 *Using the Normalized Transformation*

Consider a normalized circle-to-point RSM, as shown in Fig. 3.10. In such a circle-to-point RSM the point of road user p intersects with the circle geometry, which has a radius r, of road user c when the euclidean distance between the both road users position is $\leqslant r$. In accordance to the definition that a collision is "detected" if there is a tangential point between the road users' geometries the euclidean distance eq. (3.28) is used to indicate whether the point of road user p is "at" the circle of road user c.

$$r^2 = (\dot{x}_c - \dot{x}_p)^2 + (\dot{y}_c - \dot{y}_p)^2 \tag{3.28}$$

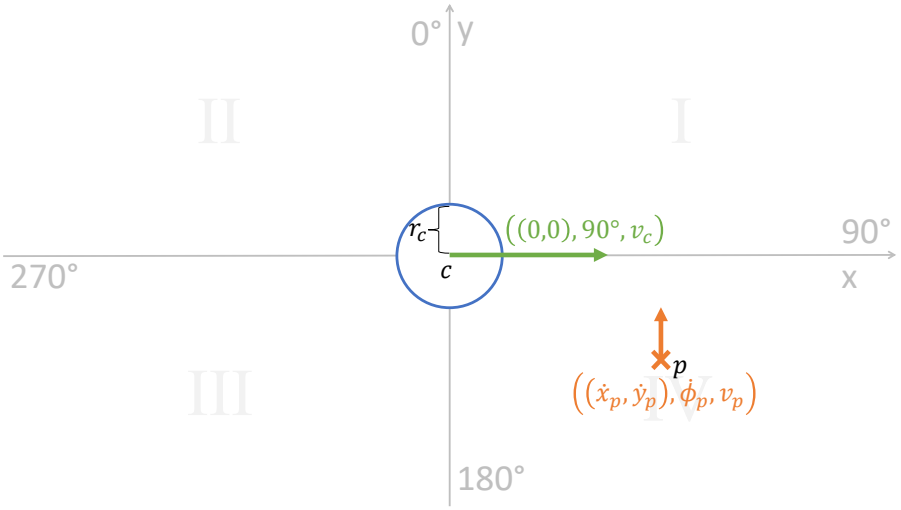


Fig. 3.10: Normalized circle-to-point scenario

Inserting the movement equations $\mathbf{r}_i(t)$ from eq. (3.2) for both road users in eq. (3.28) results in eq. (3.29).

$$\begin{aligned} r^2 = \quad & \left((v_c t \sin\dot{\phi}_c + \dot{x}_c) - (v_p t \sin\dot{\phi}_p + \dot{x}_p)\right)^2 \\ & + \left((v_c t \cos\dot{\phi}_c + \dot{y}_c) - (v_p t \cos\dot{\phi}_p + \dot{y}_p)\right)^2 \\ \iff 0 = t^2 \Bigg(& \left(v_c \sin\dot{\phi}_c - v_p \sin\dot{\phi}_p\right)^2 \\ & + \left(v_c \sin\dot{\phi}_c - v_p \sin\dot{\phi}_p\right)^2 \Bigg) \\ + 2t \Bigg(& \left(\dot{x}_c - \dot{x}_p\right)\left(v_c \sin\dot{\phi}_c - v_p \sin\dot{\phi}_p\right) \\ & + \left(\dot{y}_c - \dot{y}_p\right)\left(v_c \cos\dot{\phi}_c - v_p \cos\dot{\phi}_p\right) \Bigg) \\ + \quad & \left(\dot{x}_p - \dot{x}_c\right)^2 + \left(\dot{y}_p - \dot{y}_c\right)^2 - r^2 \end{aligned} \tag{3.29}$$

By normalizing and rearranging eq. (3.29), $\frac{p}{2}$ and q, see eq. (3.30)

$$\begin{aligned} \Rightarrow \frac{p}{2} &= \frac{\left(\dot{x}_c - \dot{x}_p\right)\left(v_c \sin\dot{\phi}_c - v_p \sin\dot{\phi}_p\right)}{\left(v_c \sin\dot{\phi}_c - v_p \sin\dot{\phi}_p\right)^2 + \left(v_c \cos\dot{\phi}_c - v_p \cos\dot{\phi}_p\right)^2} \\ &\quad + \frac{\left(\dot{y}_c - \dot{y}_p\right)\left(v_c \cos\dot{\phi}_c - v_p \cos\dot{\phi}_p\right)}{\left(v_c \sin\dot{\phi}_c - v_p \sin\dot{\phi}_p\right)^2 + \left(v_c \cos\dot{\phi}_c - v_p \cos\dot{\phi}_p\right)^2} \\ &= \frac{-\dot{x}_p\left(v_c - v_p \sin\dot{\phi}_p\right) + \dot{y}_p v_p \cos\dot{\phi}_p}{v_c^2 + v_p^2 - 2v_c v_p \sin\dot{\phi}_p} \\ q &= \frac{\left(\dot{x}_p - \dot{x}_c\right)^2 + \left(\dot{y}_p - \dot{y}_c\right)^2 - r^2}{\left(v_c \sin\dot{\phi}_c - v_p \sin\dot{\phi}_p\right)^2 + \left(v_c \cos\dot{\phi}_c - v_p \cos\dot{\phi}_p\right)^2} \\ &= \frac{\dot{x}_p^2 + \dot{y}_p^2 - r^2}{v_c^2 + v_p^2 - 2v_c v_p \sin\dot{\phi}_p} \end{aligned} \tag{3.30}$$

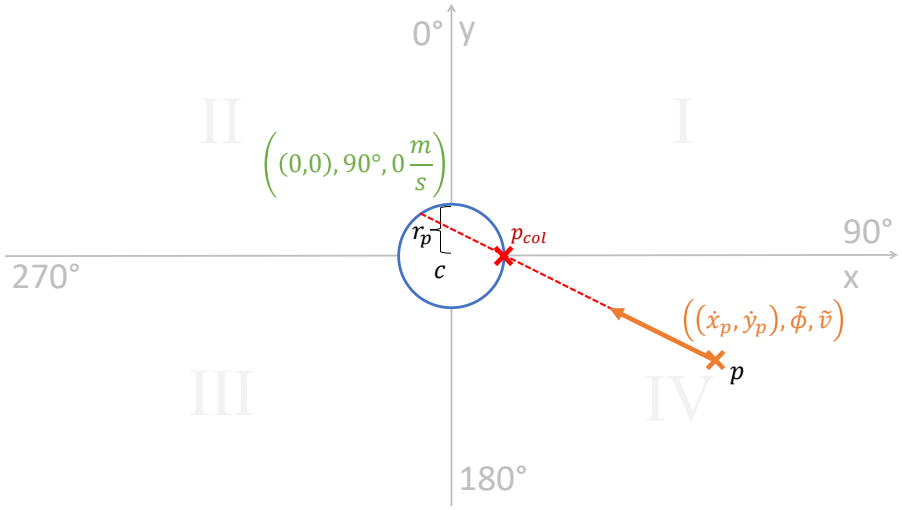


Fig. 3.11: Relative circle-to-point RSM

can be determined for the pq-formula to obtain $t_{1,2}$, see eq. (3.31).

$$\begin{aligned}
\Rightarrow t_{1,2} &= -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q} \\
&= -\frac{-\dot{x}_p \left(v_c - v_p \sin\dot{\phi}_p\right) + \dot{y}_p v_p \cos\dot{\phi}_p}{v_c^2 + v_p^2 - 2v_c v_p \sin\dot{\phi}_p} \\
&\pm \left(\left(\frac{-\dot{x}_p \left(v_c - v_p \sin\dot{\phi}_p\right) + \dot{y}_p v_p \cos\dot{\phi}_p}{v_c^2 + v_p^2 - 2v_c v_p \sin\dot{\phi}_p} \right) \right. \\
&\left. - \frac{\dot{x}_p^2 + \dot{y}_p^2 - r^2}{v_c^2 + v_p^2 - 2v_c v_p \sin\dot{\phi}_p} \right)^{\frac{1}{2}}
\end{aligned} \tag{3.31}$$

In eq. (3.31), the radius r is not related to a specific road user, i.e., it is not specified which road user has the circle geometry. Thus, the solution can be interpreted for both a circle-to-point or a point-to-circle RSM.

3.8.2 *Using the Relative Transformation*

Consider a relative circle-to-point RSM, like shown in Fig. 3.11. It can be seen that there are up to two intersection points between the movement trajectory of p and the circle geometry of c. The intersection point with the lowest time value t is the collision point p_{col}, i.e. this time t is the TTC.

Like in eq. (3.29), x_i and y_i, $i \in \{c, p\}$, from the distance equation eq. (3.28) can be substituted with the linear movement $\mathbf{r}_i(t)$ of a relative RSM, respectively. Since in a relative RSM $\dot{x}_c = 0$, $\dot{y}_c = 0$, $\tilde{v}_c = 0$, eq. (3.29) simplifies to eq. (3.32)

$$r^2 = \left(-\tilde{v} \cdot t \cdot \sin\tilde{\phi} + \dot{x}_p\right)^2 + \left(-\tilde{v} \cdot t \cdot \cos\tilde{\phi} + \dot{y}_p\right)^2 \quad (3.32)$$

$$\iff 0 = t^2 + 2t\frac{\left(\dot{x}_p \sin\tilde{\phi} + \dot{y}_p \cos\tilde{\phi}\right)}{\tilde{v}} + \frac{\dot{x}_p^2 + \dot{y}_p^2 - r^2}{\tilde{v}^2} \quad (3.33)$$

eq. (3.32) can be rearranged to eq. (3.33) to obtain the parameters $\frac{p}{2}$ and q for the p-q formulae, see eq. (3.34)

$$\Rightarrow \frac{p}{2} = \frac{\dot{x}_p \sin\tilde{\phi} + \dot{y}_p \cos\tilde{\phi}}{\tilde{v}}, q = \frac{\dot{x}_p^2 + \dot{y}_p^2 - r^2}{\tilde{v}^2} \quad (3.34)$$

and thus the times $t_{1,2}$ for both intersection points can be obtained, see eq. (3.35).

$$\begin{aligned} \Rightarrow t_{1,2} = & -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q} \\ = & -\frac{\dot{x}_p \sin\tilde{\phi} + \dot{y}_p \cos\tilde{\phi}}{\tilde{v}} \\ & \pm \sqrt{\frac{r^2 - \left(\dot{x}_p \cos\tilde{\phi} - \dot{y}_p \sin\tilde{\phi}\right)^2}{\tilde{v}^2}} \end{aligned} \quad (3.35)$$

To get the correct TTC from t_1 and t_2, the minimum of t_1 and t_2 is determined, i.e TTC $= \min(t_1, t_2)$. With the circle-to-point RSM component form this section and the rectangle-to-point RSM component from the previous section, it is possible to reconstruct a rectangle-to-circle RSM, as shown in the following section.

3.9 RECTANGLE-TO-CIRCLE/CIRCLE-TO-RECTANGLE RSM

In the preceding Sections 3.7, 3.7.3 and 3.8, it was detailed how the TTC for the rectangle-to-point and circle-to-point RSM components can be calculated. In order to obtain the TTC between c and p in a circle-to-rectangle RSM, it is possible to use the rectangle-to-point and circle-to-point components to reconstruct a solution for rectangle-to-circle and circle-to-rectangle RSM using the SubRec algorithm.

3.9.1 *Using Normalized/Relative Transformation*

Consider a normalized rectangle-to-circle RSM, see Fig. 3.12, and a relative scenario, see Fig. 3.13, respectively, with c represented as a rectangle and p as a circle. The approach to determine the TTC in a rectangle-to-circle RSM is to construct a geometry that results from surrounding the rectangle geometry of c with the circle of p, see Fig. 3.14.

The resulting RSM has a rectangle for c that is expanded by the circle radius r_p and p as a point, as shown in Fig. 3.15a for the normalized and in Fig. 3.15a for the relative RSM. The SubRec algorithm is

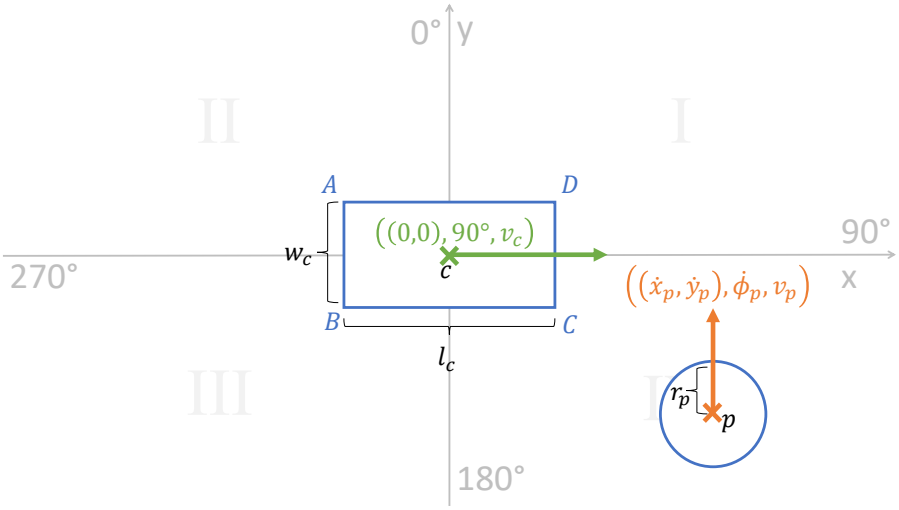


Fig. 3.12: Normalized rectangle-to-circle RSM

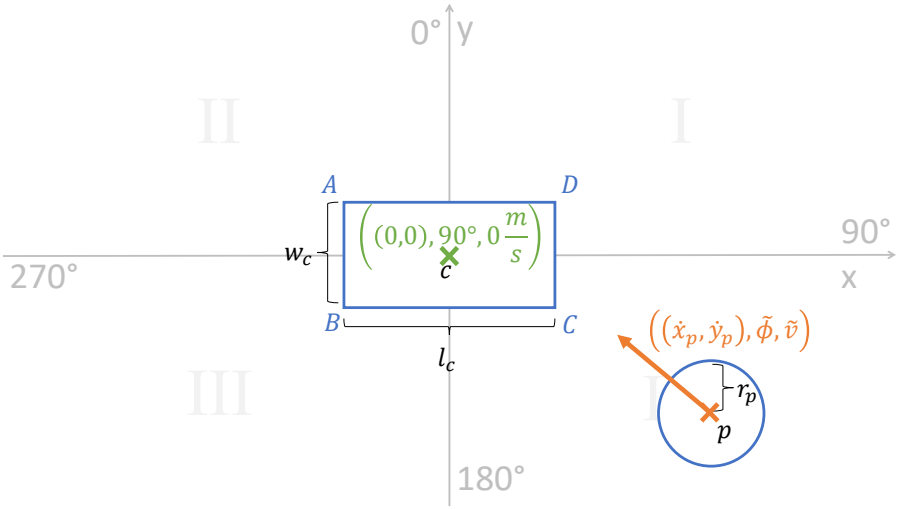


Fig. 3.13: Relative rectangle-to-circle RSM

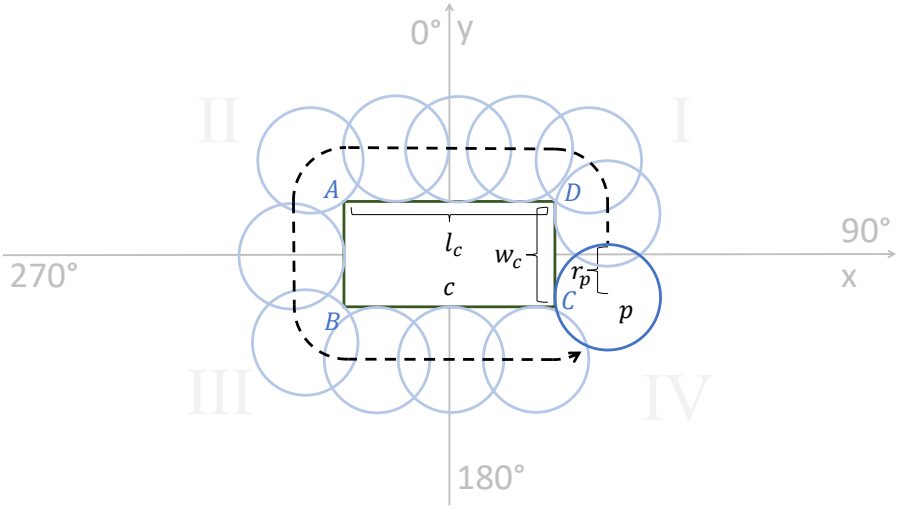
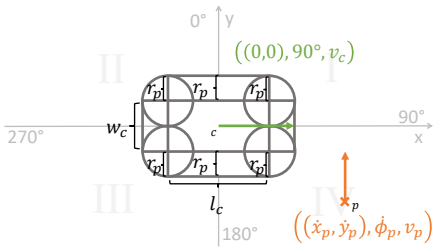
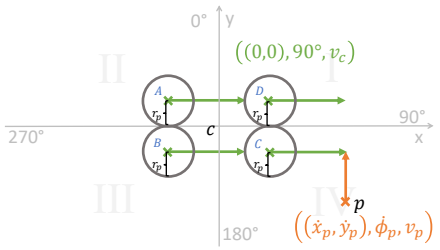


Fig. 3.14: Expansion of c's rectangle geometry by the radius r_p

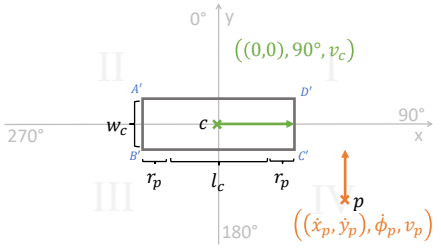
then used to subdivide the resulting expanded geometry in circle-to-point and rectangle-to-point RSM components, see Figs. 3.15 and 3.16, respectively.



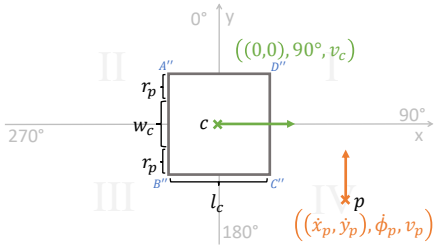
(a) Expanded rectangle-to-circle RSM



(b) Subdividing c's expanded geometry into circles A, B, C and D with radius r_p



(c) Expanding l_c of c's rectangle geometry by $2r_p$



(d) Expanding w_c of c's rectangle geometry by $2r_p$

Fig. 3.15: Subdivision and reconstruction of a normalized rectangle-to-circle RSM

The first subdivision step is to subdivide the expanded geometry into a RSM of four circles with radius r_p that are located at the vertices positions (x_A, y_A), (x_B, y_B), (x_C, y_C), (x_D, y_D) of c's rectangle geometry, see Figs. 3.15b and 3.16b for normalized and relative RSM transformation, respectively. Using the SubRec algorithm, see algorithm 2 in Section 3.6, with the input (c, p, i_c), $i \in \{A, B, C, D\}$, the minimum intersection time t_1 for the circle-to-point component is determined.

The second subdivision step, shown in Figs. 3.15c and 3.16c, is to subdivide the expanded geometry into a rectangle of which the length is expanded by $2r_p$. By using (c, p) as input for the algorithm 4, t_2 for the rectangle-to-point component can be determined.

The third and last subdivision step, shown in Figs. 3.15d and 3.16d, is to subdivide the expanded geometry into a rectangle of which the length is expanded by $2r_p$. The time t_3 is determined in the same way as in the second step using algorithm 5.

The reconstruction step for the rectangle-to-circle RSM consists of determining the actual TTC as the minimum from t_1, t_2 and t_3, i.e. $\text{TTC} = \min(t_1, t_2, t_3)$. Since rectangle-to-circle and circle-to-rectangle result in the same TTC, a solution for circle-to-rectangle RSM is acquired as well.

Algorithm 4: Get t_2 from expanded length rectangle-to-point RSM component

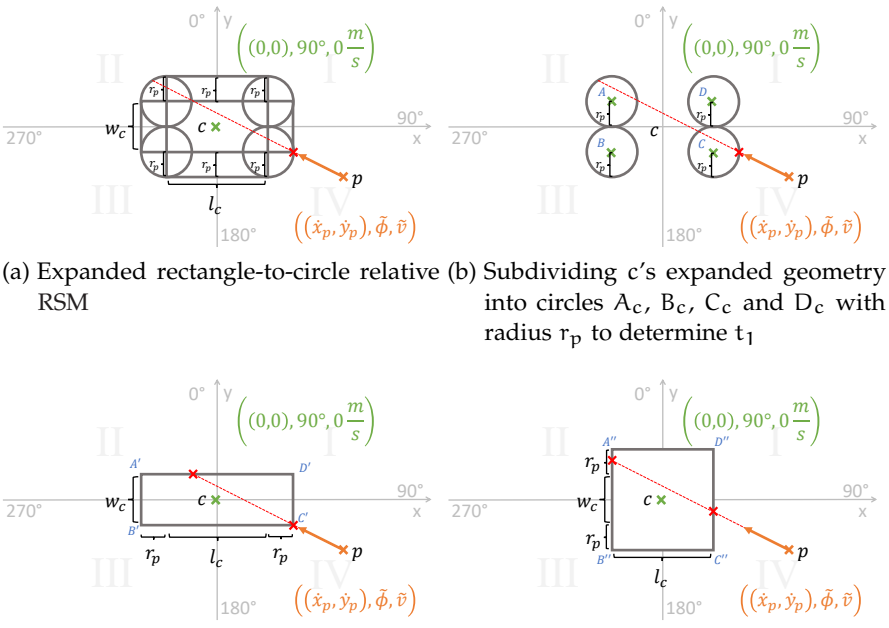
input : c, p
output: t_2

$t_2 \leftarrow$ null;
$l_c \leftarrow$ length of the rectangle geometry of road user c;
$l_c \leftarrow l_c + 2r_p$;
$t_2 \leftarrow$ get TTC from normalized/relative rectangle-to-point RSM component (Section 3.7)

Algorithm 5: Get t_3 from expanded width rectangle-to-point RSM component

input : c, p
output: t_3

$t_3 \leftarrow$ null;
$w_c \leftarrow$ width of the rectangle geometry of road user c;
$w_c \leftarrow w_c + 2r_p$;
$t_3 \leftarrow$ get TTC from normalized/relative rectangle-to-point RSM component (Section 3.7)



(a) Expanded rectangle-to-circle relative RSM

(b) Subdividing c's expanded geometry into circles A_c, B_c, C_c and D_c with radius r_p to determine t_1

(c) Expanding l_c of c's rectangle geometry by $2r_p$ to determine t_2

(d) Expanding w_c of c's rectangle geometry by $2r_p$ to determine t_3

Fig. 3.16: Subdivision and reconstruction of a relative rectangle-to-circle scenario

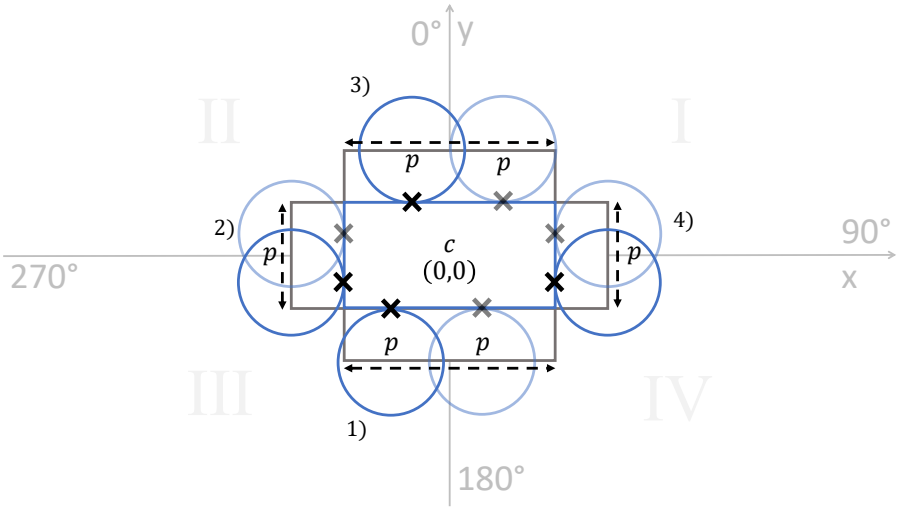


Fig. 3.17: Determining the collision point for the rectangle sides

3.9.2 *Collision Point Calculation*

To determine the collision point between the rectangle and circle geometry it is first determined if the calculated TTC is equal to t_1 for one of the vertices from Figs. 3.15b and 3.16b. In this case the collision position is equivalent to the future position determined by t_1 of this vertex. Otherwise, the circle collides with one of the edges of the rectangle instead. In this case the collision point can be calculated using the knowledge that it always lies on the girth of the circle in 1) 0°, 2) 90°, 3) 180° or 4) 270° positions along the corresponding rectangle side, as shown in Fig. 3.17. The benefit of the normalized and relative scenario transformation is that the rectangle geometry is located at the origin of the coordinate system and aligned to the x- and y-axis. Therefore, it simply has to be determined which expanded rectangle side corresponds to the future position of the point geometry to obtain the collision point.

REFERENCES

[3.1] T. Vincenty, 'Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations', *Survey review*, vol. 23, no. 176, pp. 88–93, Apr. 1975 (cit. on p. 32).

4

COLLISION DETECTION PERFORMANCE

The task for a VRU Collision Avoidance System (VRU CAS) is to determine if there is an impending collision in order to give a warning to the vehicle and, if possible also to the Vulnerable Road User (VRU). Accordingly, the performance of a VRU CAS can be interpreted as the "level correctness" of the events "collision detected" (cd) and "no collision detected" (ncd). In the most simple case, i.e. using a binary collision detection as proposed in Chapter 3, this "level of correctness" is equivalent to the "performance" of a binary classifier, i.e. if the cd event is correct for a given scenario. Depending on the scenario, there are four possibilities for the outcome of a VRU CAS: First, a True Positive (TP) cd event, i.e. a collision will happen and the systems detects the collision. Second, a False Positive (FP) cd event, i.e. no collision will happen but the system detects one. Third a True Negative (TN) ncd event, i.e. there will be no collision and the system appropriately does not indicate one. Last, a False Negative (FN) ncd event, i.e. there will be a collision but the system does not detect it.

4.1 FACTORS INFLUENCING THE COLLISION DETECTION PERFORMANCE

There are different factors which influence the correctness of the outcome of a VRU CAS. Besides primary system architecture dependent aspects, like recognition, transmission, and calculation delays, or Field of View (FOV) length and angle, one major factor is, that there will always be some amount of random error for the recognition of the road user's current movement. Even though, different sensor systems might have quite different magnitudes for movement recognition errors, such errors are physical unavoidable. In addition, the accuracy of most sensor system can not be assumed to be constant since the accuracy often depends on environmental and contextual circumstances. For example vision based sensor system like video, radar, or Light Detection and Ranging (LIDAR) require a Line-of-Sight (LOS) and might be influenced by the visibility conditions, e.g. due to rain or fog. Today's Global Navigation Satellite Systems (GNSS) as used by current smartphones suffers from the challenges of pseudoranges multipath, atmospheric delays, frequent cycle slips and losses of phase lock [4.1]. The same arguments holds also for recognizing the speed and direction of the road users.

Therefore a probabilistic notation for the outcome of the collision detection should be used. In this dissertation, $P(TP)$ denotes the prob-

ability that an impending collision is detected and $P(FN)$ the probability that an impending collision is not detected. Likewise, $P(TN)$ gives the probability that the system correctly indicates that there will not be a collision and $P(FP)$ gives the probability that a collision is indicated even though there is no impending collision.

4.2 COLLISION AND NO-COLLISION SCENARIOS

By definition, TP and FN events can only occur in a scenario in which a collision will actually happen. Likewise, the TN and FP events can only occur in scenarios in which no collision will happen. Therefore it is convenient to use the probability that the system detects a collision regardless of the scenario with is denoted as the *probability of collision detection* $P(cd)$. In order to be able to evaluate the collision detection performance of a VRU CAS the following definitions based on the $P(cd)$ notation for collision and no-collision scenarios are used

4.2.1 *Collision Scenario*

A collision scenario is defined as a scenario in which there will be a collision between the two road users, if, and only if, both road users constantly move according to the linear movement eq. (3.2), i.e. the collision will happen when non of the road users change their speed or direction (no rotation, no acceleration). The following relations hold for a collision scenario:

$$\begin{aligned} P(TP) &= P(cd) && \geqslant 0 \\ P(FN) &= 1 - P(cd) && \geqslant 0 \\ P(TN) &= 0 \\ P(FP) &= 0 \end{aligned}$$

4.2.2 *No-Collision Scenario*

Likewise, a no-collision scenario is defined as a scenario in which there will be no collision between the two road users, if both road users constantly move according to the linear movement equation eq. (3.2), i.e. no collision will happen when non of the road users change their speed or direction (no rotation, no acceleration). The following relations hold for a no-collision scenario:

$$
\begin{aligned}
P(TP) &= 0 \\
P(FN) &= 0 \\
P(TN) &= 1 - P(cd) && \geqslant 0 \\
P(FP) &= P(cd) && \geqslant 0
\end{aligned}
$$

4.3 KEY PERFORMANCE INDICATORS

To contribute a solution for the 2nd challenge (C II) "How to quantify the collisions detection performance of a VRU CAS independent to a certain architecture?" from Section 1.2, in the following two Key Performance Indicators (KPIs) are proposed to quantify the collision detection performance of a VRU CAS. The KPI are intended to be general in such a way that they can be used to quantify the performance of any VRU CAS, or even any Collision Avoidance System (CAS), depending on arbitrary factors which have an impact on the outcome of the collision detection.

4.3.1 *Probability of a Missed Alarm (P_{MA}):*

The first indicator is the Probability of a Missed Alarm (P_{MA}) .P_{MA} gives the probability that an impending collision is not detected and no alarm is given. A Missed Alarm can only happen for collision scenarios. According to the above discussed possible outcomes of a VRU CAS, P_{MA} is defined as

$$P_{MA} = P(FN) = 1 - P(cd) \tag{4.1}$$

A Missed Alarm might have severe consequences, like a fatal accident. Therefore, it is important to keep P_{MA} as low as possible.

4.3.2 *Probability of a False Alarm (P_{FA}):*

The second indicator is the probability of a False Alarm (P_{FA}). P_{FA} gives the probability that the system gives an alarm even though there is no impending collision. In contrast to a Missed Alarm, a False Alarm can only happen in no-collision scenarios. Therefore, P_{FA} is defined as

$$P_{FA} = P(FP) = P(cd) \tag{4.2}$$

In any way, a system which frequently gives False Alarms will not be accepted by the user, but additionally a False Alarm might also have severe consequences, depending on the reaction of the system to avoid the (non existing) collision. Therefore, it is not enough for a VRU CAS to achieve a low P_{MA} rate, but that it is also essential to keep P_{FA} as low as possible.

4.4 P_{MA} AND P_{FA} DEPENDING ON MOVEMENT RECOGNITION ACCURACY

In this dissertation the KPIs P_{MA} and P_{FA} will be used to answer the research question "What is the minimum required accuracy for VRU movement recognition on mobiles devices for a certain collision detection performance of a *cooperative* VRU Collision Avoidance System (cooperative VRU CAS)?". For this, the definitions for P_{MA} and P_{FA} from eqs. (4.1) and (4.2) are specified to be depended on random errors of the VRU movement recognition. For simplicity and without loss of any generality, if not explicit stated otherwise, collisions between a car and a pedestrian are assumed. It is assumed that the pedestrian carries a smartphone which is capable to recognize its current movement consisting of current position, speed, and direction.

Using the collision detection algorithm as proposed in Chapter 3 an impending collision is assumed if there is a time t for which there is a tangential point between the pedestrian's geometry whose current position is given by $\mathbf{r}_p(t)$, with the movement vector $\mathbf{m}_p$ and the geometry of the car, whose position is defined by $\mathbf{r}_c(t)$ with the movement vector $\mathbf{m}_c$. If there is such a t, then $\mathrm{TTC} = t$.

4.4.1 *Modelling Movement Recognition Accuracy*

The measured movement vector $\mathbf{m}_p$ of the pedestrian can be expressed as

$$\mathbf{m}_p = \mathbf{m}_{p_{gt}} + \Delta\mathbf{m} \tag{4.3}$$

where $\mathbf{m}_{p_{gt}} = (\mathbf{r}_{p_{gt}}, \phi_{p_{gt}}, v_{p_{gt}})$ is the ground truth movement vector of the pedestrian, i.e. the real movement of the pedestrian, and $\Delta\mathbf{m} = \mathbf{m}_p - \mathbf{m}_{p_{gt}} = ((\Delta x, \Delta y)^T, \Delta\phi, \Delta v)$ is the measurement error.

Let $\mathbf{XY} = (X, Y)$ denote the bivariate random variable for the lateral and the longitudinal positioning error. In other words, X models the random measurement error laterally to the pedestrian's moving direction while Y models the random measurement error in moving direction. Furthermore, let Φ denote a random variable for the direction error and V denote a random variable for the speed error for the recognition of the pedestrian's movement. With the assumption that these random variables are independent of each other, the probability

that a certain movement state $\mathbf{m}_p$ of the pedestrian is measured is defined as eq. (4.4),

$$P(\mathbf{m}_p) = p_{XY}(\Delta x, \Delta y) \cdot p_\Phi(\Delta\phi) \cdot p_V(\Delta v) \tag{4.4}$$

where $p_{XY}(\Delta x, \Delta y) = P(\boldsymbol{XY} = (\Delta x, \Delta y)^T)$ is the probability that a lateral position measurement error of Δx and a longitudinal position error of Δy occurs. Accordingly, $p_\Phi(\Delta\phi) = P(\Phi = \Delta\phi)$ relates to the direction measurement error probability and $p_V(\Delta v) = P(V = \Delta v)$ to the speed measurement error probability. Using this random variables based notation for the measurement errors allows to model the accuracy of the used sensor as the standard deviations of the random variables.

Assuming σ_X to be the standard deviation of X, σ_Y the standard deviation of Y, σ_V the standard deviation of V, and σ_Φ the standard deviation of Φ the set

$$\begin{aligned} \Delta M_p = \{((\Delta x, \Delta y), \Delta\phi, \Delta v) | \\ -3 \cdot \sigma_X \leqslant \Delta x \leqslant 3 \cdot \sigma_X, \\ -3 \cdot \sigma_Y \leqslant \Delta y \leqslant 3 \cdot \sigma_Y, \\ -3 \cdot \sigma_\Phi \leqslant \Delta\phi \leqslant 3 \cdot \sigma_\Phi, \\ -3 \cdot \sigma_V \leqslant \Delta v \leqslant 3 \cdot \sigma_V\} \end{aligned} \tag{4.5}$$

of all measurement errors within the coverage interval of 3σ (covering 99.7% of all values in the distribution) of the error distributions is created.

4.4.2 *Probability of Collision Detection*

Based on the set ΔM_p from eq. (4.5), the set

$$M_p = \{\mathbf{m}_p = \mathbf{m}_{gt} + \Delta\mathbf{m} | \Delta\mathbf{m} \in \Delta M_p\}$$

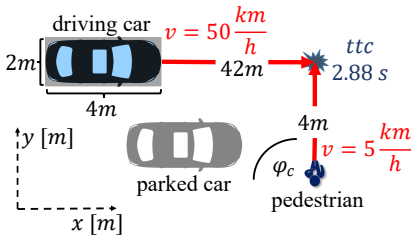
representing the set of all possible movement vectors due to recognition errors of the pedestrian's movement can be derived. By evaluating $col(\mathbf{m}_p, \mathbf{m}_c)$ (eq. (3.4) in Chapter 3), for all elements in M_p creates the set

$$M_{col} = \{\mathbf{m}_p \in M_p | col(\mathbf{m}_p, \mathbf{m}_c) = 1\} \tag{4.6}$$

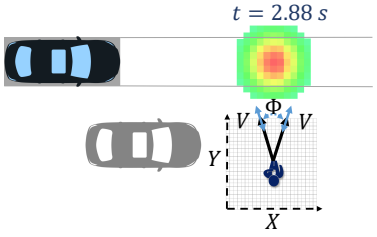
containing all the pedestrian's motion vectors that lead to a collision.

Using the set M_{col} eq. (4.6), the $P(cd)$ notation from Section 4.2 for collision scenarios and for non-collision scenarios both depending on the error distributions of the position, speed, and direction sensors can be specified as follows:

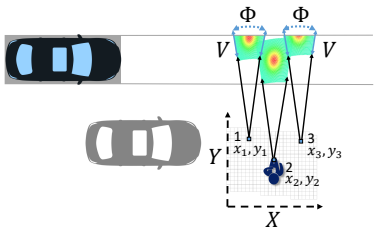
$$P(cd|((X, Y), \Phi, V) = \sum_{\mathbf{m}_p \in M_{col}} P(\mathbf{m}_p) \tag{4.7}$$



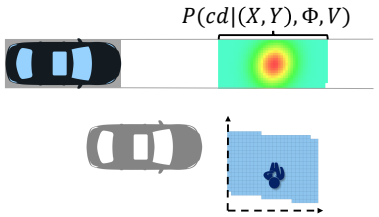
(a) CPNC-50 Collision scenario represented in a two-dimensional coordinate system



(b) Resulting position probability density for position prediction with t = 2.88 s. The colors indicate the local density where red is high and green is low



(c) Collision points for possible trajectories due to direction and speed uncertainty from three different, possible current positions



(d) Fused possible collision point density for all possible current positions. Blue positions indicate possible current positions from where a collision could be detected due to direction and speed uncertainty

Fig. 4.1: Sketches for determining the collision detection probability $P(cd|(X,Y),\Phi,V)$ due to uncertain random errors in position (X,Y), direction Φ, and speed V.

According to eq. (4.7) the *probability of collision detection depending on movement recognition accuracy* is the sum of the probabilities of all possible movement vectors due to sensor accuracy, as defined by eq. (4.5), for which a collision is detected. Fig. 4.1 shows sketches about the link between uncertain movement vectors and the $P(cd|(X,Y),\Phi,V)$ in a collision scenario between a car and a pedestrian.

Inserting eq. (4.7) in eq. (4.1) results in

$$P_{MA} = 1 - \sum_{\mathbf{m}_p \in M_{col}} P(\mathbf{m}_p) \tag{4.8}$$

which is the probability of a Missed Alarm P_{MA} depending on movement recognition errors for scenarios in which a collision occurs.

Likewise, the probability of a False Alarm can be derived by inserting eq. (4.7) in eq. (4.2)

$$P_{FA} = \sum_{\mathbf{m}_p \in M_{col}} P(\mathbf{m}_p) \tag{4.9}$$

which is the probability of a False Alarm P_{FA} for scenarios in which no collision occurs. As it can be seen in eqs. (4.8) and (4.9) to calculate P_{MA} and P_{FA} depending on movement recognition accuracy it is required to evaluate eq. (3.4) for all elements from the set M_p. In the next Chapter the basis to calculate the values of eqs. (4.8) and (4.9) is detailed.

4.5 BASIS OF THE P_{MA} AND P_{FA} CALCULATION

This section provides an approach to determine P_{MA} and P_{FA} for VRU CASes, depending on the VRU movement recognition accuracy as defined in Chapter 4. For simplicity and without loss of any generality, if not explicit stated otherwise, collisions between a car and a pedestrian are assumed. It is assumed that the pedestrian carries a smartphone which is capable to recognize its current movement consisting of current position, speed, and direction.

4.5.1 *Calculating the Movement Vector Probability* $P(\mathbf{m}_p)$

To calculate the probability that a movement vector $\mathbf{m}_p = \mathbf{m}_{p_{gt}} + ((\Delta x, \Delta y), \Delta\phi, \Delta v)$ is measured by the pedestrian's smartphone it is necessary to evaluate $P(\mathbf{m}_p) = p_{XY}(\Delta x, \Delta y) \cdot p_\Phi(\Delta\phi) \cdot p_V(\Delta v)$ as given in eq. (4.4). $p_{XY}(\Delta x, \Delta y)$, $p_\Phi(\Delta\phi)$, and $p_V(\Delta v)$ can be determined by integrating the probability density function of $\mathbf{XY}$, Φ, and V, respectively.

4.5.1.1 *Position Error Probability*

Let $\epsilon_{XY} > 0$ and e_{XY} be the density function of the bivariate position error distribution for the position sensor. The probability that a position

error of $(\Delta x, \Delta y)$ occurs for a single measurement from the position sensor then is

$$\begin{aligned} p_{XY}(\Delta x, \Delta y) = & \quad P[X \leqslant \Delta x + \frac{\epsilon_{XY}}{2}, Y \leqslant \Delta y + \frac{\epsilon_{XY}}{2}] \\ & - P[X \leqslant \Delta x - \frac{\epsilon_{XY}}{2}, Y \leqslant \Delta y + \frac{\epsilon_{XY}}{2}] \\ & - P[X \leqslant \Delta x + \frac{\epsilon_{XY}}{2}, Y \leqslant \Delta y - \frac{\epsilon_{XY}}{2}] \\ & + P[X \leqslant \Delta x - \frac{\epsilon_{XY}}{2}, Y \leqslant \Delta y - \frac{\epsilon_{XY}}{2}] \end{aligned} \tag{4.10}$$

and

$$P[X \leqslant x_e, Y \leqslant y_e] = \int_{-\infty}^{x_e} \int_{-\infty}^{y_e} e_{XY}(x, y) dx dy \tag{4.11}$$

4.5.1.2 *Direction Error Probability*

Let $\epsilon_\Phi > 0$ and e_Φ be the density function of the direction error distribution for the direction sensor. The probability that a direction error of $\Delta\phi$ occurs for a single measurement from the direction sensor then is

$$p_\Phi(\Delta\phi) = P[\Phi \leqslant \Delta\phi + \frac{\epsilon_\Phi}{2}] - P[\Phi \leqslant \Delta\phi - \frac{\epsilon_\Phi}{2}] \tag{4.12}$$

and

$$P[\Phi \leqslant \phi_e] = \int_{-\infty}^{\phi_e} e_\Phi(\phi) d\phi \tag{4.13}$$

4.5.1.3 *Speed Error Probability*

Let $\epsilon_V > 0$ and e_V be the density function of the speed error distribution for the speed sensor. The probability that a speed error of Δv occurs for a single measurement from the speed sensor then is

$$p_V(\Delta v) = P[V \leqslant \Delta v + \frac{\epsilon_V}{2}] - P[V \leqslant \Delta v - \frac{\epsilon_V}{2}] \tag{4.14}$$

and

$$P[V \leqslant v_e] = \int_{-\infty}^{v_e} e_V(v) dv \tag{4.15}$$

4.5.2 *Sum of Movement Vector Probabilities*

For determining P_{MA} and P_{FA}, the sum $\sum_{\mathbf{m}_p \in M_{col}} P(\mathbf{m}_p)$ needs to be calculated. This can be done by discretization of the set ΔM_p in the four intervals

$$I_X = \{-3\sigma_X \leqslant \Delta x \leqslant 3\sigma_X | \Delta x \mod \epsilon_{XY} = 0\}$$

$$I_Y = \{-3\sigma_Y \leqslant \Delta y \leqslant 3\sigma_Y | \Delta y \mod \epsilon_{XY} = 0\}$$

$$I_\Phi = \{-3\sigma_\Phi \leqslant \Delta \phi \leqslant 3\sigma_\Phi | \Delta \phi \mod \epsilon_\Phi = 0\}$$

$$I_V = \{-3\sigma_V \leqslant \Delta v \leqslant 3\sigma_V | \Delta v \mod \epsilon_V = 0\}$$

Using those intervals allows to compute $\sum_{\mathbf{m}_p \in M_{col}} P(\mathbf{m}_p)$ as follows

$$\begin{aligned}\sum_{\mathbf{m}_p \in M_{col}} P(\mathbf{m}_p) &= \sum_{\mathbf{m}_p \in M_p} col(\mathbf{m}_c, \mathbf{m}_p) \cdot P(\mathbf{m}_p) \\ &= \sum_{\Delta x \in I_X} \sum_{\Delta y \in I_Y} \sum_{\Delta v \in I_V} \sum_{\Delta \phi \in I_\Phi} \\ &\quad \cdot col(\mathbf{m}_c, \\ &\qquad ((x_{p_{gt}} + \Delta x, y_{p_{gt}} + \Delta y), \\ &\qquad \phi_{p_{gt}} + \Delta \phi, \\ &\qquad v_{p_{gt}} + \Delta v)) \\ &\quad \cdot p_{XY}(\Delta x, \Delta y) \cdot p_\Phi(\Delta \phi) \cdot p_V(\Delta v)\end{aligned} \tag{4.16}$$

The value for ϵ_i, $i \in \{XY, \Phi, V\}$ determine the precision for the values of $P(\mathbf{m}_p)$. The smaller the ϵ_i are chosen, the more precise $P(\mathbf{m}_p)$ gets. However, small ϵ_i values directly increase the number of element in the I_i intervals. Thus, calculate $\sum_{\mathbf{m}_p \in M_{col}} P(\mathbf{m}_p)$ with high precision demands more computing operations. For the upcoming results in this dissertation the values in eq. (4.17) for ϵ_i were used.

$$\begin{aligned}\epsilon_{XY} &= 0.02 \\ \epsilon_V &= 0.01 \\ \epsilon_\Phi &= 0.1\end{aligned} \tag{4.17}$$

In the next Chapter 5 an approach called "Direction Ranges" is proposed that allows to calculate eq. (4.16) without to need to evaluate every single movement vector $\mathbf{m}_p$ in M_p.

REFERENCES

[4.1] M. Bachmann, M. Morold, K. David and P. Henkel, 'The Wireless Seat Belt Requirements, Experiments, and Solutions for Pedestrian Safety', in *14th Workshop on Context and Activity Modeling and Recognition*, Athens, Greece, Mar. 2018, pp. 250–255. DOI: 10.1109/PERCOMW.2018.8480123 (cit. on p. 53).

5

USING COLLISION DIRECTION RANGES FOR P_{MA} AND P_{FA} CALCULATION

Acknowledgement: Parts of this Chapter, especially the "subdivision and reconstruction" part and most of the figures, were initially created by Johann Götz during his Master-thesis supervised by the author.

A Collision Direction Range (CDR) is a continuous set of movement directions for road user p that lead to a collision with the car. CDRs arise whenever geometries for the road users are considered, i.e. at least one road user is not represented as a point. A CDR consists of a minimum and a maximum direction threshold $[\varphi_{min}, \varphi_{max}]$, for p (normalized to a direction interval of $[0°, 360°)$). Depending on the movement vector $\mathbf{m}_c$ of the road user c and the position and speed information $(\mathbf{r}_p, v_p)$ of the road user p , a scenario can have either none (no collision in any direction), one (see Fig. 5.1a) or two direction ranges (see Fig. 5.1b) for the road user p.

For scenarios where at least one CDRs exists, let

$$\mathbf{dr}_k(\mathbf{m}_c, (\mathbf{r}_p, v_p)) = [\varphi_{k_{min}}, \varphi_{k_{max}}], k \in \{1, 2\} \tag{5.1}$$

denote the up to two CDRs. Let $inRange$ (in collision direction range) denote the event that a measured direction for the pedestrian is inside a CDR eq. (5.1).

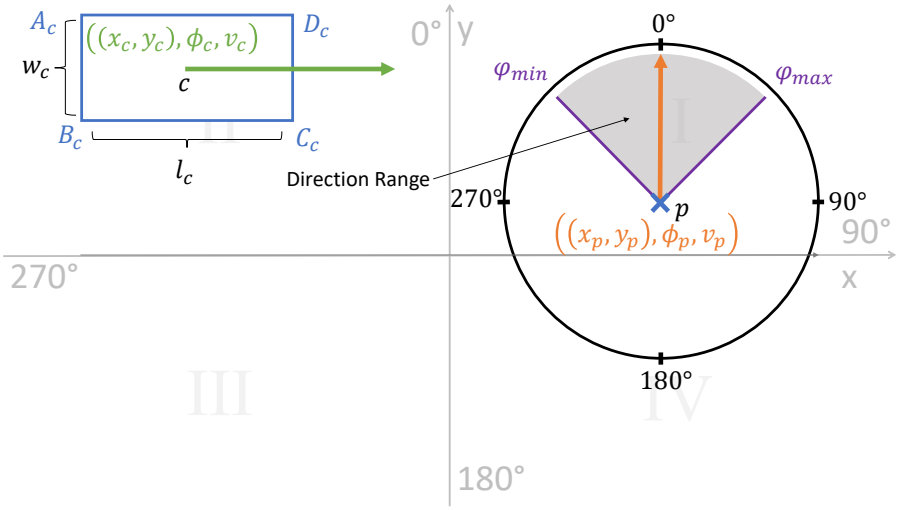
By using eq. (5.1) the *measurement in collision range probability depending on* Φ, $p_\Phi(inRange)$ can be denoted as

$$p_\Phi(inRange) = \sum_{k=1}^{2} P[\varphi_{k_{min}} \leqslant \Phi \leqslant \varphi_{k_{max}}] = \sum_{k=1}^{2} \int_{\varphi_{k_{min}}}^{\varphi_{k_{max}}} e_\Phi(\varphi) d\varphi \tag{5.2}$$

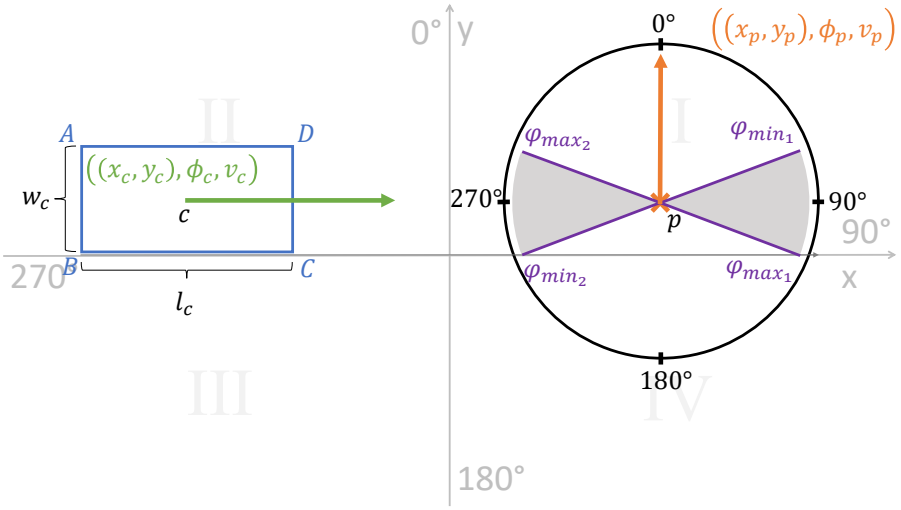
Replacing the inner sum factor $\sum_{\Delta\phi \in I_\Phi} col(\mathbf{m}_c, \mathbf{m}_p)$ from eq. (4.16) with eq. (5.2) results in eq. (5.3).

$$\begin{aligned} \sum_{\mathbf{m}_p \in M_{col}} P(\mathbf{m}_p) &= \sum_{\mathbf{m}_p \in M_p} col(\mathbf{m}_c, \mathbf{m}_p) \cdot P(\mathbf{m}_p) \\ &= \sum_{\Delta x \in I_X} \sum_{\Delta y \in I_Y} \sum_{\Delta v \in I_V} \cdot p_{XY}(\Delta x, \Delta y) \cdot p_V(\Delta v) \\ &\quad \cdot \sum_{k=1}^{2} p_\Phi(inRange) \end{aligned} \tag{5.3}$$

Thus, the $|I_\Phi|$ function calls for $col(\mathbf{m}_c, \mathbf{m}_p)$ in eq. (4.16) are replaced by once determining eq. (5.1), i.e replace the iteration over $|I_\Phi|$



(a) One range



(b) Two ranges

Fig. 5.1: Collision Direction Ranges

values with evaluating the probability density function of Φ for the up to two CDRs. In the remainder of this section, the Subdivision and Reconstruction (SubRec) approach from Section 3.6 is used and combined with either normalized or relative Road Space Model (RSM) transformation to determine the CDRs $\mathbf{dr}_j(\mathbf{m}_c, (\mathbf{r}_p, v_p))$.

5.1 BASIC APPROACH FOR COLLISION DIRECTION RANGES

The general idea of CDR calculation is to equate $t_x = \frac{x_c - x_p}{v_p \sin\phi_p - v_c \sin\phi_c}$ and $t_y = \frac{y_c - y_p}{v_p \cos\phi_p - v_c \cos\phi_c}$ from eq. (3.6) for the point-to-point collision detection in Section 3.3 and rearrange this equation to ϕ_p:

$$
\begin{aligned}
& t_x = t_y \\
\iff & \frac{x_c - x_p}{v_p \sin\phi_p - v_c \sin\phi_c} = \frac{y_c - y_p}{v_p \cos\phi_p - v_c \cos\phi_c} \\
\iff & \bar{\varphi}_j = ...
\end{aligned}
\tag{5.4}
$$

Note that ϕ_p in eq. (5.4) is an argument of $\sin\phi_p$ and $\cos\phi_p$. Thus rearranging eq. (5.4) will require to apply the Pythagorean trigonometric identity ($\sin^2\phi_p + \cos^2\phi_p = 1$). It is important to emphasize that the resulting collision directions are *not* the direction ϕ_p of the road user p but the directions for which there will be a collision between p and c if p *would* move in one of the collision directions. To avoid confusion between the actual moving direction ϕ_p of road user p and a collision direction, collision direction values are indicated with φ, see last line of eq. (5.4).

The result will be a set of *pseudo* collision direction thresholds ($\{\bar{\varphi}_j\}$, $j \in \mathbb{N}$). Among the correct collision direction thresholds $\varphi_{k_{min}}, \varphi_{k_{max}}$, $k \in \{1, 2\}$, $\{\bar{\varphi}_j\}$ will unavoidable contain incorrect values due to the ambiguity of the square-root function and the symmetry of the used trigonometric functions. Thus, the correct $\varphi_{k_{min}}, \varphi_{k_{max}}$ CDR thresholds need to be obtained by filtering $\{\bar{\varphi}_j\}$, as detailed in Section 5.6.

Before the collision directions $\bar{\varphi}_j$ are determined using he Pythagorean trigonometric identity, the equation can and should be shortened using RSM transformation, like in Chapter 3. Considering eq. (5.4) it can be noted that, strictly speaking, for a point-to-point RSM the result is yet not a collision direction *range* but up to two single directions since points have no area. However, bringing in geometries, as it will be detailed to the following, naturally will end up with direction ranges.

5.2 ROAD SPACE MODEL TRANSFORMATIONS FOR COLLISION DIRECTION RANGES

Similarly to the RSM transformation for collision detection in Section 3.5, the RSM transformation for CDR can be used to calculate direction thresholds more efficiently. While the normalized RSM transformation for CDR calculation is the same as for collision detection, the relative RSM transformation for RSM is further optimized into a rotated relative scenario transformation. For this, the RSM is rotated so that $\mathbf{r}_p$ is on the x-axis of the coordinate system, as detailed in the following section.

5.2.1 *Rotated Relative RSM*

To obtain a rotated relative RSM, three steps are required. First, the RSM is normalized. Second, the normalized RSM is transformed into a

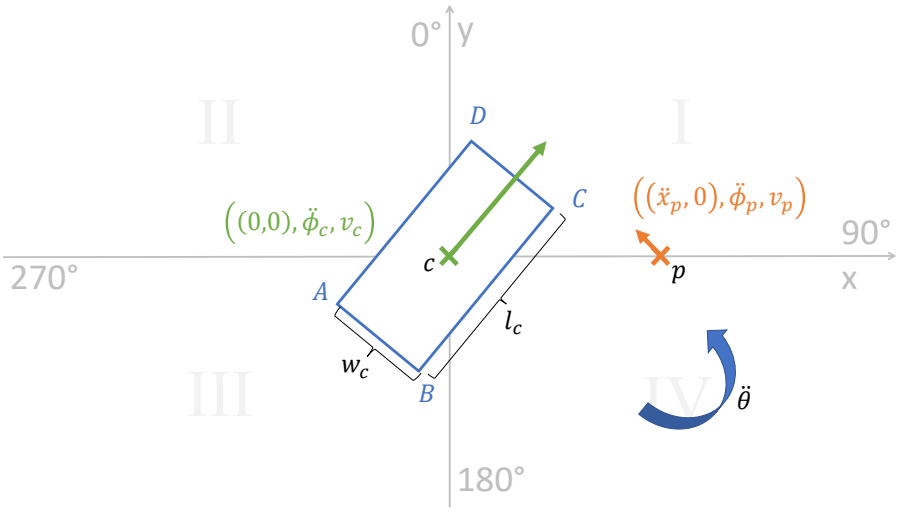


Fig. 5.2: Rotated normalized Road Space Model

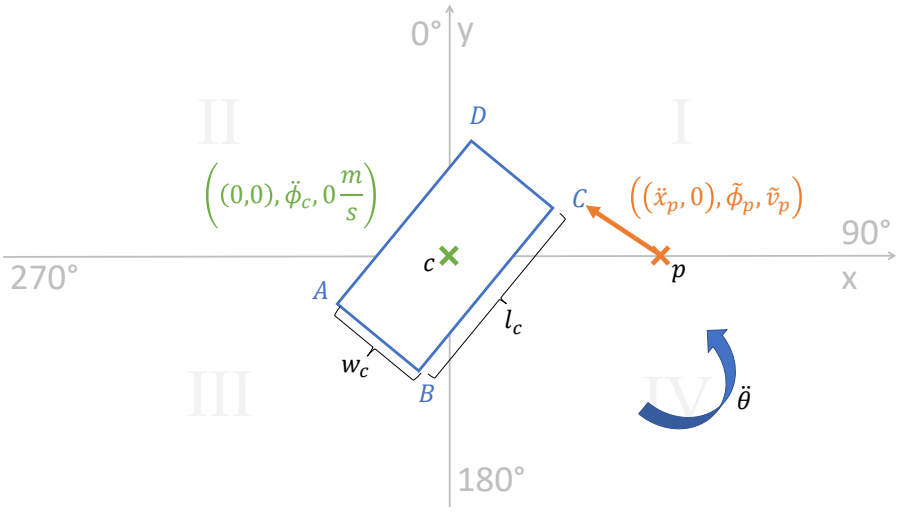


Fig. 5.3: Rotated relative Road Space Model

rotated normalized RSM like shown in Fig. 5.2. In a rotated normalized scenario, p is aligned directly at the x-axis so that p's y-position is $\ddot{y}_p = 0$. To align p, a normalized vector $\mathbf{b} = (1, 0)$ is defined which is parallel to the x-axis. Next, the rotation angle $\ddot{\theta}$ between $\mathbf{b}$ and the position $\dot{\mathbf{r}}_p = (\dot{x}_p, \dot{y}_p)^T$ is calculated, as shown in eq. (5.5).

$$\begin{aligned}\ddot{\theta} &= \operatorname{atan2}(\dot{y}_p, \dot{x}_p) - \operatorname{atan2}(b_y, b_x) \\ &= \pm \arccos\left(\frac{\mathbf{b}\dot{\mathbf{r}}}{\|\mathbf{b}\| \, \|\dot{\mathbf{r}}\|}\right) \\ &= \pm \arccos\left(\frac{\dot{x}_p}{\sqrt{\dot{x}_p^2 + \dot{y}_p^2}}\right)\end{aligned} \tag{5.5}$$

Using eq. (5.6), $\dot{\mathbf{r}}_p$ is rotated by $\ddot{\theta}$ around the origin to obtain $\ddot{\mathbf{r}}_p = (\ddot{x}_p \ddot{y}_p)$. The directions of p and c are rotated by $\ddot{\theta}$ as well.

$$\begin{aligned}\ddot{x}_p &= \dot{x}_p \cos\left(-\ddot{\theta}\right) - \dot{y}_p \sin\left(-\ddot{\theta}\right), \\ \ddot{y}_p &= 0, \\ \ddot{\phi}_p &= \dot{\phi}_p + \ddot{\theta}, \\ \ddot{\phi}_c &= \dot{\phi}_c + \ddot{\theta}\end{aligned} \tag{5.6}$$

Third, the relative RSM transformation is applied (as shown in Section 3.5.2) on the rotated normalized RSM to obtain a rotated relative RSM, as shown in Fig. 5.3.

By using the rotated relative RSM transformation, it is possible to determine relative direction ranges more easily compared to the sole relative RSM transformation. When there is an impending collision between p and c in a rotated relative RSM, p is forced to move towards c which only happens in the I and IV quadrants of the coordinate system. Thus, the relative movement of p can be restricted so that $\tilde{\phi} \in [180°, 360°)$, i.e., so that p moves from "right" to "left", limiting the value domain of direction ranges by half.

5.3 POINT-TO-POINT COLLISION DIRECTIONS ALGORITHM

Equation eq. (5.4) for a point-to-point RSM in which c and p are modelled as points, can be shortened by using the normalized or the relative RSM transformation, as detailed in the following subsections. As it will be discussed in 'SubRec for Pseudo Collision Direction Thresholds' on page 72 the point-to-point RSM is a basic component for any CDR algorithm. Thus, the algorithms for RSM with geometries are all based on these transformation methods.

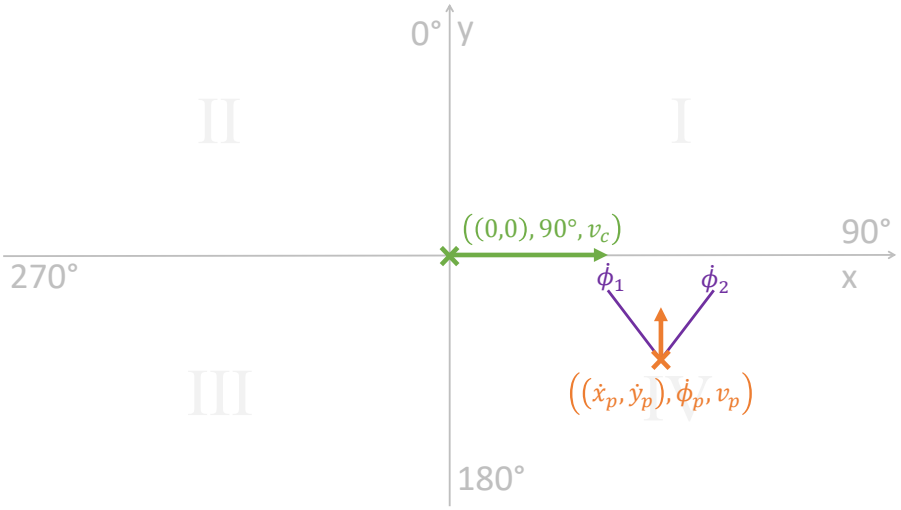


Fig. 5.4: Normalized point-to-point RSM

5.3.1 *Using the Normalized RSM Transformation*

Consider a normalized point-to-point RSM, like shown in Fig. 5.4. In this normalized point-to-point RSM, eq. (5.4) can be shortened with $\dot{x}_c = 0$, $\dot{y}_c = 0$, $\sin\dot{\phi}_c = 1$ and $\cos\dot{\phi}_c = 0$ and rearranged, see eq. (5.7).

$$
\begin{aligned}
& t_x = t_y \\
\iff & \frac{\dot{x}_c - \dot{x}_p}{v_p \sin\dot{\phi}_p - v_c \sin\dot{\phi}_c} = \frac{\dot{y}_c - \dot{y}_p}{v_p \cos\dot{\phi}_p - v_c \cos\dot{\phi}_c} \qquad (5.7) \\
\iff & \frac{-\dot{x}_p}{v_p \sin\dot{\phi}_p - v_c} = \frac{-\dot{y}_p}{v_p \cos\dot{\phi}_p} \\
\iff & 0 = v_p^2 \left(\dot{x}_p^2 + \dot{y}_p^2\right) \cos^2\dot{\phi}_p \\
& \quad + 2\dot{x}_p v_p \dot{y}_p v_c \cos\dot{\phi}_p \\
& \quad + \dot{y}_p^2 \left(v_c^2 - v_p^2\right) \qquad (5.8)
\end{aligned}
$$

To obtain the normalized pseudo collision direction thresholds $\dot{\varphi}_j$ for p the pq-formula eq. (5.9)

$$
\begin{aligned}
\Rightarrow \quad \frac{p}{2} &= \frac{\dot{x}_p \dot{y}_p v_c}{v_p \left(\dot{x}_p^2 + \dot{y}_p^2\right)} \\
q &= \frac{\dot{y}_p^2 \left(v_c^2 - v_p^2\right)}{v_p^2 \left(\dot{x}_p^2 + \dot{y}_p^2\right)} \\
\left(\frac{p}{2}\right)^2 - q &= \frac{\dot{y}_p^2 \left(\dot{x}_p^2 v_c^2 - \left(v_c^2 - v_p^2\right)\left(\dot{x}_p^2 + \dot{y}_p^2\right)\right)}{v_p^2 \left(\dot{x}_p^2 + \dot{y}_p^2\right)^2} \qquad (5.9)
\end{aligned}
$$

is applied to solve eq. (5.8). By inserting eq. (5.9) in eq. (5.10), $\cos\dot{\phi}_p$ and thus $\dot{\varphi}_j$ can be calculated.

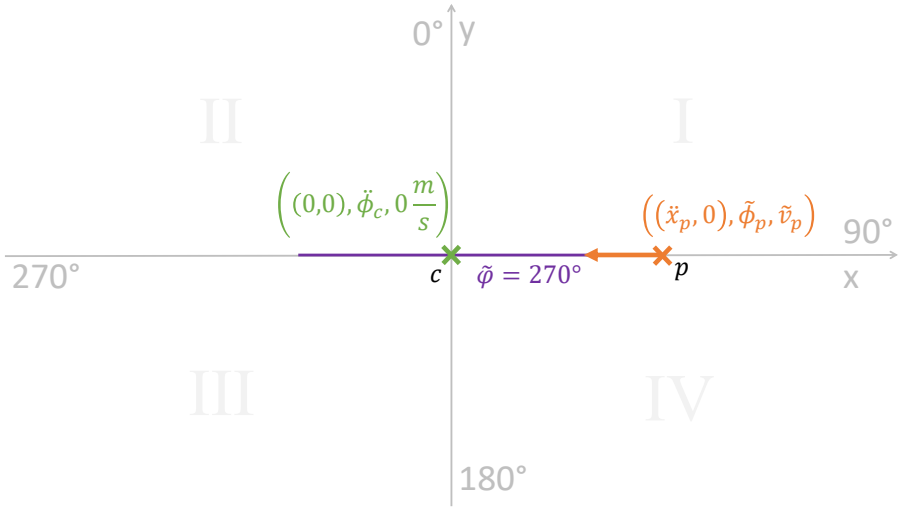


Fig. 5.5: $\tilde{\varphi} = 270°$ in a rotated relative point-to-point RSM

$$
\begin{aligned}
\Rightarrow \cos\dot{\phi}_p &= -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q} \\
&= -\frac{\dot{x}_p \dot{y}_p v_c}{v_p\left(\dot{x}_p^2 + \dot{y}_p^2\right)} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q} \\
\Rightarrow \dot{\varphi}_j &= \pm \arccos\left(-\frac{\dot{x}_p \dot{y}_p v_c}{v_p\left(\dot{x}_p^2 + \dot{y}_p^2\right)}\right. \\
&\quad \left. \pm \sqrt{\frac{\dot{y}_p^2\left(\dot{x}_p^2 v_c^2 - \left(v_c^2 - v_p^2\right)\left(\dot{x}_p^2 + \dot{y}_p^2\right)\right)}{v_p^2\left(\dot{x}_p^2 + \dot{y}_p^2\right)^2}}\right)
\end{aligned}
\tag{5.10}
$$

5.3.2 *Using Rotated Relative RSM Transformation*

Consider a rotated relative point-to-point RSM, like shown in Fig. 5.5. In this rotated relative point-to-point RSM, the position of road user c is established as the origin $(0,0)$ of the RSM coordinate system and thus its relative speed to the coordinate system frame is $\tilde{v}_c = 0$. In a rotated relative point-to-point RSM, the collision between c and p is only possible if p will be at origin at any time. Since in the rotated relative RSM p is placed on the x-axis and its speed $\tilde{v}_p > 0$, the only collision trajectory for p is along the x-axis from position $(\ddot{x}_p, 0)$ to $(0,0)$. That effects that the relative direction threshold $\tilde{\varphi}$ for rotated relative point-to-point RSMs is always 270° and therefore, compared to a normalized point-to-point RSM, the relative collision direction does not need to be calculated. However, due to the used rotations of the RSM, neither $\dot{\tilde{\varphi}}_j$ from the normalized RSM nor $\tilde{\varphi}$ from the rotated relative RSM are already the World Geodetic System 1984 (WGS84) pseudo collision directions for the modelled real world scenario. Thus,

a "backward calculation" from $\dot{\varphi}_j$ and $\tilde{\varphi}$ to $\bar{\varphi}_j$ is necessary which is detailed in the next section.

5.4 BACKWARDS CALCULATION

Before the WGS84 direction thresholds $\varphi_{k_{min}}$ and $\varphi_{k_{msx}}$, $k \in \{1,2\}$ for a real world scenario can be extracted from the pseudo collision direction thresholds $\bar{\varphi}_j$, the pseudo collision direction thresholds from a normalized or a rotated relative RSM need to be calculated backwards.

When a rotated relative RSM was used, the first step is to calculate $\ddot{\varphi}_j$ from $\tilde{\varphi}$, i.e., the collision direction thresholds for the rotated normalized RSM: Let $\tilde{v}_x = \tilde{v}\sin\tilde{\phi}$ as introduced in Section 3.5.2 for the relative RSM transformation. The terms $\tilde{v}$ and $\tilde{v}_x$ can be substituted back with $\tilde{v}_x = v_{p_x} - v_{c_x}$, $\tilde{v}_y = v_{p_y} - v_{c_y}$ and $v_{i_x} = v_i \sin\ddot{\phi}_i$, $v_{i_y} = v_i \cos\ddot{\phi}_i$.

$$\begin{aligned} & \tilde{v}_x = \tilde{v}\sin\tilde{\phi} \\ \iff\ & 0 = v_p^2\cos^2\ddot{\phi}_p \\ & \quad + 2v_p\sin\tilde{\phi}\left(v_{c_x}\cos\tilde{\phi} - v_{c_y}\sin\tilde{\phi}\right)\cos\ddot{\phi}_p \\ & \quad + \left(v_{c_x}\cos\tilde{\phi} - v_{c_y}\sin\tilde{\phi}\right)^2 - v_p^2\cos^2\tilde{\phi} \end{aligned} \tag{5.11}$$

For eq. (5.11), the pq-formula can be applied to get $\frac{p}{2}$ and q eq. (5.12) which are used to calculate $\ddot{\varphi}_j$ eq. (5.13).

$$\begin{aligned} \Rightarrow \frac{p}{2} &= \frac{v_p\sin\tilde{\phi}\left(v_{c_x}\cos\tilde{\phi} - v_{c_y}\sin\tilde{\phi}\right)}{v_p^2} \\ &= \frac{\sin\tilde{\phi}\left(v_{c_x}\cos\tilde{\phi} - v_{c_y}\sin\tilde{\phi}\right)}{v_p}, \\ q &= \frac{\left(v_{c_x}\cos\tilde{\phi} - v_{c_y}\sin\tilde{\phi}\right)^2 - v_p^2\cos^2\tilde{\phi}}{v_p^2} \\ &= \frac{\left(v_{c_x}\cos\tilde{\phi} - v_{c_y}\sin\tilde{\phi}\right)^2}{v_p^2} - \cos^2\tilde{\phi} \end{aligned} \tag{5.12}$$

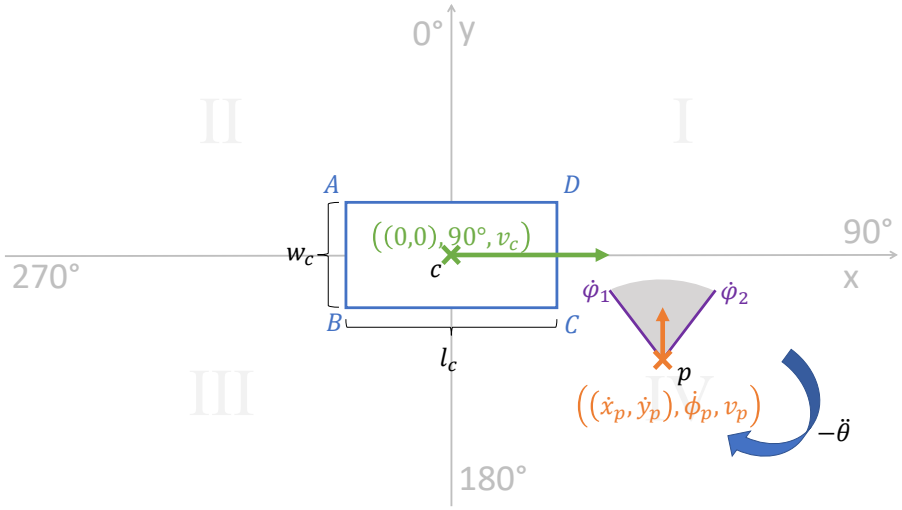


Fig. 5.6: Backwards rotation from a rotated relative RSM to a normalized RSM without pseudo collision direction thresholds

$$
\begin{aligned}
\Rightarrow \cos \ddot{\phi}_{p_{1,2}} &= -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q} \\
&= -\frac{\sin\tilde{\phi}\left(v_{c_x}\cos\tilde{\phi} - v_{c_y}\sin\tilde{\phi}\right)}{v_p} \\
&\quad \pm \sqrt{\frac{\cos^2\tilde{\phi}\left(-\left(v_{c_x}\cos\tilde{\phi} - v_{c_y}\sin\tilde{\phi}\right)^2 - v_p^2\right)}{v_p^2}}, \qquad (5.13) \\
\Rightarrow \ddot{\varphi}_j &= \pm \arccos\left(-\frac{\sin\tilde{\phi}\left(v_{c_x}\cos\tilde{\phi} - v_{c_y}\sin\tilde{\phi}\right)}{v_p}\right. \\
&\quad \left. \pm \sqrt{\frac{\cos^2\tilde{\phi}\left(-\left(v_{c_x}\cos\tilde{\phi} - v_{c_y}\sin\tilde{\phi}\right)^2 - v_p^2\right)}{v_p^2}}\right)
\end{aligned}
$$

In the next step, the $\ddot{\varphi}_j$ have to be rotated back by $-\ddot{\theta}$ to obtain the pseudo collision direction thresholds $\dot{\varphi}_j$ for the normalized RSM, as shown in Fig. 5.6. Finally, the pseudo collision direction thresholds have to be rotated once more by $-\dot{\theta}$ to obtain the WGS84 pseudo collision direction thresholds $\hat{\varphi}_j$ for the real world scenario, as shown in Fig. 5.7.

The correct collision direction thresholds are then determined by using the filtering algorithm, introduced in Section 5.6. In the next sections, algorithms for the pseudo collision direction range calculation with geometries, such as a rectangle and a circle, are provided which are constructed using SubRec.

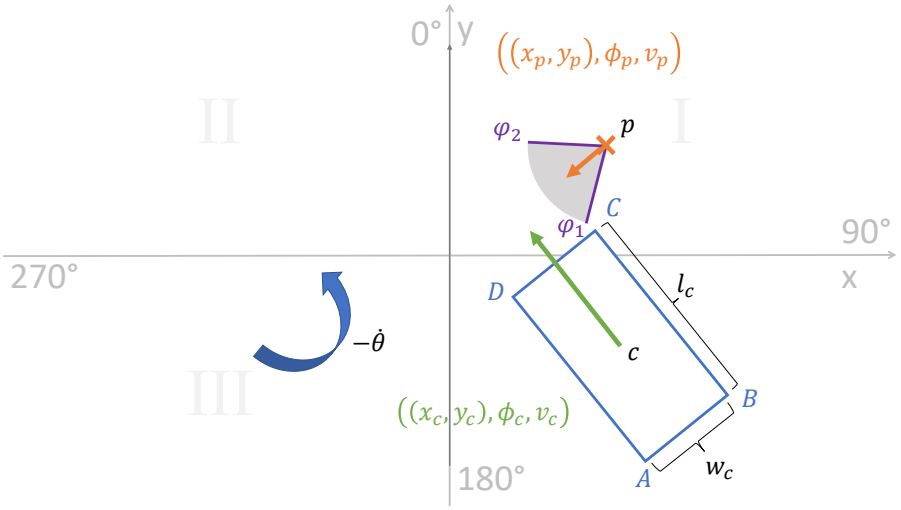


Fig. 5.7: Backwards rotation from a normalized RSM to the original RSM (1:1 WGS84 mapping) without pseudo collision direction thresholds

5.5 SUBREC FOR PSEUDO COLLISION DIRECTION THRESHOLDS

When c and p collide with each other at their respecting geometries, this means there is a single tangential point between the geometries. The direction thresholds of the CDRs are calculated from c's trajectory and p's position and speed to find all values for CDRs thresholds. Therefore, it is possible to reduce the geometries in a RSM to four points for each direction threshold calculation.

Let G_c be a geometry of c, consisting of geometry parts $g_i \in G_c$, and G_p a geometry of p. The SubRec algorithm for collision detection from Section 3.6, that subdivides the geometries of the road users into g-to-point, $g \in G_c$ components, can be used in a similar way for for CDRs. However, since each collision at the direction thresholds take place between two points, it is possible to reuse either the normalized or the relative point-to-point component in order to reconstruct more complex RSMs. Therefore, the basic calculation for point-to-point CDRs is more useful compared to the point-to-point collision detection from section Section 3.3. In the following, two SubRec algorithms for normalized and relative CDR calculations are provided.

5.5.1 *Algorithm for Normalized RSM Transformation*

This SubRec algorithms for normalized CDRs has two road users c and p as input whereby c's geometry G_c can be subdivided into four points or circles g_{c_i}, $i \in \{1, 2, 3, 4\}$. The parameters speed an direction for g_{c_i} are copied from the road user c. The RSM for (g_i, p) needs to re-transformed once again in a normalized or a relative RSM (effecting an additional rotation by $\check{\theta}$) so that the g_i-RSM is transformed for the g_i component instead for the original geometry G. Afterwards, the

Algorithm 6: CDR calculation for normalized RSM

input : (c, p, G_c)
output: $\mathbf{d}_k$

$\Theta \leftarrow \emptyset$;
foreach $g_i \in G_c$ **do**
  $\check{\theta} \leftarrow$ adjust the RSM transformation for (g_i, p);
  $\check{\varphi}_{i_j} \leftarrow$ calculate pseudo collision direction thresholds for (g_i, p) according to eq. (5.10);
  $\bar{\varphi}_{i_j} \leftarrow (\check{\varphi}_{i_j} - (\check{\theta} + \dot{\theta}))$;
  Θ.append$(\bar{\varphi}_{i_j})$;
end
$\mathbf{d}_k \leftarrow$ filter(Θ);

g_i-to-point component for direction ranges can be used to calculate the pseudo collision direction thresholds $\check{\varphi}_j$, $j \in \mathbb{N}$. Then, $\check{\varphi}_j$ are rotated back by $-(\check{\theta} + \dot{\theta})$ to obtain the pseudo collision direction thresholds $\bar{\varphi}_j$ for original WGS84 RSM. Finally, the pseudo collision direction thresholds $\bar{\varphi}_j$ are filtered and assembled into direction ranges $\mathbf{d}_k$ using the filtering algorithm from Section 5.6.

5.5.2 *Algorithm for Rotated Relative RSM Transformation*

The CDR algorithm 7 for relative direction ranges has two road users c and p as input whereby c′ has a geometry that can be subdivided into points g_i, $i \in \mathbb{N}$. The difference of algorithm 7 in contrast to algorithm 6 is that no rotation adjustment for of the rotated relative RSM is required for the single g_i components. This is achieved by using difference vector calculation between p and every g_i component. The algorithm works as follows: First the two possible relative collision direction thresholds are initialized to the intrinsic relative collision direction in a point-to-point rotated relative RSM by $\tilde{\varphi}_1 = \tilde{\varphi}_2 = 270°$, compare section 5.3.2. Afterwards, for each g_i, the angle φ_Δ between p and g_i is calculated using the difference vector calculation. If $\varphi_\Delta < \tilde{\varphi}_1$, φ_Δ is used as the new minimum collision direction threshold $\tilde{\varphi}_1 = \varphi_\Delta$, i.e. finding the "start" value for the relative collision direction range. If $\varphi_\Delta > \tilde{\varphi}_2$, φ_Δ is used as the new maximum collision direction threshold $\tilde{\varphi}_2 = \varphi_\Delta$, i.e. setting the "end" value for the relative collision direction range. Afterwards, the direction thresholds $\ddot{\varphi}$ are calculated from $\tilde{\varphi}_{1,2}$ using the backwards calculation eq. (5.13). Finally, the direction thresholds are filtered and assembled into direction ranges $\mathbf{d}_k$ using the filtering algorithm 8 from Section 5.6.

Algorithm 7: CDR calculation for rotated relative RSM

input : (c, p, G_c)
output: $\mathbf{d}_k$

$\Theta \leftarrow \emptyset$;
$\tilde{\varphi}_1 \leftarrow \tilde{\varphi}_2 \leftarrow 270°$;
foreach $g_i \in G_c$ **do**
  $\delta = (-\mathbf{e}_x \angle(\mathbf{r}_{g_i} - \ddot{\mathbf{r}}_p))$;
  if $y_{g_i} > 0$ **then**
    $\varphi_\Delta \leftarrow 270° + \delta$;
  else
    $\varphi_\Delta \leftarrow 270° - \delta$;
  end
  if $\varphi_\Delta < \tilde{\varphi}_1$ **then**
    $\tilde{\varphi}_1 \leftarrow \varphi_\Delta$;
  end
  if $\varphi_\Delta > \tilde{\varphi}_2$ **then**
    $\tilde{\varphi}_2 \leftarrow \varphi_\Delta$;
  end
end
for $j \leftarrow 1$ **to** 2 **do**
  $\ddot{\bar{\varphi}}_j \leftarrow$ use backwards calculation for $\tilde{\varphi}_j$ according to eq. (5.13);
  $\bar{\varphi}_{i_j} \leftarrow \ddot{\bar{\varphi}}_{i_j} - (\dot{\theta} + \ddot{\theta})$;
  Θ.append$(\bar{\varphi}_{i_j})$;
end
$\mathbf{d}_k \leftarrow$ filter(Θ);

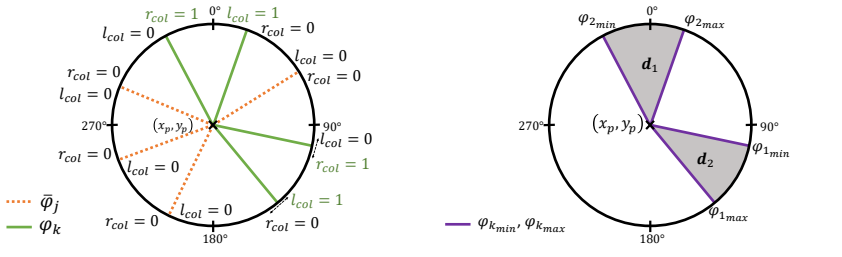
5.6 DIRECTION RANGE FILTERING

Some of direction thresholds $\bar{\varphi}_j$, $j \in \mathbb{N}$ that are calculated by the algorithms 6 and 7, are pseudo collision direction thresholds, i.e. duplicate values or not part of the correct CDRs. Such pseudo collision direction thresholds naturally result due to the ambiguity of the square-root function and the symmetry of the used trigonometric functions. Therefore, a filtering algorithm for pseudo CDR thresholds is introduced in the following which searches for the correct collision direction thresholds $\varphi_{k_{min}}$, $\varphi_{k_{max}}$ and merges them into CDRs $\mathbf{d}_k$.

5.6.1 *Filtering Algorithm*

The filtering algorithm consists of two steps. First, filtering duplicate and incorrect values and, second, create the resulting $\mathbf{d}_k$ CDRs from the correct remaining collision direction thresholds. In the filter step, shown in algorithm 8, every direction threshold $\bar{\varphi}_j$ in the input set Θ is sorted from the lowest to the highest value while duplicate values are removed. The beginning $\varphi_{k_{min}}$ of one of the $k = \{1, 2\}$ CDRs is detected by checking for each threshold $\bar{\varphi}_j$ whether $\text{col}(\mathbf{m}_c, \mathbf{m}_p) \neq 0$ with $\mathbf{m}_p = (\mathbf{r}_p, v_p, \bar{\varphi}_j \pm \epsilon_\varphi)$, $\epsilon_\varphi > 0$, i.e., check if a small change of $\bar{\varphi}_j$ in clockwise or counter-clockwise direction from the threshold, still lead to a collision ($l_{col} = 1$ for collision in counter-clockwise (left) and $r_{col} = 1$ for collision in clockwise (right) direction). Note that it is crucial to use a precise collision detection algorithm for $\text{col}(\mathbf{m}_c, \mathbf{m}_p)$ like the one proposed in Chapter 3 to determine l_{col} and r_{col} reliable for small ϵ_φ. From experiments it was found that $\epsilon_\varphi = 0.01$ can be used as a default value for determining slightly lower and higher values. Depending on the accuracy of floating numbers in a real life simulation, ϵ_φ can be varied to find left and right collisions more reliably.

When $l_{col} = 0$ and $r_{col} = 1$ for $\bar{\varphi}_j \pm \epsilon_\varphi$ the start of a collision detection range is found and thus $\varphi_{k_{min}} = \bar{\varphi}_j$. Likewise, a collision direction range ends with a direction threshold $\varphi_{k_{max}} = \bar{\varphi}_j$ when $l_{col} = 1$ and $r_{col} = 0$. A direction threshold $\bar{\varphi}_j$ belongs to a CDR if it



(a) Find $\varphi_{j_{min}}$ using l_{col} and r_{col} (b) Merge φ_j into direction ranges

Fig. 5.8: CDR filtering

is inside a range, i.e., if $l_{col} = 1$ and $r_{col} = 1$. Every other direction threshold, that is outside of both direction CDRs, is discarded.

The up to two values for $\varphi_{k_{min}}$ and $\varphi_{k_{max}}$ are then processed by the function createCDR(), defined in algorithm 9, that creates the actual CDRs $\mathbf{d}_k$.

Algorithm 8: Filter CDR Threholds

input : Θ, ϵ_φ
output: $\mathbf{d}_k$

$\mathbf{d}_k \leftarrow \emptyset$;
$\varphi_{1_{min}} \leftarrow \varphi_{2_{min}} \leftarrow \varphi_{1_{max}} \leftarrow \varphi_{2_{max}} \leftarrow$ null;
sort(Θ);
removeDuplicates(Θ);
$k \leftarrow 0$;
foreach $\varphi_j \in \Theta$ **do**
    $l_{col} \leftarrow \text{col}(\mathbf{m}_c, (\mathbf{r}_p, \nu_p, \bar{\varphi}_j + \epsilon_\varphi))$;
    $r_{col} \leftarrow \text{col}(\mathbf{m}_c, (\mathbf{r}_p, \nu_p, \bar{\varphi}_j - \epsilon_\varphi))$;
    if $(l_{col} = 0 \wedge r_{col} = 0) \vee (l_{col} = 1 \wedge r_{col} = 1)$ **then**
        discard($\bar{\varphi}_j$);
    end
    if $l_{col} = 0 \wedge r_{col} = 1$ **then**
        if $\varphi_{1_{min}} =$ null **then**
            $\varphi_{1_{min}} \leftarrow \bar{\varphi}_j$;
            $k \leftarrow 1$;
        else
            $\varphi_{2_{min}} \leftarrow \bar{\varphi}_j$;
            $k \leftarrow 2$;
        end
    end
    if $l_{col} = 1 \wedge r_{col} = 0$ **then**
        if $\varphi_{1_{max}} =$ null **then**
            $\varphi_{1_{max}} \leftarrow \bar{\varphi}_j$;
        else
            $\varphi_{2_{max}} \leftarrow \bar{\varphi}_j$;
        end
    end
end
$\mathbf{d}_k \leftarrow$ createCDR($\varphi_{k_{min}}, \varphi_{k_{max}}$);

Algorithm 9: Create CDR

```
1  Function createCDR(φ_{k_min}, φ_{k_max}) : d_k is
2      d_k ← ∅;
3      if j > 0 then
4          if φ_{1_min} < φ_{1_max} then
5              d_1 ← (φ_{1_min}, φ_{1_max});
6              if j > 1 then
7                  d_2 ← (φ_{2_min}, φ_{2_max});
8              end
9          else
10             if j > 1 then
11                 d_1 ← (φ_{2_min}, φ_{1_max});
12                 d_2 ← (φ_{1_min}, φ_{2_max});
13             else
14                 d_1 ← (φ_{1_min}, φ_{1_max});
15             end
16         end
17     end
18     return d_k;
19 end
```

5.7 RECTANGLE-TO-POINT COLLISION DIRECTION RANGES ALGORITHM

Consider a rectangle-to-point RSM in which c is represented by a rectangle with length l_c and width w_c whereas p is represented as a point. In such a RSM the CDRs are always limited by the vertices of the rectangle, compare Fig. 5.9. For this reason, it is not necessary to consider the edges of the rectangle geometry to determine CDRs and to replace the rectangle by its four vertices, i.e., points, in order to determine the collision direction thresholds for a rectangle-to-point RSM.

5.7.1 *Using the Normalized Transformation*

In a normalized rectangle-to-point RSM, according to Fig. 5.9, c moves from the origin at starting time t_0 to a future position at time t_n, $n \in \mathbb{R}$. For each vertex point i_c, $i \in \{A, B, C, D\}$ a point-to-point RSM is created, see Fig. 5.10. The resulting point-to-point RSMs are used as the input (c, p, i_c) for algorithm 6 to determine the up to two direction ranges $\mathbf{d}_k$, $k \in \{1, 2\}$.

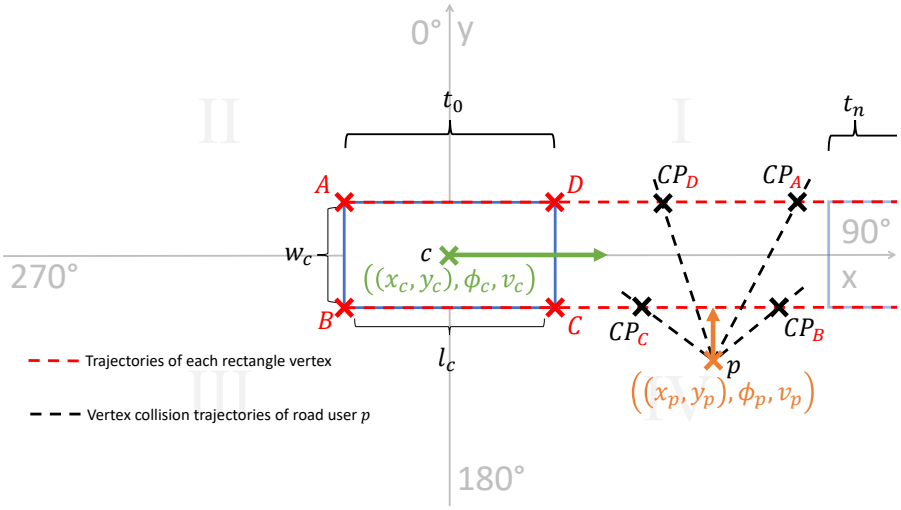
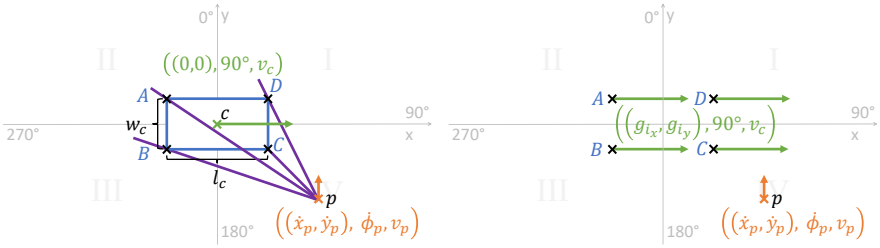


Fig. 5.9: Determining the collision points (CP) between the linear movement trajectories of a rectangle and a point



(a) Subdividing c'a rectangle into four vertex points

(b) Using each vertex point for collision direction range calculation

Fig. 5.10: DeterminingCDRs by subdividing c's rectangle geometry into four point-to-point RSMs

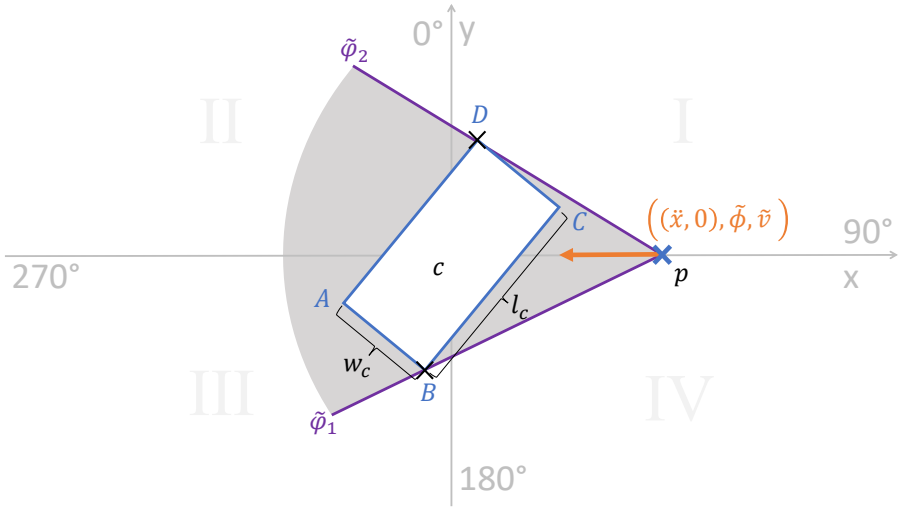


Fig. 5.11: Determining $\tilde{\varphi}_{1,2}$ in a rotated relative rectangle-to-point RSM

5.7.2 *Using the Rotated Relative Transformation*

Consider a rotated relative rectangle-to-point RSM, as shown in Fig. 5.11. To determine the CDRs in a rotated relative rectangle-to-point RSM, the algorithm 7 from Section 5.5.2 is used. The algorithm internally determines the greatest angle between the position of road user p (who is always located at the x-axis) and all four vertices of the rectangle of road user c. The resulting two $\tilde{\varphi}_{1,2}$ are then calculated backwards (Section 5.4) and filtered (Section 5.6) to determine the up to two direction ranges $\mathbf{d}_k$, $k \in \{1, 2\}$.

5.8 CIRCLE-TO-POINT/POINT-TO-CIRCLE COLLISION DIRECTION RANGES ALGORITHM

To determine the CDRs in a circle-to-point RSM, the two tangential points P_1 and P_2 from the tangent lines between the point to the circle are used, see Fig. 5.12. Let (x, y) be the position of a tangential point P_i, $i \in \mathbb{N}$. To calculate (x, y), a circle M_P with a radius r_P is constructed so that p, c, P_1 and P_2 are tangential points of M_P, as shown in Fig. 5.13. From the positions of p, P_1 and c, a right-angled triangle can be constructed. Using Thales's theorem [5.1], it is possible to calculate (x, y) from the distance equations of c and M_P. According to the Fig. 5.13, the distance from c to P_1 is r_c and thus x can be calculated using the distance equation, given in section 5.8,

$$\begin{aligned} c: \quad 0 &= (x - x_c) + (y - y_c)^2 - r_c^2 \\ \iff \quad x &= \pm\sqrt{r_c^2 - y^2} \end{aligned} \tag{5.14}$$

that can be simplified assuming that the RSM is normalized and thus c is at the origin $(0, 0)$.

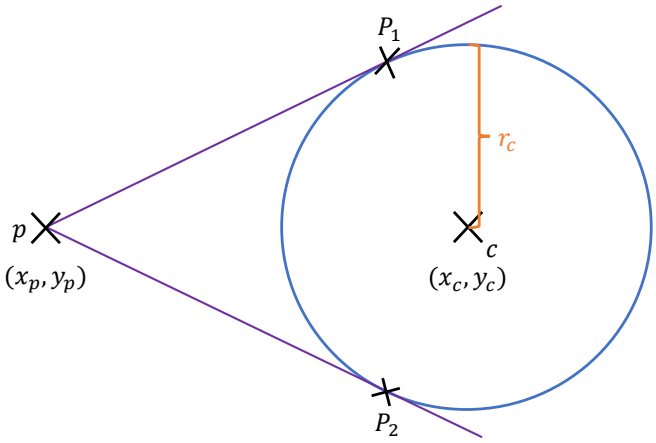


Fig. 5.12: Tangential points P_1, P_2 from p to the circle of c

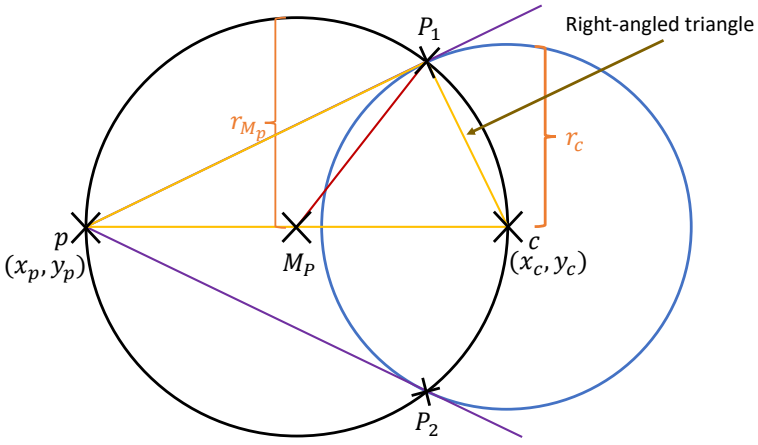


Fig. 5.13: Thales's theorem for tangential point calculation

The distance between p and c is $2r_{M_P}$ and so the radius r_P can be calculated in the same way, as given in section 5.8.

$$\begin{aligned} 2r_{M_P} &= \sqrt{(x_c - x_p)^2 + (y_c - y_p)^2} \\ \iff \quad r_{M_P}^2 &= \frac{1}{4}\left(x_p^2 + y_p^2\right) \end{aligned} \tag{5.15}$$

The position of M_P can be calculated from the distance between M_P and P_1, as given in (5.16),

$$\left(M_{P_x}, M_{P_y}\right) = \frac{1}{2}\left(x_p + x_c,\ y_p + y_c\right) = \frac{1}{2}\left(x_p,\ y_p\right) \tag{5.16}$$

using the knowledge that M_P is positioned in the centre between p and c. By inserting eq. (5.16) in section 5.8, the distance equation between M_P and P_1 can be used to obtain a circle equation that has the parameters x, y, x_p and y_p, as given in section 5.8.

$$\begin{aligned} M_P: \quad & \frac{1}{4}\left(x_p^2 + y_p^2\right) = \left(x - \frac{1}{2}x_P\right)^2 + \left(y - \frac{1}{2}y_p\right)^2 \\ \iff \quad & 0 = x^2 + y^2 - x x_p - y y_p \end{aligned} \tag{5.17}$$

Finally, the circle equations of M_P and c can be equated and rearrange them for $P_{1,2}$ to obtain the y-coordinates $P_{y_{1,2}}$, given in eq. (5.18)

$$\begin{aligned} & x^2 + y^2 - x x_p - y y_p = x^2 + y^2 - r^2 \\ \iff \quad & 0 = \left(x_p^2 + y_p^2\right) y^2 - 2r_c^2 y_p y + r_c^4 - x_p^2 r_c^2 \end{aligned} \tag{5.18}$$

The x-coordinates $x_{1,2}$ for $P_{1,2}$, are obtained by solving eq. (5.18) for y and inserting these y-coordinates in section 5.8. Eq. (5.18) can be solved either in a normalized RSM using the pq-formulae or directly in a relative RSM as detailed in the following two subsections, respectively.

5.8.1 *Using the Normalized Transformation*

Using the normalized RSM transformation eq. (5.18) can be solved using the pq-formula (eq. (5.19))

$$\Rightarrow \frac{p}{2} = -\frac{r_c^2 \dot{y}_p}{\dot{x}_p^2 + \dot{y}_p^2},\ q = \frac{r_c^4 - \dot{x}_p^2 r_c^2}{\dot{x}_p^2 + \dot{y}_p^2} \tag{5.19}$$

to determine the y-coordinates of the tangential points $P_{y_{1,2}}$ from $\frac{p}{2}$ and q, as given in eq. (5.20).

$$\begin{aligned} \Rightarrow P_{y_{1,2}} &= \frac{r_c^2 \dot{y}_p}{\dot{x}_p^2 + \dot{y}_p^2} \pm \sqrt{\left(\frac{r_c^2 \dot{y}_p}{\dot{x}_p^2 + \dot{y}_p^2}\right)^2 - \frac{r_c^4 - \dot{x}_p^2 r_c^2}{\dot{x}_p^2 + \dot{y}_p^2}} \\ &= \frac{r_c^2 \dot{y}_p}{\dot{x}_p^2 + \dot{y}_p^2} \pm \sqrt{\frac{\dot{x}_p^2 r_c^2 \left(\dot{x}_p^2 + \dot{y}_p^2 - r_c^2\right)}{\left(\dot{x}_p^2 + \dot{y}_p^2\right)^2}} \end{aligned} \tag{5.20}$$

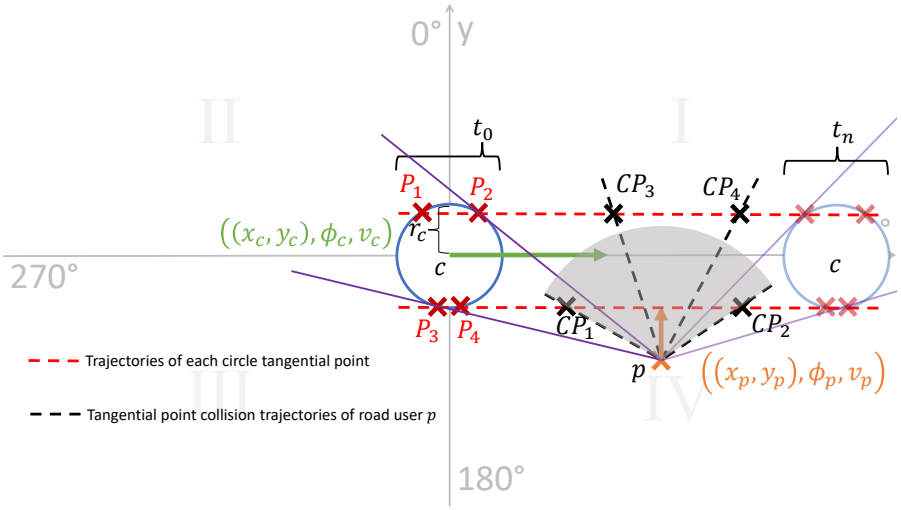
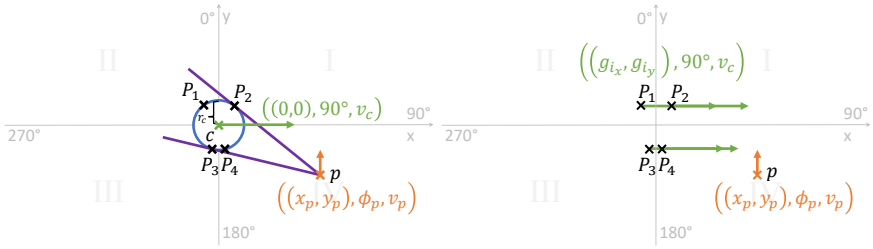


Fig. 5.14: Determining the collision points (CP) between the movement trajectories of tangential points of a circle and a point



(a) Tangential points for a normalized circle-to-point RSM

(b) Using the tangential points for direction range calculation

Fig. 5.15: Subdividing the circle geometry of c into four points in normalized cricle-to-point RSM

Due to the ambiguity of $\pm\sqrt{r_c^2 - \dot{y}^2}$ this results in four tangential points P_i, $i \in \{1, 2, 3, 4\}$ instead of two. As shown in Fig. 5.14, P_1 and P_2 are symmetric to each other on the circle's y-axis, as well as P_3 and P_4. This means that P_1 and P_2 have the same movement trajectory, as well as P_3 and P_4.

Consider the normalized circle-to-point RSM, as shown in Fig. 5.15a. To calculate the collision direction threshold values ϕ_j that lead to collisions with P_i, the circle geometry of c is subdivided into the four tangential points i_c, $i \in \{P_1, P_2, P_3, P_4\}$. The normalized point-to-point direction range component is used to calculate the CDR. For this, the SubRec algorithm 6 from Section 5.5 with the input (c, p, i_c) is used, to determine the up to two direction ranges $\mathbf{d}_k$, $k \in \{1, 2\}$.

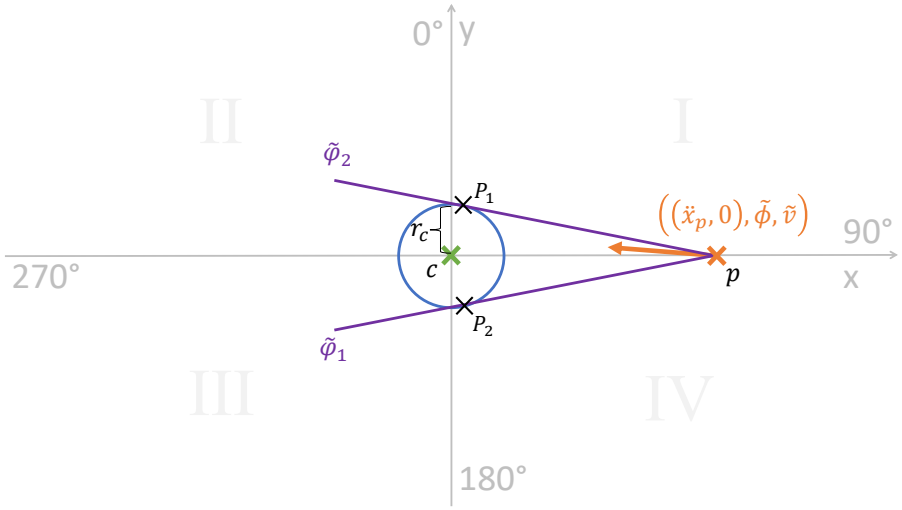


Fig. 5.16: Determining $\tilde{\varphi}_1$ and $\tilde{\varphi}_2$ from a rotated relative circle-to-point RSM

5.8.2 *Using the Rotated Relative Transformation*

Using the rotated relative RSM transformation, it is possible to directly calculate $P_{y_{1,2}}$, as given in eq. (5.21),

$$\begin{aligned} & (\ddot{x}_p - \dot{x}_c)(x - \dot{x}_c) + (\ddot{y}_p - \dot{y}_c)(y - \dot{y}_c) = r_c^2 \\ \iff \quad & P_{y_{1,2}} = \pm\sqrt{\frac{r_c^2\left(\ddot{x}_p^2 - r_c^2\right)}{\ddot{x}_p^2}} \end{aligned} \tag{5.21}$$

without using the pq-formula since $\dot{x}_c = \dot{y}_c = \ddot{y}_p = 0$. Furthermore, due to the restricted movement of p in a rotated relative RSM, the result is limited to only two tangential points because the x-coordinates of P_i can only be in the I. and IV. quadrant of the RSM coordinate system, compare Fig. 5.16, and thus $P_{x_{1,2}} = \sqrt{r_c^2 - P_{y_{1,2}}^2}$. P_1 and P_2 are then be used in the SubRec algorithm 7 from Section 5.5.2 with $(c, p, \{P_1, P_2\})$ as input to determine the up to two direction ranges $\mathbf{d}_k$, $k \in \{1, 2\}$.

5.9 RECTANGLE-TO-CIRCLE COLLISION DIRECTION RANGES ALGORITHM

Consider a rectangle-to-circle RSM in which c is represented as a rectangle with length l_c and width w_c and p as a circle with radius r_c, as shown in Fig. 3.12. The rectangle has four vertices, refereed to as i_c, $i \in \{A, B, C, D\}$. Similarly to the rectangle-to-point CDRs algorithm from Section 5.7, i_c can be used as components for the SubRec algorithms to calculate collision direction thresholds for p. When a circle with a radius r_p "collides" with one of the rectangle vertices, i.e. there is a tangential point between the circle and the the

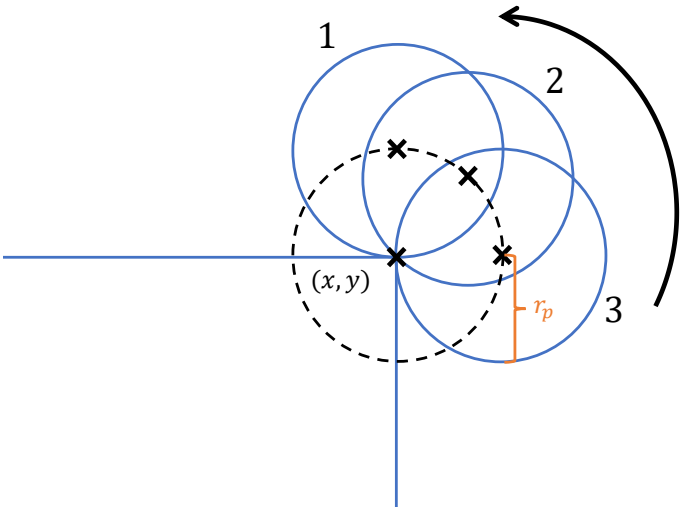


Fig. 5.17: Collision of a circle with a vertex of a rectangle

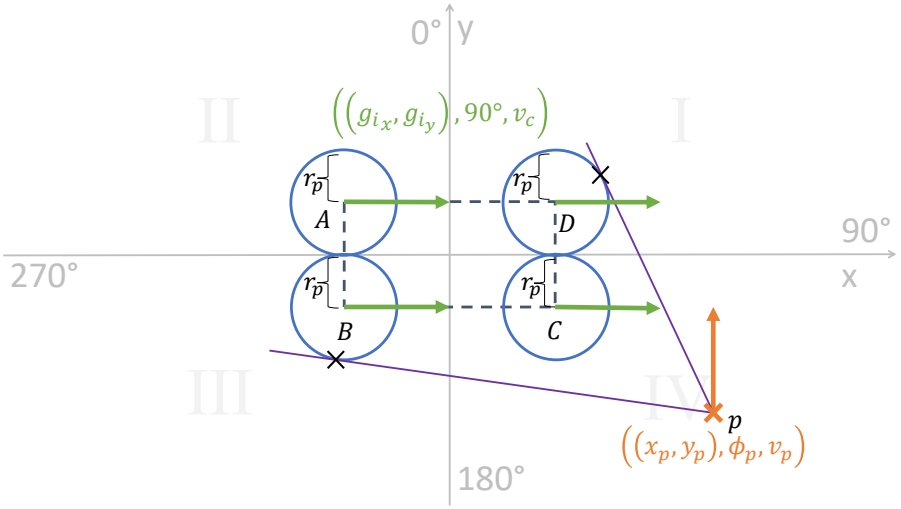


Fig. 5.18: Determining tangent intersection points for each circle of c's expanded rectangle corners

vertex, as shown in Fig. 5.17, the distance from the position of the circle to the vertex is always r_p. Thus it is equivalent to replace the vertex with a circle with the radius r_p, located at the position (x, y), and model p as a point. Accordingly, the rectangle geometry of c can be replaced by the four i_c components which are then expanded into circles with the radius r_p, and p is reduced to a point.

5.9.1 *Using the Normalized Transformation*

The rectangle geometry of c is subdivided at the vertices into four separate circles g_i, $g_i \in \{A, B, C, D\}$ with radius r_p which are located at the positions (A_x, A_y), (B_x, B_y), (C_x, C_y), $(D_x D_y)$, respectively, as shown in Fig. 5.18. p is reduced to a point. It is possible to use

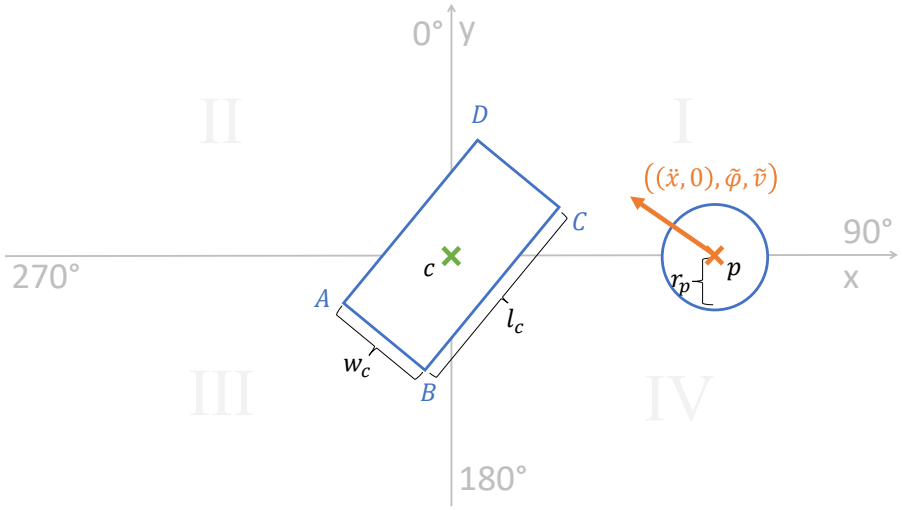
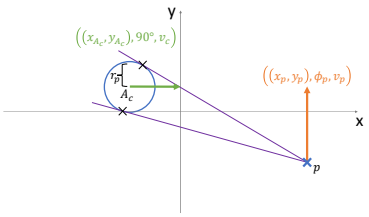
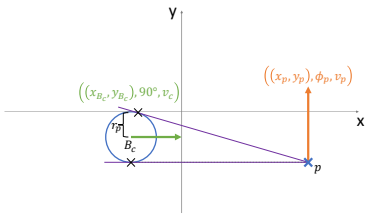


Fig. 5.20: Rotated relative rectangle-to-circle RSM

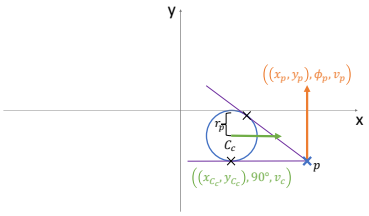
the normalized circle-to-point component four times, like shown in Fig. 5.18, in order to calculate direction ranges for a rectangle-to-circle scenario. Using the the SubRec algorithm 6 from Section 5.5.1 with the input $(c, p, \{g_i\})$, the up to two direction ranges $\mathbf{d}_k$, $k \in \{1, 2\}$ can be obtained.



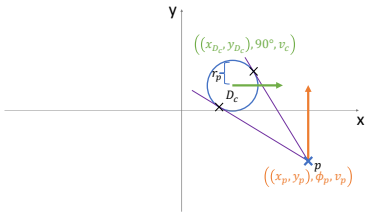
(a) Circle-to-point scenario for A_c and p



(b) Circle-to-point scenario for B_c and p



(c) Circle-to-point scenario for C_c and p



(d) Circle-to-point scenario for D_c and p

Fig. 5.19: Determining collision direction ranges for p by subdividing c into circles A_c to D_c with radius r_p to reuse the normalized circle-to-point component

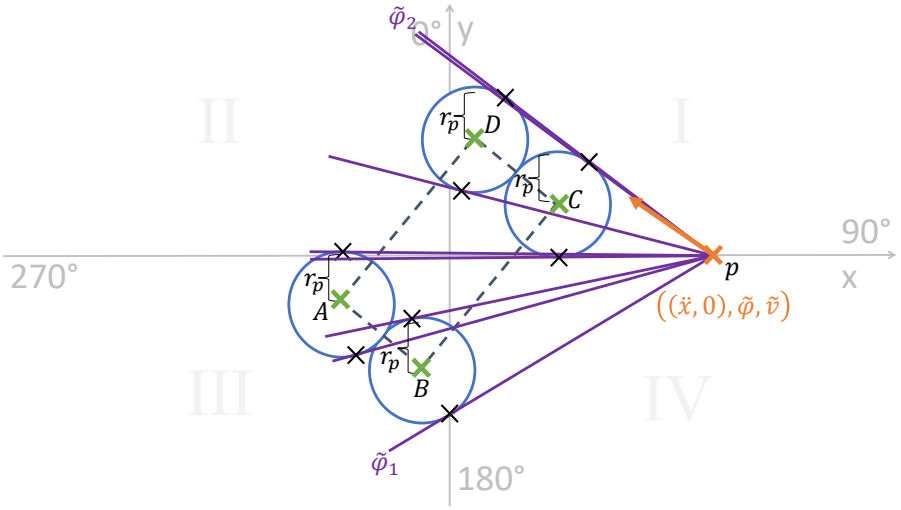


Fig. 5.21: Determining direction thresholds $\tilde{\varphi}_1$ and $\tilde{\varphi}_2$ from a rotated relative rectangle-to-circle RSM

5.9.2 *Using the Rotated Relative Transformation*

Consider a rotated relative rectangle-to-circle RSM, like shown in Fig. 5.20. The rectangle geometry of c is subdivided into four separate circles i_c, $i \in \{A, B, C, D\}$ with radius r_p which are located at the positions (A_x, A_y), (B_x, B_y), (C_x, C_y), $(D_x D_y)$, respectively, like shown in Fig. 5.21. p is reduced to a point. Instead of reusing the relative circle-to-point component as for the normalized transformation, the relative collision direction thresholds $\tilde{\varphi}_k$, $k \in \mathbb{N}$ can be determined more efficiently by using the tangential points between p and each circle i_c, $i \in \{A, B, C, D\}$. From all $\tilde{\varphi}_k$, the lowest $\tilde{\varphi}_{min}$ and the highest $\tilde{\varphi}_{max}$ are determined. For this, $\tilde{\varphi}_{min}$ and $\tilde{\varphi}_{max}$ are be used in the SubRec algorithm 7 from Section 5.5.2 with $(c, p, \{\tilde{\varphi}_{min}, \tilde{\varphi}_{max}\})$ as input to determine the up to two direction ranges $\mathbf{d}_k$, $k \in \{1, 2\}$.

REFERENCES

[5.1] T. L. Heath, *A History of Greek Mathematics: From Thales to Euclid*. Oxford, 1921. [Online]. Available: https://archive.org/details/cu31924008704219 (cit. on p. 79).

6 REQUIRED MOVEMENT RECOGNITION ACCURACY

In the preceding chapters the two challenges "How to precisely detect collisions in a *cooperative* VRU Collision Avoidance System (cooperative VRU CAS)?" (C I) and "How to quantify the collisions detection performance of a VRU Collision Avoidance System (VRU CAS) independent to a certain architecture?" (C II) were solved. In this Chapter the research question "What is the minimum required accuracy for Vulnerable Road User (VRU) movement recognition on mobiles devices for a certain collision detection performance of a cooperative VRU CAS?" will finally be answered. For this, a simulator was developed based on Road Space Models (RSMs) from Section 3.1. For the simulations, a cooperative VRU CAS, like proposed in [6.1] is assumed, in which the car and a pedestrian are equipped with a mobile devices, e.g. smartphones: The mobile devices of the pedestrian recognizes the pedestrian's movement and sends the movement information to the mobile device in the car via a wireless network, e.g. Wireless Local Area Network (WLAN) or Long Term Evolution (LTE). The mobile device at the car then performs a collision detection as detailed in Chapter 3 using its own and the pedestrian's movement vector. For simplification, linear movement is assumed, i.e., no turning or accelerating for both the car and the pedestrian. Furthermore, communication delays are neglected, since in the case of linear movement, they can be corrected by linear movement extrapolation using a sender timestamp.

The first step for conducting the simulations is to identify collision scenarios which are representative for collisions between pedestrians and vehicles. The aim is to find the smallest number of scenarios which, by slightly adjustment of the scenario parameters, cover most of the statistical relevant situations between pedestrians and vehicles. In this way, it will be possible to determine the requirements for cooperative VRU CASes using the minimum number of scenarios to cover a statistical relevant number of real collision situations between pedestrians and vehicles. For this, in Section 6.1, most relevant collision scenarios between vehicles and VRUs are presented. The section is based on the results from [6.2], who conducted a analyses on the German In-Depth Accident Study (GIDAS) database[1]. Based on the findings from scenarios from Section 6.1, in Section 6.2 standardized collision scenarios from the European New Car Assessment Programme (Euro NCAP) are then selected for the simulation based evaluations in the remainder

1 'German In-Depth Accident Study (GIDAS)'. [Online]. https://www.gidas.org/, accessed 13.03.2021

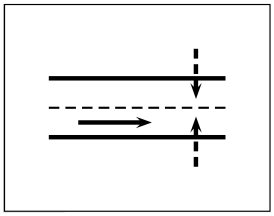
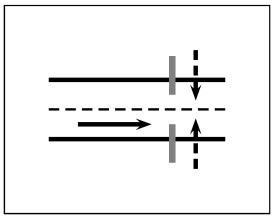
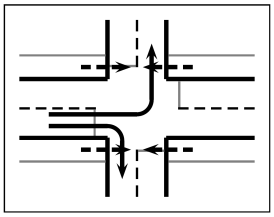
of this chapter. The modelling of the selected scenarios in the RSM is detailed in Section 6.3.

The first evaluation in Section 6.4 consists of analysing the sole impact of position, speed, and direction error on P_{MA} and P_{FA} separately. Afterwards in Section 6.5, the mutual dependency between position, direction and speed errors and their influence on the P_{MA} and P_{FA} thresholds is investigated. From this results, in Section 6.6, a indicator called "error volume maximization" is introduced and used to find the mutual maximums for position, direction and speed errors for given P_{MA} and P_{FA} values. Based on the error volume maximization indicator the *minimum required accuracy for movement recognition* for $P_{\{MA,FA\}} = \{0.1, 0.05, 0.01\}$ are determined in Section 6.7. In Section 6.8 the investigation is extended to scenarios with different impact positions and different street crossing angles of the pedestrians. Finally, in Section 6.9, it is analysed how shorter Time to Collision (TTC) values, i.e. smaller distances to the collision point, influence the required movement recognitions accuracy. The results are summarized in Section 6.10.

6.1 GERMAN IN-DEPTH ACCIDENT STUDY (GIDAS)

In Germany, every traffic accident with severe injuries or fatalities is recorded in the GIDAS. The three most frequent collision scenarios between vehicles and pedestrians are shown in Table 6.1. The results show that 29.7% of all accident scenarios between pedestrians and vehicles occur when a pedestrian crosses the street with visual obstruction (Non-Line-of-Sight (NLOS) between vehicle and pedestrian), like a parked vehicle. Accident scenarios in which a pedestrian crosses the street without visual obstruction (Line-of-Sight (LOS) between vehicle and pedestrian) comprise 49.4%. The third most frequent scenarios are at crossings when a vehicle turns into the street and represents 7.6% of all collision scenarios. In this work we will use the first and second most frequent scenario as the basis of our investigation. Since the only difference between the first and the second scenario is the LOS condition, the results for the NLOS scenario can be transferred to the LOS scenario without losing any generality. Therefore, the crossing scenario covers approximately 80% of all accident scenarios between pedestrians and vehicles. Such crossing scenarios are also considered in the test protocol Autonomous Emergency Braking (AEB) VRU systems of the Euro NCAP. For the purpose of comparability, all of the following investigations will therefore be based on the standardized Euro NCAP scenarios, which are detailed in the following Section 6.2.

Table 6.1: The most frequent collision scenarios between vehicles and pedestrians from the German In-Depth Accident Study database [6.2]

Frequency	Overview	Description
49.4%		For pedestrians, nearly every second accident happens when a pedestrian crosses the street with a LOS between the vehicle and the pedestrian.
27.7%		The second most common accident is similar to the first one, but in this case the pedestrian is obscured, e.g. by a parked car.
7.6%		The third most common accident happens when a vehicle is turning into a street and the pedestrian is crossing this street.

6.2 THE EUROPEAN NEW CAR ASSESSMENT PROGRAMME

The European New Car Assessment Programme (Euro NCAP) was developed in 1997 in the United Kingdom with the aim to provide consumers with a safety performance assessment for the majority of the most popular cars in the European Union [6.3], [6.4].

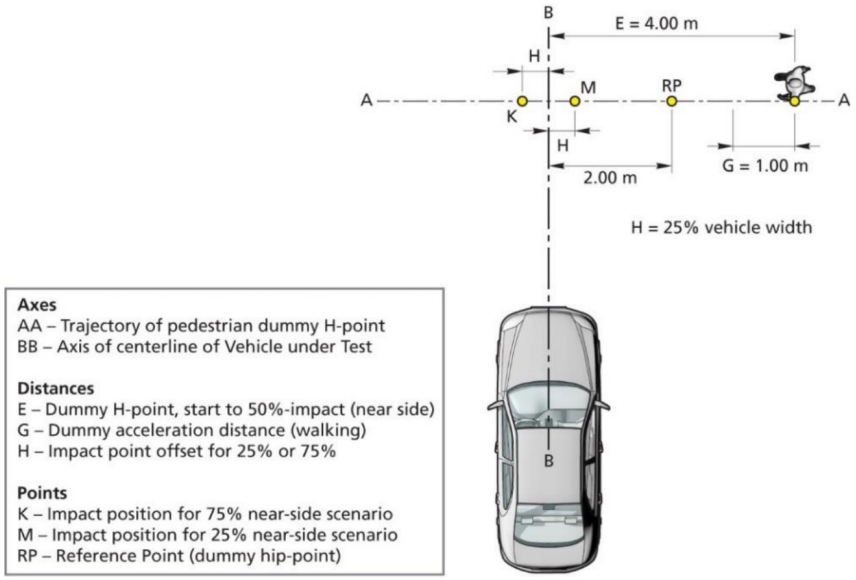
From the Euro NCAP the "Test protocol for AEB VRU systems" in version 1.0.1 was released in 2015 [6.5]. At the time writing this dissertation the latest available version is 3.0.3 from June 2020 [6.6]. Accordingly, within the scope of this dissertation all referenced content from the Test protocol for AEB VRU systems refers to version 3.0.3. In the Test protocol for AEB VRU systems nine test scenarios for assessing the performance of "AEB pedestrian systems", prefix "Car-to-Pedestrian" (CP), are defined. Besides

- two *longitudinal* collision scenarios, called "Car–to–Pedestrian Longitudinal Adult" (CPLA), "in which a vehicle travels forwards towards an adult pedestrian walking in the same direction in front of the vehicle",
- a *turning* collision scenario, called "Car-to-Pedestrian Turning Adult" (CPTA), "in which a vehicle turns towards an adult pedestrian crossing its path walking from the opposite direction at an intersection (before the Vehicle under Test (VUT) made the turn)", and
- two *reverse* collision scenario, called "Car-to-Pedestrian Reverse Adult" (CPRA) "in which a vehicle travels rearwards towards an adult pedestrian standing still, facing sideways"

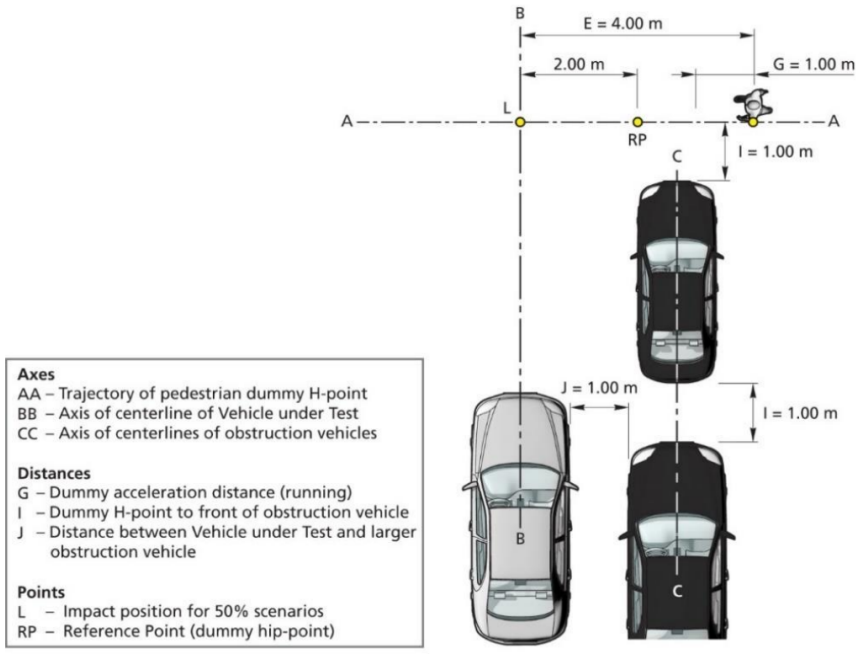
four *crossing* collision scenarios "in which a vehicle travels forwards towards a pedestrian crossing its path" are defined. Those collision scenarios consists of

1. One scenario in which an adult pedestrian approaches from the vehicle's left hand side, called Car-to-Pedestrian Farside Adult (CPFA),
2. Two collision scenarios considering an adult pedestrian approaching from the vehicle's right hand sight, called Car-to-Pedestrian Nearside Adult (CPNA), see Fig. 6.1a,
3. One collision scenario considering a child obstructed by parked cars, called Car-to-Pedestrian Nearside Child Car-to-Pedestrian Nearside Child (CPNC), see Fig. 6.1b.

Note that the primarily difference between the the CPNA and CPNC scenarios is the initial visual obstruction between the vehicle and the pedestrian due to the parked vehicles.



(a) "Car-to-Pedestrian Nearside Adult 25/75" (CPNA-25 and CPNA-75)



(b) "Car-to-Pedestrian Nearside Child 50" (CPNC-50)

Fig. 6.1: Official scenario sketches from the 'Test Protocol - AEB VRU systems (Version 3.0.3).' [6.6]

6.2.1 *Investigated Scenarios*

In respect to the results from the GIDAS analysis from Section 6.1, the assessments in this dissertation are based on the three crossing collision scenarios:

- Car-to-Pedestrian Nearside Adult 25% (CPNA-25)
- Car-to-Pedestrian Nearside Child 50% (CPNC-50)
- Car-to-Pedestrian Nearside Adult 75% (CPNA-75)

The number in the scenario names refers to the percentage of the vehicle's frontal structure width that will strike the VRU when no braking action is applied. The CPFA is neglected due to its symmetry in contrast to the other collision scenarios. As it can be recognized in the scenario names, it is distinguished between children and pedestrians which may be due to the fact that a child might be harder to detect in NLOS conditions for car-based VRU CAS. Since the aim in this dissertation is to investigate the required minimum movement recognition accuracy for cooperative VRU CAS, which are not influenced by NLOS conditions, this distinction between the children and adults is neglected. The general CPNC-50 scenario setup is shown in Fig. 6.1b. Since the only difference between the CPNA and CPNC scenarios is the LOS condition, the Car-to-Pedestrian Nearside scenarios should be representative for approximately 80% of all accident scenarios between pedestrians and vehicles.

To investigate the required movement recognition accuracy for non-collision scenarios, i.e., no collision should be detected, the same scenario with a different setup for the vehicle's starting position is used, so that the pedestrian crosses the street before the vehicle arrives at the collision points.

6.3 SCENARIO MODELLING

The CPNC-50 scenario is modelled in a rectangle-to-circle RSM as follows, see Fig. 6.2:

The car's velocity is set parallel on the x-axis with a speed of $50\,\frac{km}{h} = 13.\overline{8}\,\frac{m}{s}$. Since the car's position is given by the geometric centre of the rectangle, the initial position of the car is set to $\mathbf{r}_{c_0} = (-40\,m - \frac{l_c}{2} - r_p, 0\,m)$, where $\frac{L_c}{2} = 2\,m$ is the half of the car length and $r_p = 0.5\,m$ is the radius of the pedestrian. This effects that the car's front $(x_c + 2\,m)$ will be at the impact position $(-0.5\,m,\ 0\,m)$ after

$$t_c = \frac{40m}{50km/h} = 2.88\,s$$

For finding P_{MA} and P_{FA}, two cases for the pedestrian's initial position $\mathbf{r}_p$ are used:

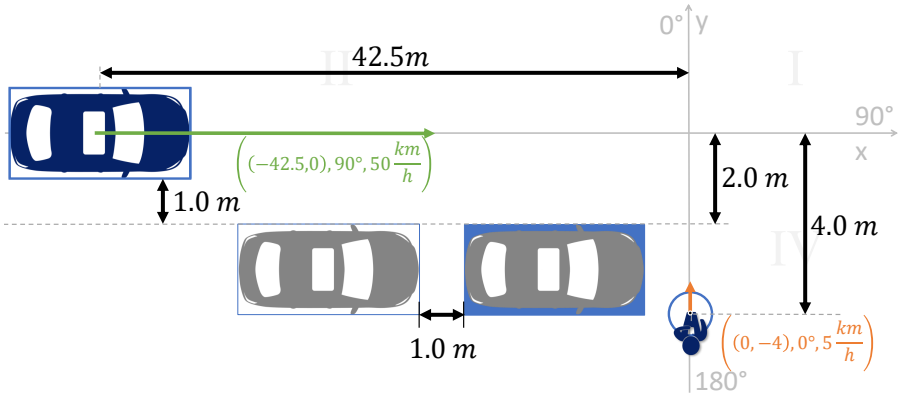


Fig. 6.2: Rectangle-to-circle RSM of the Euro NCAP CPNC-50 scenario.

1. P_{MA}: The pedestrian's initial position is set to $\mathbf{r}_{p_{P_{MA}}} = (0\,m, -4\,m)$ and its velocity is set parallel to the y-axis with a speed of $5\,\frac{km}{h} = 1.3\overline{8}\,\frac{m}{s}$,

$$\mathbf{r}_{p_{P_{MA}}}(t) = 1.3\overline{8}\,\frac{m}{s} \cdot t \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0\,m \\ -4\,m \end{pmatrix}$$

which causes that the pedestrian will be at the origin (0.0) of the coordinate system after

$$t_p = \frac{4m}{5km/h} = 2.88\,s$$

The car, with its starting position $\mathbf{r}_{c_0} = (-42.5\,m, 0\,m)$ and speed $v_c = 50\,\frac{km}{h}$, will then be at the pedestrian circle's circumference, i.e. the impact position $(-0.5\,m, 0\,m)$, see Fig. 6.3. Thus, there will be a collision after $TTC = t_c = t_p = 2.88\,s$.

2. P_{FA}: For the non-collision scenario the pedestrian's initial position is changed to $\mathbf{r}_{p_{P_{FA}}} = (0\,m, -1.5\,m)$, resulting in

$$\mathbf{r}_{p_{P_{FA}}}(t) = 1.3\overline{8}\,\frac{m}{s} \cdot t \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0\,m \\ -1.5\,m \end{pmatrix}$$

which results in a distance between the car and the pedestrian $\geqslant 1.0\,m$ at any time, i.e., no collision will happen.

As shown in [6.7], the errors in recognizing the movement vector, using smartphone-grade sensors, can be approximated by normal distributions. Accordingly, zero-mean normally distributed random variables are used to model the random sensor errors for position, direction and speed measurements. In detail, the position error was modelled as the bivariate random variable $\mathbf{X} \sim \mathcal{N}(\mu_X, \Sigma_X)$ of the lateral and longitudinal position measurement errors with $\mu_X = (0, 0)$ and

$$\Sigma = \begin{pmatrix} \sigma_{XY} & 0 \\ 0 & \sigma_{XY} \end{pmatrix}$$

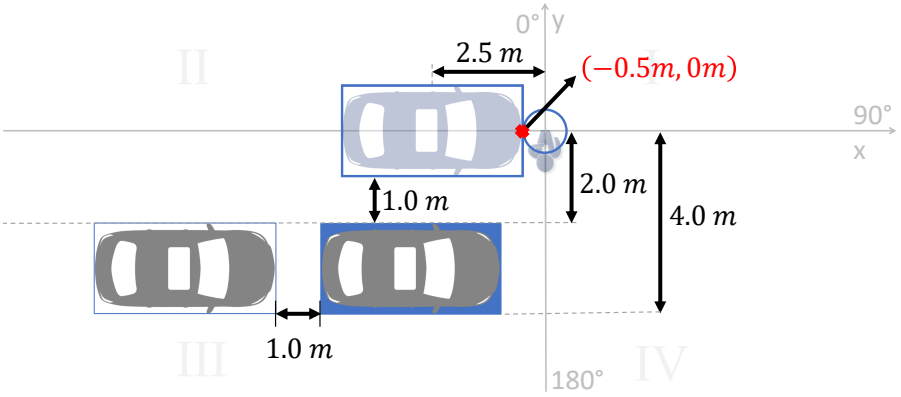


Fig. 6.3: Impact position in the Rectangle-to-circle RSM of the Euro NCAP CPNC-50 scenario.

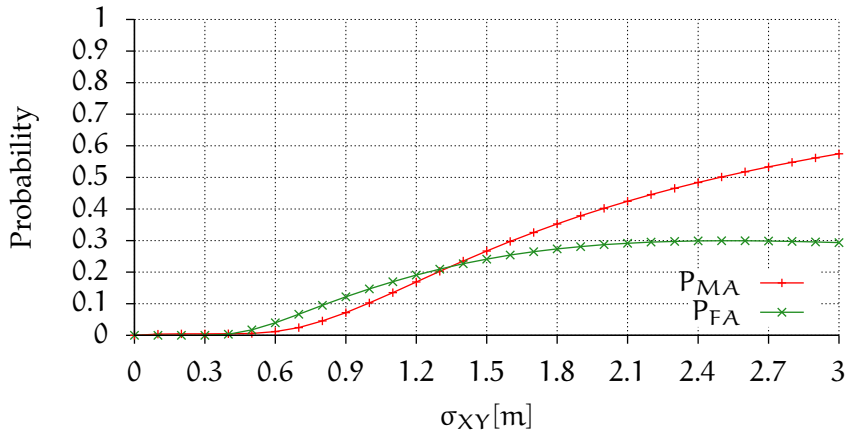
so that the position error in the lateral and longitudinal direction of the pedestrian has the same magnitude. Furthermore, $V \sim \mathcal{N}(0, \sigma_V)$ is the normally distributed random variable of the speed measurement error and $\Phi \sim \mathcal{N}(0, \sigma_\Phi)$ is the normally distributed random variable of the direction measurement error. In this way the standard deviations of the chosen normal distributions model the Root-Mean-Square Error (RMSE) for recognizing the VRU movement on a mobile device. Those random variables were used to create the set of possible movement vectors of the pedestrian together with their respective probability according to eq. (4.16). The set of movement vector probabilities is used to determine P_{MA} and P_{FA} according to eqs. (4.8) and (4.9) in Section 4.4.2 using the Collision Direction Ranges (CDRs) based on eq. (5.3) from Chapter 5.

6.4 SOLE INFLUENCE OF POSITION, SPEED AND DIRECTION ERRORS ON P_{MA} AND P_{FA}

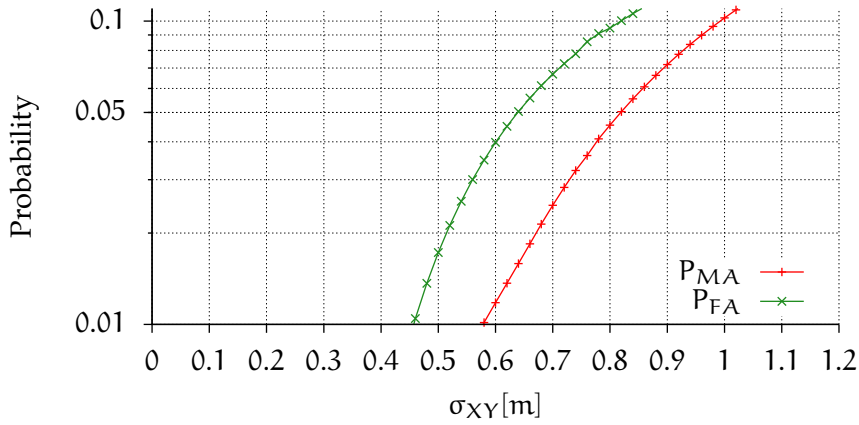
In this section the sole impact of σ_{XY}, σ_V and σ_Φ separately on P_{MA} and P_{FA} in the CPNC-50 scenario is investigated, i.e., considering only one component of the movement vector of the pedestrian as inaccurate and the other components as error free. The P_{MA} and P_{FA} values were determined for the current ground-truth positions of the road users specified in Section 6.3. Accordingly, the following results refer to a TTC of 2.88 s in case of a collision scenario for P_{MA}.

6.4.1 *Position Error (σ_{XY})*

Fig. 6.4 shows the influence of the position error standard deviation σ_{XY} on P_{MA} (red line with plus signs) and P_{FA} (green dashed line with crosses). The standard deviation σ_{XY} of the position error was varied from 0 m to 3 m by 0.1 m steps. Within this interval, P_{MA} continuously increases from 0 to ≈ 0.57 while P_{FA} reaches a maximum

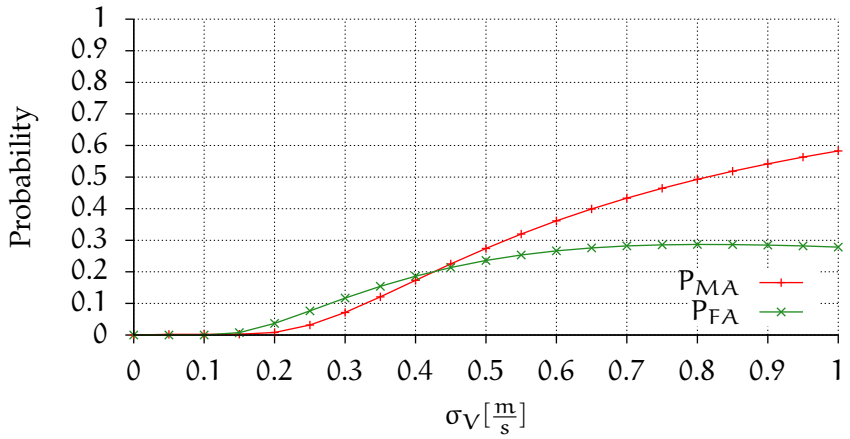


(a) Linear overview for error magnitudes $0\,m \leqslant \sigma_{XY} \leqslant 3\,m$

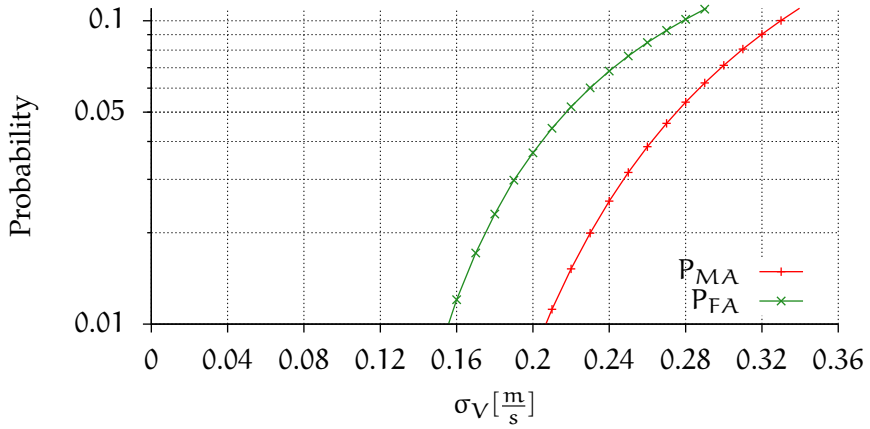


(b) Log-scale detail in the domain of $0.01 \leqslant P_{\{MA,FA\}} \leqslant 0.1$

Fig. 6.4: Sole impact of position error (σ_{XY}) only: P_{MA} and P_{FA} for rectangle-to-circle RSM



(a) Linear overview for speed error standard deviations $0\,\frac{m}{s} \leqslant \sigma_V \leqslant 1\,\frac{m}{s}$



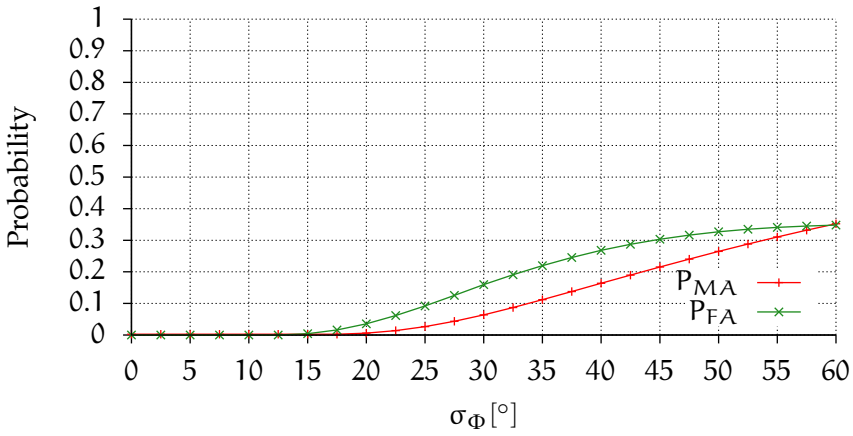
(b) Log-scale detail in the domain of $0.01 \leqslant P_{\{MA,FA\}} \leqslant 0.1$

Fig. 6.5: Sole impact of speed error (σ_V) only: P_{MA} and P_{FA} for rectangle-to-circle RSM

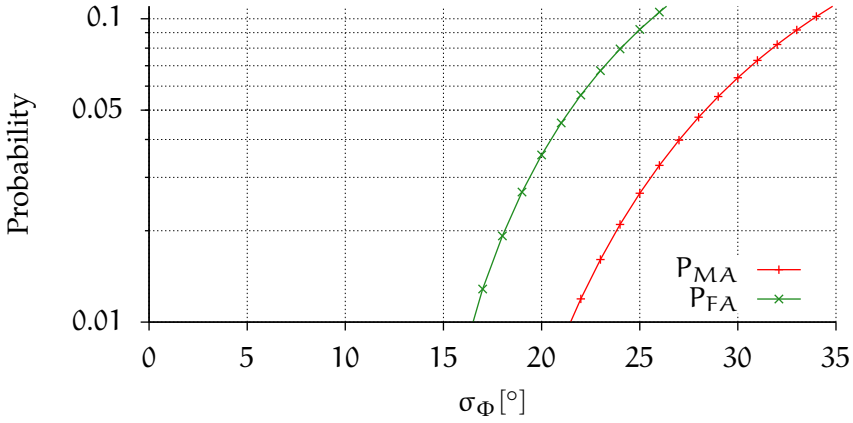
of ≈ 0.30 for $\sigma_{XY} = 2.6\,m$ and decreases for greater values. Actually, for very large σ_{XY}, P_{FA} converges towards 0. This is due to the fact, that for increasing σ_{XY}, an increasing proportion of trajectories do not lead to collisions any more [6.8]. Additionally, it can be noted that $P_{FA} \geqslant P_{MA}$ for values of $\sigma_{XY} < 1.35\,m$. For $P_{MA/FA} < 0.1$, the standard deviation needs to be $\sigma_{XY} < 0.82\,m$, and for $P_{MA/FA} < 0.01$ the standard deviation σ_{XY} must not exceed $0.46\,m$.

6.4.2 *Speed Error (σ_V)*

In Fig. 6.5, the influence of the speed error standard deviation on P_{MA} (red line with plus signs) and P_{FA} (green line with crosses) is shown. The speed error's standard deviation (σ_V) was varied from from $0\,\frac{m}{s}$ to $1\,\frac{m}{s}$ in $0.1\,\frac{m}{s}$ steps. Within this interval P_{MA} increases continuously from 0 to ≈ 0.58. P_{FA} reaches its maximum of 0.29 for



(a) Linear overview for direction error standard deviations $0° \leqslant \sigma_\Phi \leqslant 60°$



(b) Log-scale detail on $0.01 \leqslant P_{\{MA,FA\}} \leqslant 0.1$

Fig. 6.6: Sole impact of speed error (σ_Φ) only: P_{MA} and P_{FA} for rectangle-to-circle RSM

$\sigma_V = 0.81\,\frac{m}{s}$ and decreases for greater values of σ_V. Like for the position error standard deviation in Section 6.4.1, P_{FA} takes initially larger values than P_{FA} for values of $\sigma_V < 0.42\,\frac{m}{s}$. For $P_{MA/FA} < 0.1$, the standard deviation of the speed error must not exceed $0.28\,\frac{m}{s}$. For $P_{MA/FA} < 0.01$, a speed error standard deviation of $\sigma_V < 0.158\,\frac{m}{s}$ is required.

6.4.3 *Direction Error (σ_Φ)*

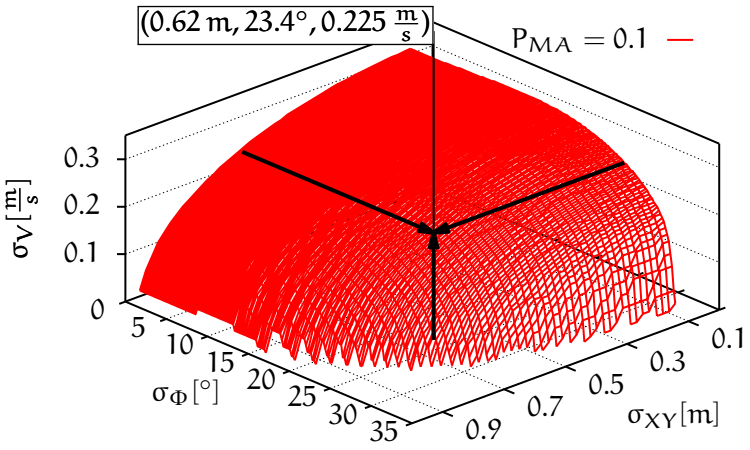
Fig. 6.6 shows the dependency of P_{MA} and P_{FA} on the direction error standard deviation σ_Φ. The value of σ_Φ was varied from 0° to 60° in 1° steps. Both P_{MA} and P_{FA} first remain close to 0 until $\sigma_\Phi \approx 15°$. The initial constant values of the curves are due to the collision direction range, see Chapter 5, of 111.8° in the CPNC-50 scenario. This direction effects that in the CPNC-50 scenario there is not only a collision for a

street crossing angle of 90° but also for street crossing angles between 37° to 149°.

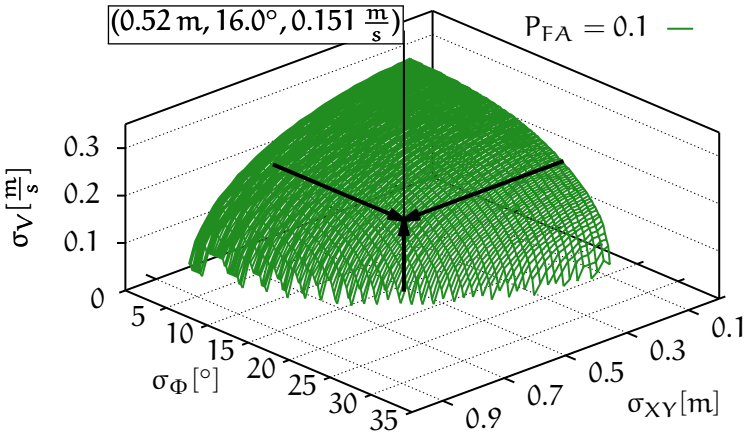
For $\sigma_\Phi < 60°$, P_{FA} takes higher values than P_{MA}. In the investigated domain of $0° \leqslant \sigma_\Phi \leqslant 60°$ P_{FA} continuously increases up to 0.35. In Fig. 6.6b, it can be seen that for $P_{\{MA,FA\}} < 0.1$, the standard deviation of the direction error has to be $\sigma_\Phi < 25.6°$ and for $P_{\{MA,FA\}} < 0.01$, it has to be $\sigma_\Phi < 16.5°$.

6.5 DEPENDENCE OF POSITION, SPEED, AND DIRECTION ERROR ON EACH OTHER

From the graphs in Figs. 6.4 to 6.6, the thresholds for the standard deviation of each, the position, the speed, and the direction error alone were identified to get $P_{\{MA,FA\}} < 0.1$. Now this investigation is extended for finding the mutual threshold values for position, speed, and direction errors when all three movement vector components are subject to recognition error. Using $k \in \{P_{MA}, P_{FA}\}$ as the performance indicator and $\tau \in \{0.1, 0.05, 0.01\}$ as the threshold for the performance indicator, sets $E_{k,\tau}$ of "error-tuples" are calculated. A set $E_{k,\tau}$ contains possible combinations of $(\sigma_{XY}, \sigma_\Phi, \sigma_V)$ that result in $k = \tau$. In Fig. 6.7a, the the values of $(\sigma_{XY}, \sigma_\Phi, \sigma_V)$ for $E_{P_{MA},0.1}$ are plotted. Likewise, Fig. 6.7b shows the values for $E_{P_{FA},0.1}$. The three axes in the three-dimensional coordinate system indicate the standard deviation of each, the position, the speed, and the direction error, respectively. The surface plots in the 3-dimensional coordinate system represents all possible combinations of $(\sigma_{XY}, \sigma_\Phi, \sigma_V)$ for getting $P_{MA} = 0.1$ and $P_{FA} = 0.1$, respectively. The nodes of the grids are the actual calculated error-tuples while the lines in between are interpolated.



(a) $P_{MA} = 0.1$



(b) $P_{FA} = 0.1$

Fig. 6.7: Mutual dependence on the magnitudes of σ_{XY}, σ_V, and σ_Φ

6.6 ERROR VOLUME MAXIMIZATION

As shown in Figs. 6.7a and 6.7b, the thresholds for the standard deviations of the position, the speed, and the direction error are dependent on each other. Therefore, an indicator is needed to quantify the "quality" of the combination. For this assessment the volume

$$vol = \sigma_{XY} \cdot \sigma_{\Phi} \cdot \sigma_{V} \tag{6.1}$$

of the error-tuples is used as the quality indicator which indicates how much "error volume" a given error-tuple encloses.

Calculating eq. (6.1) for all values in a $E_{k,\tau}$ set results in the new set

$$Vol_{k,\tau} = \{\sigma_{XY} \cdot \sigma_{\Phi} \cdot \sigma_{V} | (\sigma_{XY}, \sigma_{\Phi}, \sigma_{V}) \in E_{k,\tau}\} \tag{6.2}$$

which contains all volumes that the tuples $(\sigma_{XY}, \sigma_{\Phi}, \sigma_{V})$ in $E_{k,\tau}$ enclose. From the $Vol_{k,t}$ set, the tuple which encloses the maximum error volume

$$\hat{vol} = max(Vol_{k,\tau}) = \hat{\sigma}_{XY} \cdot \hat{\sigma}_{\Phi} \cdot \hat{\sigma}_{V} \tag{6.3}$$

in the 3-dimensional space is used as the balanced combination of $(\sigma_{XY}, \sigma_{\Phi}, \sigma_{V})$ for a given k and τ. In the following, the notation $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$ is used for referring to the error volume maximization tuple. For $P_{MA} = 0.1$ the volume maximizing values $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$ are $(0.62\,m, 23.4°, 0.225\,\frac{m}{s})$ and for P_{FA} it is $(0.52\,m, 16.0°, 0.151\,\frac{m}{s})$, indicated by the three black arrows in in Figs. 6.7a and 6.7b , respectively.

6.7 MINIMUM MOVEMENT RECOGNITION REQUIREMENTS FOR THE CPNC-50 SCENARIO

This section extends the investigation of finding the minimum required movement recognition accuracy in the CPNC-50 scenario based on the $\hat{vol}$ indicator for values of $P_{\{MA,FA\}} \in \{0.01, 0.05, 0.1\}$. For this, the scenario setup from Section 6.3 is used. For this setup the sets $Vol_{k,\tau}$, see Section 6.6, for $k \in \{P_{MA}, P_{FA}\}$ and $\tau \in \{0.1, 0.05, 0.01\}$ were calculated. The corresponding values of $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$ for the

Table 6.2: Accuracy requirements (maximum standard deviations of the error in position $\hat{\sigma}_{XY}$, direction $\hat{\sigma}_{\Phi}$, and speed $\hat{\sigma}_{V}$) for P_{MA}.

P_{MA}	$\hat{\sigma}_{XY}$	$\hat{\sigma}_{\Phi}$	$\hat{\sigma}_{V}$	P_{FA}
0.10	0.62	23.4	0.225	0.22
0.05	0.50	21.0	0.189	0.16
0.01	0.38	15.4	0.123	0.05

Table 6.3: Accuracy requirements (maximum standard deviation of the error in position $\hat{\sigma}_{XY}$, direction $\hat{\sigma}_{\Phi}$, and speed $\hat{\sigma}_{V}$) for P_{FA}.

P_{FA}	$\hat{\sigma}_{XY}$	$\hat{\sigma}_{\Phi}$	$\hat{\sigma}_{V}$	P_{MA}
0.10	0.52	16.0	0.151	0.03
0.05	0.36	14.2	0.142	0.02
0.01	0.26	11.6	0.104	0.01

Table 6.4: Accuracy requirements (maximum standard deviations of the error in position $\hat{\sigma}_{XY}$, direction $\hat{\sigma}_{\Phi}$, and speed $\hat{\sigma}_{V}$) for $P_{MA/FA}$.

$P_{\{MA,FA\}}$	$\hat{\sigma}_{XY}$	$\hat{\sigma}_{\Phi}$	$\hat{\sigma}_{V}$	P_{MA}	P_{FA}
$\leqslant 0.10$	0.52	16.0	0.151	0.03	0.10
$\leqslant 0.05$	0.36	14.2	0.142	0.01	0.05
$\leqslant 0.01$	0.26	11.6	0.104	0.01	0.01

different P_{MA} and P_{FA} thresholds are shown in the Tables 6.2 and 6.3, respectively. It can be seen from both Figs. 6.7a and 6.7b and Tables 6.2 and 6.3, that for all thresholds, P_{FA} requires a higher accuracy (i.e. smaller values for σ_{XY}, σ_{Φ}, and σ_{V}) than P_{MA}.

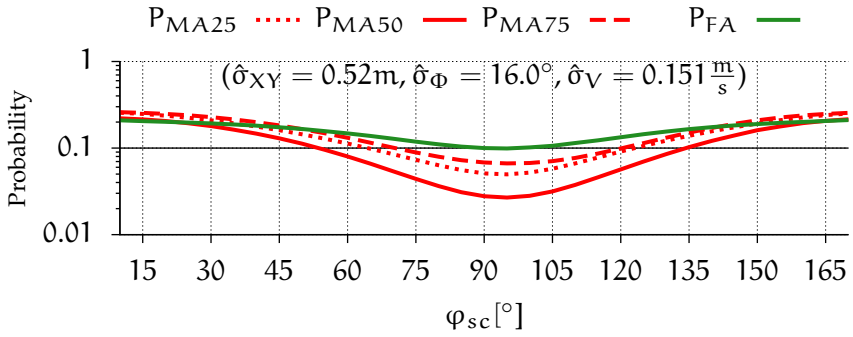
This has the effect that the volume maximization for a certain P_{MA} does not ensure that P_{FA} is also below the desired threshold, e.g. $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V}) = (0.62\,\mathrm{m}, 23.4°, 0.225\,\frac{\mathrm{m}}{\mathrm{s}})$ for $P_{MA} = 0.1$ will lead to $P_{FA} = 0.22$.

Accordingly, for finding values $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$ so that both P_{MA} and P_{FA} are below a desired threshold (indicated by $P_{\{MA,FA\}} < \tau$), the values $(\hat{\sigma}_{XY,P_{FA}}, \hat{\sigma}_{\Phi,P_{FA}}, \hat{\sigma}_{V,P_{FA}})$ for P_{FA} are decisive. The values of $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$ for $P_{\{MA,FA\}}$ are shown in Table 6.4. In the last two columns of this table, the resulting values for P_{MA} and P_{FA} after the combination are given.

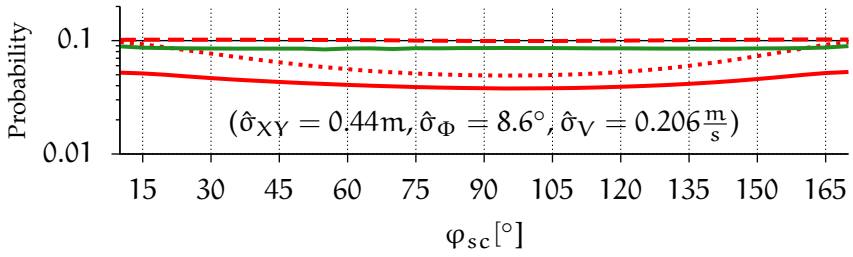
6.8 IMPACT POSITIONS AND CROSSING ANGLES

In the previous section, the minimum required movement recognition accuracy was investigated for a pedestrian in the CPNA-50 scenario to keep the probability of Missed and False Alarms below 0.1, 0.05, and 0.01. In this CPNA-50 scenario a pedestrian crosses the street with a crossing angle (φ_{sc}) of $\varphi_{sc} = 90°$. However, the numbers in the GIDAS statistics, see Section 6.1, which state that approximately 80% of collisions between pedestrians and vehicles occur when the pedestrian crosses the street, are not restricted to a street crossing angle of 90°.

Thus, the investigation from Section 6.6 is extended in this Section to arbitrary street crossing angles between $15° \leqslant \varphi_{sc} \leqslant 165°$. Additionally, the analysis of the Missed Alarm Probability is exten-



(a) $P_{MA\{25,50,75\}} \wedge P_{FA}$ for different φ_{sc} using $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$ for $P_{\{MA*,FA\}} \leqslant 0.1$



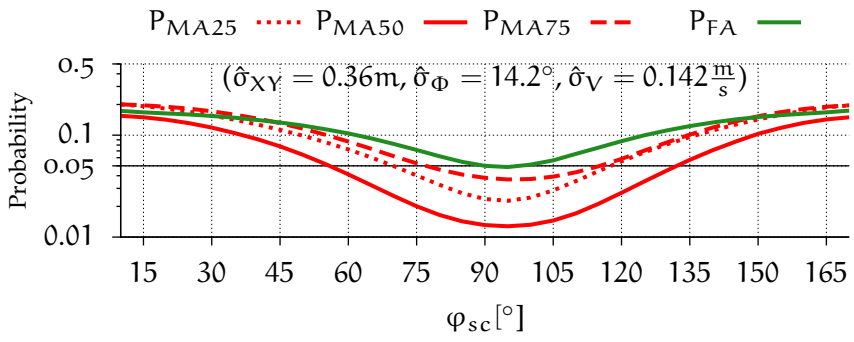
(b) Adjusted $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$ for $P^{15°\leqslant\varphi_{sc}\leqslant165°}_{\{MA*,FA\}} \leqslant 0.1$

Fig. 6.8: Influence of street crossing angles of the pedestrian (φ_{sc}) on $P_{MA*} \wedge P_{FA} \leqslant 0.1$ log-scale ($\log_{10}$)

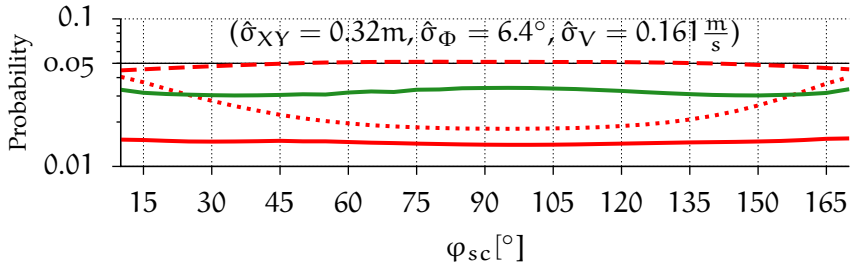
Table 6.5: Accuracy requirements for street crossing angles $15° \leqslant \varphi_{sc} \leqslant 165°$ (maximum standard deviations of the error in position $\hat{\sigma}_{XY}$, direction $\hat{\sigma}_{\Phi}$, and speed $\hat{\sigma}_{V}$) for .

$P^{15°\leqslant\varphi_{sc}\leqslant165°}_{\{MA*,FA\}}$	$\hat{\sigma}_{XY}$	$\hat{\sigma}_{\Phi}$	$\hat{\sigma}_{V}$
$\leqslant 0.10$	0.44	8.6	0.206
$\leqslant 0.05$	0.32	6.4	0.142
$\leqslant 0.01$	0.20	4.2	0.094

ded by the two other impact positions of 25% and 75% according to the Euro NCAP CPNA-25 and CPNA-75 scenarios, see Section 6.2.1. The Missed Alarm Probability for the 25%, 50%, and 75% collision scenario are referred as P_{MA25}, P_{MA50}, and P_{MA75}, respectively. For convenience the following notation will be used: $P_{\{MA*,FA\}} \leqslant \tau$ for all values of P_{MA25}, P_{MA50}, and P_{MA75} and P_{FA} equal or below the threshold τ for a street crossing angle of $\varphi_{sc} = 90°$. In addition $P^{15°\leqslant\varphi_{sc}\leqslant165°}_{\{MA*,FA\}} \leqslant \tau$ indicates the same as $P_{\{MA*,FA\}} \leqslant \tau$ but for all street crossing angles between $15° \leqslant \varphi_{sc} \leqslant 165°$. The scenario setup, as described in Section 6.4, was altered so that the pedestrian crosses the street with an angle of φ_{sc}. Since the TTC changes for different impact positions and for $\varphi_{c} \neq 90°$, each scenario was adjusted by setting the initial position of the pedestrian closer to the car so that TTC $= 2.88\,s$ in every scenario. Fig. 6.8 (a) shows the

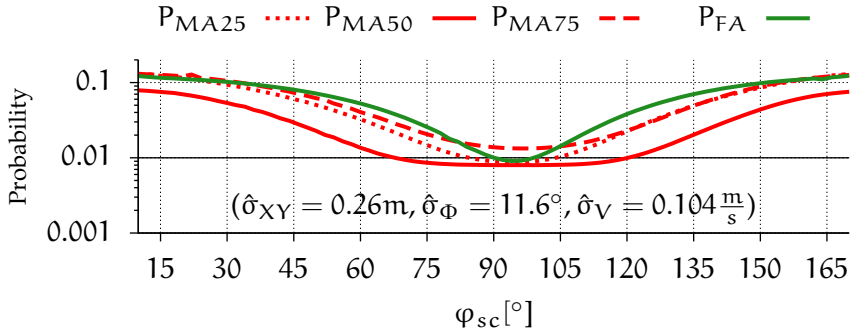


(a) $P_{MA\{25,50,75\}} \wedge P_{FA}$ for different φ_{sc} using $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$ for $P_{\{MA*,FA\}} \leqslant 0.05$

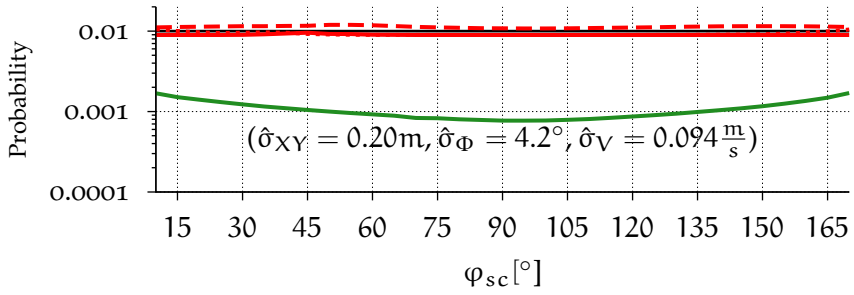


(b) Adjusted $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$ for $P^{15° \leqslant \varphi_{sc} \leqslant 165°}_{\{MA*,FA\}} \leqslant 0.05$

Fig. 6.9: Influence of street crossing angles of the pedestrian (φ_{sc}) on $P_{MA\{25,50,75\}} \wedge P_{FA} \leqslant 0.05$ log-scale ($\log_{10}$)



(a) $P_{MA\{25,50,75\}} \wedge P_{FA}$ for different φ_{sc} using $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$ for $P_{\{MA*,FA\}} \leqslant 0.01$



(b) Adjusted $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$ for $P^{15° \leqslant \varphi_{sc} \leqslant 165°}_{\{MA*,FA\}} \leqslant 0.01$

Fig. 6.10: Influence of street crossing angles of the pedestrian (φ_{sc}) on $P_{MA,\{25,50,75\}} \wedge P_{FA} \leqslant 0.01$ log-scale ($\log_{10}$)

resulting values of $P_{MA,25}$, $P_{MA,50}$, and $P_{MA,75}$, as well as P_{FA}, for street crossing angles between 15° and 165° using the initially determined accuracy thresholds from Table 6.4 for $P_{\{MA,FA\}} = 0.1$ in the basic CPNC-50 scenario. For $P_{\{MA,FA\}} = 0.5$ and $P_{\{MA,FA\}} = 0.01$, these values can be seen in Fig. 6.9 (a) and Fig. 6.10 (a). From these plots, it can be seen that $P_{MA,75}$ (CPNA-75 scenario) has the highest requirements on movement recognition accuracy within the scope of all three collision scenarios. Thus, the maximum error standard deviations $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$, i.e. the maximum error volume, were recalculated for the CPNA-75 scenario for all for street crossing angles between $15° \leqslant \varphi_{sc} \leqslant 165°$. It showed up that maximum error standard deviations $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$ for the CPNA-75 scenario also results P_{FA} under the desired thresholds. Thus, the maximum error standard deviations $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$ for the CPNA-75 scenario for all for street crossing angles between $15° \leqslant \varphi_{sc} \leqslant 165°$ can be used for all the non-collision and all three collision scenarios. The concrete values of $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$ for $P^{15°\leqslant\varphi_{sc}\leqslant165°}_{\{MA*,FA\}} \leqslant \{0.1, 0.05, 0.01\}$ and the resulting influence of φ_{sc} in the different scenarios can be seen in Figs. 6.8 to 6.10 (b).

6.9 TIME TO COLLISION DEPENDENCY

In the previous sections, the required accuracy for the recognition of the pedestrian's movement for TTC = 2.88 s was analysed. Accordingly, in the collision case, the pedestrian needs to move 4 m until he reaches the collision point. For simplification, the term Time to Collision (TTC) is used for collision and non-collision scenarios even though in the non-collision scenario, it is just the time when the car reaches the collision position from the collision scenario.

In this section, it is now analysed how the distance between the car and the pedestrian, i.e., the TTC, influences the required movement recognition accuracy. According to the results from Section 6.8, the CPNA-75 scenario requires the highest movement recognition accuracy within the scope of the investigated street crossing angles between 15° and 165°, see Figs. 6.8 to 6.10 (b). Thus, to find the required movement recognition accuracy for different TTC values, the starting positions for the CPNA-75 scenario was changed for the pedestrian ($\mathbf{r}_p$) and the car ($\mathbf{r}_c$) to $\mathbf{r}_i(2.88 - \Delta t)$ ($i \in \{p, c\}$) for $\Delta t \in [0, 2.8]$ in 400 ms steps. For each of the resulting configurations the $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$ values were then determined in the same way as in Section 6.7.

The Tables 6.6 to 6.8 show the corresponding values of $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$ for $P^{15°\leqslant\varphi_{sc}\leqslant165°}_{\{MA*,FA\}} \leqslant \{0.1, 0.05, 0.01\}$, respectively, for the different TTC values. The values for TTC = 1.28 s are marked in bold since this is approximately a common time to come to a full stop for a car driving with $50\frac{km}{h}$ under good road conditions. It can be seen that using the error volume maximization $(\hat{\sigma}_{XY}, \hat{\sigma}_{\Phi}, \hat{\sigma}_{V})$, the requirement for

Table 6.6: Accuracy requirements (maximum standard deviation of the error in position $\hat{\sigma}_{XY}$, direction $\hat{\sigma}_{\Phi}$, and speed $\hat{\sigma}_{V}$) to keep $P^{15^\circ \leqslant \varphi_{sc} \leqslant 165^\circ}_{\{MA*,FA\}} \leqslant 0.1$ for different TTC values

TTC [s]	$\hat{\sigma}_{XY}$ [m]	$\hat{\sigma}_{\Phi}$ [°]	$\hat{\sigma}_{V}$ [m/s]
2.88		8.6	0.206
2.48		10.2	0.237
2.08	↑	12.2	0.287
1.68	0.44	15.6	0.350
1.28		***21.8***	***0.450***
0.88	↓	44.0	0.500
0.48		60.0	0.813
0.08		60.0	1.000

Table 6.7: Accuracy requirements (maximum standard deviation of the error in position $\hat{\sigma}_{XY}$, direction $\hat{\sigma}_{\Phi}$, and speed $\hat{\sigma}_{V}$) to keep $P^{15^\circ \leqslant \varphi_{sc} \leqslant 165^\circ}_{\{MA*,FA\}} \leqslant 0.05$ for different TTC values

TTC [s]	$\hat{\sigma}_{XY}$ [m]	$\hat{\sigma}_{\Phi}$ [°]	$\hat{\sigma}_{V}$ [m/s]
2.88		6.4	0.161
2.48		7.8	0.186
2.08	↑	9.4	0.219
1.68	0.32	11.6	0.272
1.28		***17.2***	***0.320***
0.88	↓	35.8	0.370
0.48		> 60.0	0.683
0.08		> 60.0	0.821

Table 6.8: Accuracy requirements (maximum standard deviation of the error in position $\hat{\sigma}_{XY}$, direction $\hat{\sigma}_{\Phi}$, and speed $\hat{\sigma}_{V}$) to keep $P^{15^\circ \leqslant \varphi_{sc} \leqslant 165^\circ}_{\{MA*,FA\}} \leqslant 0.01$ for different TTC values

TTC [s]	$\hat{\sigma}_{XY}$ [m]	$\hat{\sigma}_{\Phi}$ [°]	$\hat{\sigma}_{V}$ [m/s]
2.88		4.2	0.094
2.48		4.8	0.094
2.08	↑	5.2	0.103
1.68	0.20	7.4	0.105
1.28		***11.2***	***0.151***
0.88	↓	25.4	0.156
0.48		> 60.0	0.479
0.08		> 60.0	0.579

direction and speed accuracy relax clearly with decreasing TTC while the required position accuracy remains constant.

6.10 SUMMARY

It his Chapter the minimum required movement recognition accuracy for linear pedestrian movement in terms of the standard deviation of normally distributed random measurement errors for position, direction, and speed measurements was analysed.

To estimate the collision detection performance depending on the measurement error distributions, the probability of a Missed Alarm (i.e., a collision is not detected) and the probability of a False Alarm (i.e., a collision is erroneously detected) from Section 4.3 was used. This performance indicators were evaluated for the "Car-to-Pedestrian Nearside" scenarios from the "Test Protocol for AEB VRU Systems" of the Euro NCAP. In these scenarios, the pedestrian is about to cross a street while a car is approaching from the pedestrian's left-hand side. In the basic case of these scenarios, the pedestrian crosses the street with a crossing angle of 90°, and the impact position of the car's width is 50%. It was found that in this basic scenario, an accuracy of ± 0.52 m for the position, $\pm 16.0°$ for direction and $\pm 0.151 \frac{\text{m}}{\text{s}}$ for the speed recognition is required to keep the probability of a Missed Alarm below 0.1 at a time to collision of 2.88 s. These accuracies assure that in the case of a non-collision scenario the probability of a False Alarm is also below 0.1.

For scenarios in which the pedestrian crosses the street with arbitrary crossing angles between 15° and 165° and for impact positions between 25% and 75%, an accuracy for the position of 0.44 m, 8.6° for direction, and $0.206 \frac{\text{m}}{\text{s}}$ for speed is required for a Missed and False Alarm probability $< 10\%$.

Considering smaller distances between the pedestrian and the car, i.e., a decreased time to collision, it was found that the requirements for direction and speed recognition significantly decrease while the required position accuracy remains constant.

REFERENCES

[6.1] K. David and A. Flach, 'CAR-2-X and Pedestrian Safety: Innovative Collision Avoidance System', *IEEE Vehicular Technology Magazine*, vol. 5, no. 1, pp. 70–76, 2010. DOI: 10.1109/MVT.2009.935536 (cit. on p. 87).

[6.2] S. Engel, C. Kratzsch and K. David, 'Car2Pedestrian–Communication: Protection of Vulnerable Road Users Using Smartphones', in *Proc. Advanced Microsystems for Automotive Applications 2013*, Heidelberg, 2013, pp. 31–41. DOI: 10.1007/978-3-319-00476-1_4 (cit. on pp. 87, 89).

[6.3] C. Adrian Hobbs and Paul J. McDonough, 'Development of the European New Car Assessment Programme', in *Proc. 16th In-*

ternational Technical Conference on the Enhanced Safety of Vehicles, Windsor, Canada, Oct. 1998, pp. 2439–2453 (cit. on p. 90).

[6.4] M. van Ratingen, A. Williams, A. Lie, A. Seeck, P. Castaing, R. Kolke, G. Adriaenssens and A. Miller, 'The European New Car Assessment Programme: A historical review', *Chinese journal of traumatology*, vol. 19, no. 2, pp. 63–69, 2016. DOI: 10.1016/j.cjtee.2015.11.016 (cit. on p. 90).

[6.5] European New Car Assessment Programme (Euro NCAP). 'Test Protocol - AEB VRU systems (Version 1.0.1)'. checked 14.12.2020. (Jun. 2015), [Online]. Available: https://cdn.euroncap.com/media/21509/euro-ncap-aeb-vru-test-protocol-v101.pdf (cit. on p. 90).

[6.6] European New Car Assessment Programme (Euro NCAP). 'Test Protocol - AEB VRU systems (Version 3.0.3)'. checked 14.12.2020. (Jun. 2020), [Online]. Available: https://cdn.euroncap.com/media/58226/euro-ncap-aeb-vru-test-protocol-v303.pdf (cit. on pp. 90, 91).

[6.7] M. Bachmann, M. Morold and K. David, 'On the Required Movement Recognition Accuracy in Cooperative VRU Collision Avoidance Systems', *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1708–1717, Mar. 2021. DOI: 10.1109/TITS.2020.2976593 (cit. on p. 93).

[6.8] M. Bachmann, M. Morold, K. David and P. Henkel, 'The Wireless Seat Belt Requirements, Experiments, and Solutions for Pedestrian Safety', in *14th Workshop on Context and Activity Modeling and Recognition*, Athens, Greece, Mar. 2018, pp. 250–255. DOI: 10.1109/PERCOMW.2018.8480123 (cit. on p. 96).

7

ON THE LIMITS OF CAR-BASED COLLISION AVOIDANCE

In this Section the collision detection performance of a Line-of-Sight (LOS)-dependent car-based VRU Collision Avoidance System (VRU CAS) and a *cooperative* VRU Collision Avoidance System (cooperative VRU CAS) in terms of the Missed and False Alarm probability (P_{MA} and P_{FA}) is compared for the CPNC-50 scenario from the "Test protocol for Autonomous Emergency Braking Autonomous Emergency Braking (AEB) Vulnerable Road User (VRU) systems" of the European New Car Assessment Programme (Euro NCAP). For this, the simulations for calculating the performance indicators P_{MA} and P_{FA} from Chapter 6 are extended by a LOS-dependent car-based VRU CAS. In this way, the performance metric will be comparable for both, the cooperative VRU CAS and a car-cased VRU CAS.

7.1 METHODOLOGY

The performance comparison is conducted between a camera-based VRU CAS and the cooperative VRU CAS, proposed [7.1] in the NLOS CPNC-50 scenario from Euro NCAP, and analysed regarding the required movement recognition accuracy in Chapter 6.

For both systems, it is assumed that the collision detection is performed in the car using linear extrapolation of the car's movement vector and the recognized movement vector of the pedestrian. The camera-based VRU CAS derives the movement vector of the pedestrian from the camera's image data. The cooperative VRU CAS receives the movement vector from the pedestrian's mobile device using a reliable, low-latency communication, like 5G or Dedicated Short Range Communication (DSRC). Like in Chapter 6, linear movement vectors comprising the current position, direction, and speed for both the car

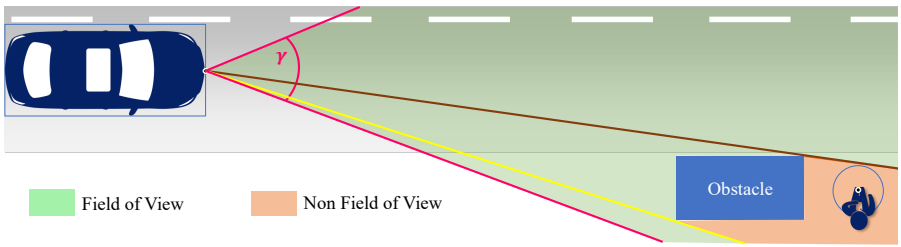


Fig. 7.1: Sketch of a camera-based VRU CAS in a Non-Line-of-Sight (NLOS) CPNC-50 scenario with a Field of View (FOV) angle of γ and a pedestrian visually obstructed by an obstacle.

and the pedestrian are assumed. The comparison is conducted for two pedestrian speeds

1. a walking pedestrian (5 $\frac{km}{h}$)
2. and a running pedestrian (15 $\frac{km}{h}$)

and car driving in

3. a traffic calmed area (30 $\frac{km}{h}$)
4. and a non-calmed urban area (50 $\frac{km}{h}$).

The four scenarios are evaluated in a simulator using a rectangle-to-circle Road Space Model (RSM) as described in Section 3.1. In this RSM the car is modelled as a rectangle with a width $W_c = 2\,m$ and a length of $L_c = 4\,m$. The pedestrian is modelled as a circle with a radius of $r_c = 0.5\,m$, see Fig. 7.1.

The comparison between the camera-based VRU CAS and a the cooperative VRU CAS, consist of evaluating the P_{MA}, as defined in Section 4.2. The P_{MA} will be evaluated for the four collision scenarios within the time domain of the Time to Collision (TTC).

7.1.1 *Latest Time to Avoid Collision*

For every scenario, the Latest Time to Avoid Collision (LTAC), as shown in eq. (7.1), is determined:

$$\text{LTAC} = \frac{v_c}{a} + \frac{0.2\,m}{v_c} \tag{7.1}$$

v_c represents the car speed and a the emergency brake deceleration. For this investigation a deceleration of $a = 11.347\,\frac{m}{s^2}$ is used based on the assumption that the braking distance of a common car driving at 100 $\frac{km}{h}$ is 35 m. The term $\frac{0.2\,m}{v_c}$ effects that the car will stop 20 cm before the collision point.

7.1.2 *Precondition Vision-Depended Collision Avoidance*

Considering a LOS-depended car-based VRU CAS there is an crucial precondition influencing the probability to detect a collision in a given scenario.

This precondition is, if the pedestrian is visible to the LOS-depended car-based VRU CAS before the LTAC. If the pedestrian is not visible to the LOS-depended car-based VRU CAS before LTAC the VRU CAS will not be able to avoid the collision. To be fair, it need to be noted that even if the pedestrian is only visible after LTAC, the system might still reduce the impact speed and thus reduce the risk of fatality. However, in this assessment it if focused on the question if the VRU CAS can prevent the collision.

Table 7.1: Camera parameters

Field of View angle (γ)	90°
Field of View range	70 m
Time to detect pedestrian	200 ms
Time to get movement information	200 ms

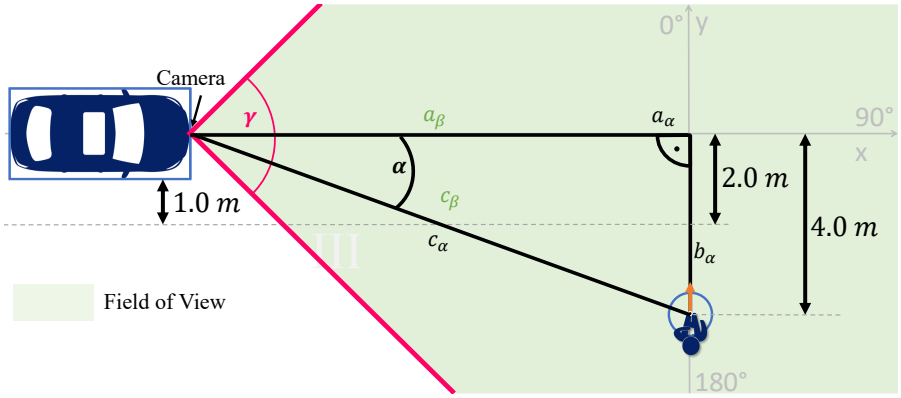


Fig. 7.2: LOS scenario (Euro NCAP CPNA scenario) with a sketch of the Field of View trigonometry of a camera-based system

Thus, let the Time of entry the Field Of View (TFOV) be the time before an impending collision at which the pedestrian is initially in the FOV, i.e. visible to the vision-depended VRU CAS. Furthermore let LTAC be the latest time at which a collision can be prevented. The event that a collision is avoidable ("ca") is then

$$ca = \begin{cases} 1 & , \text{TFOV} \geqslant \text{LTAC} \\ 0 & , \text{TFOV} < \text{LTAC} \end{cases}$$

For the cooperative VRU CAS it is assumed that $ca = 1$ for any TTC $>$ LTAC, as along as the car and pedestrian are in communication range.

7.1.3 *Modelling the Camera-Based VRU Collision Avoidance System*

To model the camera-based system in the rectangle-to-circle RSM, the parameters shown in Table 7.1 were assumed. A sketch of the car-based VRU CAS model with the corresponding trigonometric relations is shown in Fig. 7.2. To detect if the pedestrian is in the FOV of the camera, the angle α between the car's movement direction and the pedestrian's position is determined according to eq. (7.2), see Fig. 7.2.

$$\alpha = \tan^{-1}\left(\frac{b_\alpha}{a_\alpha}\right) \tag{7.2}$$

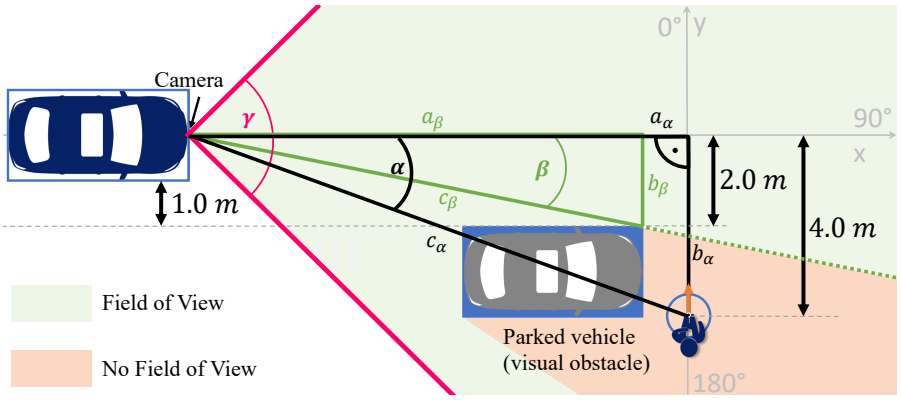


Fig. 7.3: NLOS scenario (Euro NCAP CPNC scenario) with a sketch of the Field of View trigonometry of a camera-based system influenced by an obstacle (parked car)

Thus, given the camera field of view angle $\gamma = 90°$, the pedestrian is in FOV if $\alpha \leqslant \frac{\gamma}{2} = 45°$.

7.1.4 *Modelling Visual Obstruction*

The sketch in Fig. 7.3 shows, how a parked vehicle covers the camera's view on a pedestrian. To determine if the pedestrian is visible to the camera, the angle α from eq. (7.2) detection is evaluated first. If the pedestrian is beneath the centre line of the car's movement, the pedestrian is in the FOV if $0 < \alpha < \beta$, where β is given by eq. (7.3).

$$\beta = \tan^{-1}\left(\frac{b_\beta}{a_\beta}\right) \tag{7.3}$$

For this investigation it is assumed that the pedestrian is in the FOV when the centre point of the pedestrian's circle is inside the FOV of the camera.

7.2 SIMULATION SETUP

The four CPNC-50 based scenarios with $v_c \in \{30\,\frac{km}{h}, 50\,\frac{km}{h}\}$ and $v_p \in \{5\,\frac{km}{h}, 15\,\frac{km}{h}\}$ were configured in the rectangle-to-circle RSM according to Table 7.2.

The four scenarios were simulated within the time domain of $TTC \geqslant ttc \geqslant 0$, where TTC is the constant initial TTC at the starting position $\mathbf{r}_{i_0}$. To find the values of P_{MA} for the camera-based VRU CAS and the cooperative VRU CAS for a walking pedestrian and a running pedestrian for car speeds of $30\,\frac{km}{h}$ and $50\,\frac{km}{h}$, respectively, the initial position for the car was adjusted so that depending on the speed of the car and the pedestrian, the car and the pedestrian reach the collision point at the same time, i.e., the pedestrian would be struck by the centre of the car's front. Note that in the 4th scenario ($v_c = 50\,\frac{km}{h}$ and

Table 7.2: Setup parameters for the four CPNC-50 base NLOS scenarios with different speeds for the car and the pedestrian.

Scenario	v_c $\left[\frac{km}{h}\right]$	v_p $\left[\frac{km}{h}\right]$	$\mathbf{r}_{0_c}$ [m]	$\mathbf{r}_{0_p}$ [m]	TTC [s]	LTAC [s]
1	30	5	(-26.51, 0)	(0, -4)	2.88	0.76
2	30	15	(-10.50, 0)	(0, -4)	0.96	0.76
3	50	5	(-42.50, 0)	(0, -4)	2.88	1.24
4	50	15	(-22.50, 0)	(0, -6)	1.44	1.24

Table 7.3: Error Models for the Cooperative System

Position error longitudinal	$X \sim \mathcal{N}(0, 0.44)$
Position error lateral	$Y \sim \mathcal{N}(0, 0.44)$
Direction error	$\Phi \sim \mathcal{N}(0, 8.6°)$
Speed error	$V \sim \mathcal{N}\left(0, 0.206 \frac{m}{s}\right)$

$v_p = 15 \frac{km}{h}$) the initial distance of the pedestrian to the collision point was set to 6 m instead of 4 m in the other scenarios. This was done to effect that the initial TTC > LTAC. In correspondence to the CPNC-50 scenario a rectangle obstacle with a length of 4 m and a width of 2 m, representing a parked car, was placed with its geometric centre at $(-3\,m, -3\,m)$, see Fig. 7.3.

During the simulation, the time, when the pedestrian enters the FOV of the camera-based VRU CAS was determined according to eq. (7.3). To calculate P_{MA} for the cooperative VRU CAS, the accuracies for $P^{15° \leqslant \varphi_{sc} \leqslant 315°}_{\{MA*,FA\}} \leqslant 0.1$ from Section 6.8 were used, see Table 7.3. The accuracies for the camera-based VRU CAS were taken from [7.2] and are shown in Table 7.4.

For the camera-based VRU CAS system, the P_{MA} was calculated as follows:

1. If the pedestrian is not in the FOV of the camera or the pedestrian is less than 200 ms in the FOV, P_{MA} is assumed to be 1.

2. If the pedestrian is in the FOV for more than 200 ms but less than 400 ms, P_{MA} is calculated assuming a speed of 0 for the pedestrian to take the lack of movement information into account.

3. If the pedestrian is in the FOV for more than 400 ms, the P_{MA} is calculated with full movement information.

For the cooperative VRU CAS, P_{MA} is always calculated with full movement information, since there is no FOV restriction for the cooperative system.

Table 7.4: Error Models for the Camera-based System

Position error longitudinal	$X \sim \mathcal{N}(0, 1\,\mathrm{m})$
Position error lateral	$Y \sim \mathcal{N}(0, 0.5\,\mathrm{m})$
Direction error	$\Phi \sim \mathcal{N}(0, 10°)$
Speed error	$V \sim \mathcal{N}\left(0, 0.5\,\frac{\mathrm{m}}{\mathrm{s}}\right)$

7.3 RESULTS

The P_{MA} and TFOV time (pedestrian is initially in FOV) values for a walking pedestrian, with a speed of 5 $\frac{\mathrm{km}}{\mathrm{h}}$, and a running pedestrian with a speed of 15 $\frac{\mathrm{km}}{\mathrm{h}}$. For both pedestrian speeds, the scenario was evaluated for car speeds of $v_c = 30\,\frac{\mathrm{km}}{\mathrm{h}}$ and $v_c = 50\,\frac{\mathrm{km}}{\mathrm{h}}$, respectively. For each car speed, it was determined if the camera-based VRU CAS is able to recognize the pedestrian before the LTAC, i.e if TFOV > LTAC. Additionally, for every car and pedestrian speed combination, the minimum required FOV angle (γ_{min}) were calculated. This γ_{min} indicates the required FOV angle of LOS-dependent car-based VRU CAS so that the pedestrian is immediately in the FOV when he leaves the visual obstruction of the obstacle.

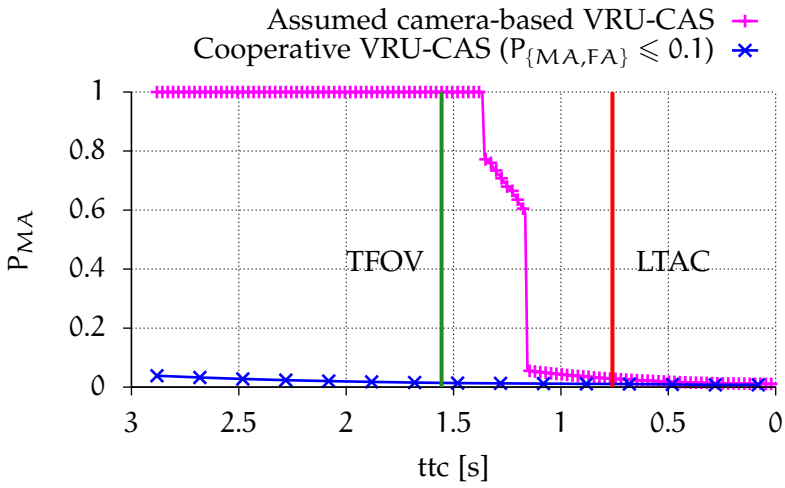
The plots in the following subsections (Figs. 7.4 and 7.5) show the P_{MA} values for a cooperative VRU CAS, the camera-based VRU CAS, and the times for TFOV and LTAC. The blue graphs with crosses indicates the values for the cooperative VRU CAS configured with the movement recognition accuracy for $P_{\{MA*,FA\}}^{15°\leqslant\varphi_{sc}\leqslant165°} \leqslant 0.1$, i.e. $\hat{\sigma}_{XY} = 0.44\,\mathrm{m}$, $\hat{\sigma}_{\Phi} = 8.6°$, and $\hat{\sigma}_{V} = 0.206\frac{\mathrm{m}}{\mathrm{s}}$, compare Section 6.10. The values for the assumed camera-based VRU CAS are indicated by the magenta graphs with plus signs.The vertical red lines mark the LTAC. The TFOV values are indicated as green vertical lines if TFOV > LTAC, or as orange vertical lines if TFOV < LTAC and thus the collision can not be avoided.

7.3.1 *Car 30 km/h*

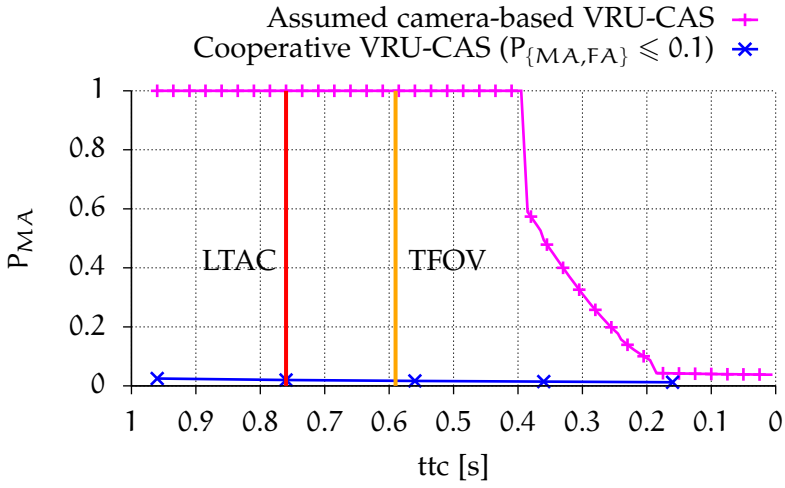
With a speed of 30 $\frac{\mathrm{km}}{\mathrm{h}}$, the car needs 0.74 s to come to a full stop. Accordingly, with the 20 cm distance buffer, the LTAC is 0.76 s in this scenario.

7.3.1.1 *Walking Pedestrian (5 km/h)*

The P_{MA} results in the ttc domain for a walking pedestrian are shown in Fig. 7.4b. To ensure that the pedestrian is in the FOV, an angle $\gamma_{min} > 18.23°$ is required. The pedestrian enters the FOV of the camera system at TFOV = 1.55 s before the collision. Accordingly, the camera-based VRU CAS detects the pedestrian at ttc = 1.35 s and has fully recognized the pedestrian's movement at ttc = 1.15 s. At this



(a) Walking Pedestrian ($5\frac{km}{h}$)



(b) Running Pedestrian ($15\frac{km}{h}$)

Fig. 7.4: Collision detection performance in the NLOS scenario for a car speed of $30\frac{km}{h}$

time P_{MA50} is 0.03. Since LTAC is 0.76 s in this scenario, the system has 0.42 s left to react after the pedestrian is fully recognized.

Thus, a collision is preventable in this scenario using the assumed camera-based VRU CAS.

7.3.1.2 *Running Pedestrian (15km/h)*

The scenario for a running pedestrian is shown in Fig. 7.4b. The running pedestrian leaves the visual obstruction at TFOV = 0.59 s. To be in the FOV of the camera at this time, the camera needs a FOV angle of $\gamma_{min} > 48.70°$.

Since LTAC = 0.73 s, and thus LTAC > TFOV, a collision is not preventable with any LOS-dependent car-based VRU CAS.

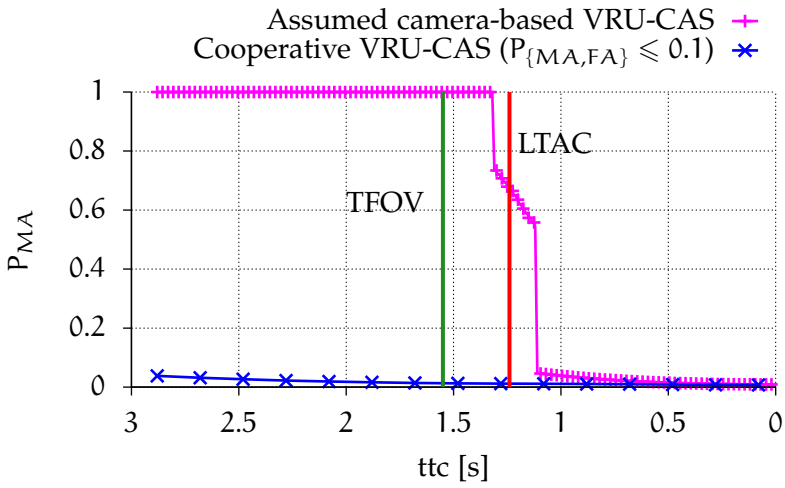
The running pedestrian is detected by the assumed camera-based system at $ttc = 0.39$ s and the movement information is available at $ttc = 0.19$ s. Even when all recognition delays for the camera-based VRU CAS are neglected and the emergency breaking is performed immediately at TFOV = 0.59 s, the impact speed at the pedestrian will still be 5.90 $\frac{km}{h}$. Assuming the more realistic case that emergency braking is performed when the pedestrian was detected at $ttc = 0.39$ s, the impact speed of the car is still 23.79 $\frac{km}{h}$.

7.3.2 *Car 50 km/h*

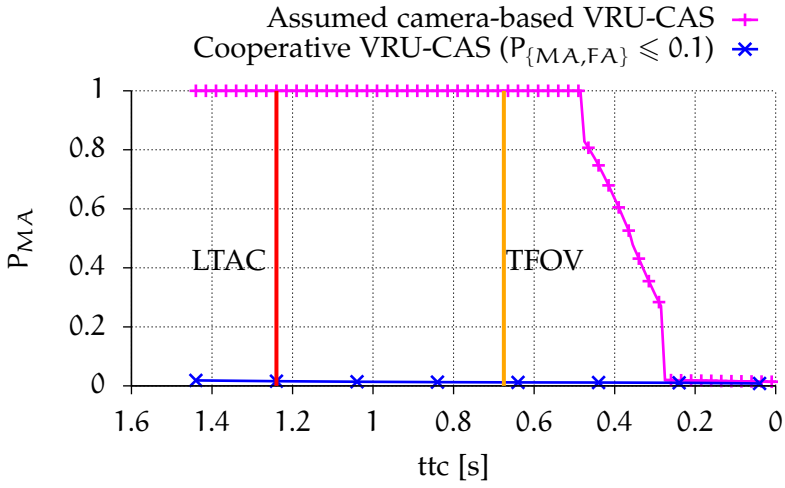
The same scenarios for the walking and running pedestrian were considered, but with an increased car speed of 50 $\frac{km}{h}$. With a car speed of 50 $\frac{km}{h}$, the car needs 1.22 s to come to a full stop for the assumed emergency deceleration of 11.347 $\frac{m}{s^2}$. Accordingly, LTAC = 1.24 s in this scenario.

7.3.2.1 *Walking Pedestrian (5 km/h)*

In Fig. 7.5a, the values of P_{MA}, depending on ttc for the walking pedestrian are shown. For a camera-based VRU CAS with a minimum FOV angle of $\gamma_{min} > 12.40°$, the walking pedestrian is in the FOV at TFOV = 1.55 s. Thus, the walking pedestrian is detected at $ttc = 1.35$ s. At this time, P_{MA} for the camera-based VRU CAS is 0.74, due to missing movement information of the walking pedestrian. Between $1.35\,s > ttc > 1.15\,s$, P_{MA} decreases to 0.56. At $ttc = 1.15$ s the pedestrian is fully recognized, causing a further decrease of P_{MA} to 0.05. This is 90 ms after the LTAC. Therefore, the collision cannot be avoided with the assumed parameters. But, depending on the reaction of the system to initiate an emergency brake, the impact speed into the pedestrian might be significantly reduced. In detail, if there is no further delay, the car's speed is 5.07 $\frac{km}{h}$ at the collision point. Since TFOV > LTAC the collision is in theory preventable. The time available between TFOV and LTAC is $1.55\,s - 1.24\,s = 310\,ms$. Therefore, it



(a) Walking Pedestrian ($5\frac{km}{h}$)



(b) Running Pedestrian ($15\frac{km}{h}$)

Fig. 7.5: Collision detection performance in the NLOS scenario for a car speed of $50\frac{km}{h}$

depends on the recognition parameters of LOS-dependent car-based VRU CAS , whether the collision is preventable in this scenario.

7.3.2.2 *Running Pedestrian (15km/h)*

With a car speed of 50 $\frac{km}{h}$, the LTAC $= 1.24\,s$ is greater than the initial TTC $= 0.96\,s$ at the start of the CPNC-50 scenario. For this reason the distance between the starting position of the running pedestrian and the collision point was changed from $4\,m$ to $6\,m$ for this scenario. This adjustment changes the initial TTC to $1.44\,s$. Provided a $\gamma_{min} > 34.51°$, the running pedestrian is in FOV at TFOV $= 0.68\,s$. For the assumed camera-based VRU CAS the running pedestrian is detected at $ttc = 0.48\,s$ and full movement information would be available at $ttc = 0.28\,s$. Accordingly, like for a car speed of 30 $\frac{km}{h}$, LTAC $>$ TFOV and thus a collision cannot be prevented with any LOS-dependent car-based VRU CAS . Because of the higher car speed of 50 $\frac{km}{h}$, the impact speed on the pedestrian is still 30.39 $\frac{km}{h}$ if an emergency brake is performed after detection, and 38.56 $\frac{km}{h}$ if the emergency brake is performed after full movement recognition. When all delays for a LOS-dependent car-base VRU CAS are neglected and emergency breaking is performed immediately at TFOV the impact speed would still be 22.22 $\frac{km}{h}$.

7.4 SUMMARY

In Table 7.5, the results from all four scenarios are summarized. In the first third of the table the scenario parameters, including the minimum required FOV angle and the LTAC values are summarized. The second third of the table shows the time of the camera-based VRU CAS events, consisting of the time at which the pedestrian enters the FOV (TFOV), the time at which the pedestrian is detected, and the time at which the movement of the pedestrian is fully recognized. TFOV, detected, and "movement recognized" events are marked in bold when they are greater than the LTAC. This means the collision can be avoided by the assumed camera-based VRU CAS.

From the results, it is evident that a car-based VRU CAS is not able to detect a running pedestrian, moving with a speed $\geqslant 15$ $\frac{km}{h}$ in the considered CPNC-50 scenario of the Euro NCAP. The inability to detect the running pedestrian is caused by the physical and geometric constraints in the scenario and not due to any parameter of the camera-based VRU CAS.

Pedestrian activity	Walking		Running	
Pedestrian speed (v_p) $[\frac{km}{h}]$	5		15	
Car speed (v_c) $[\frac{km}{h}]$	30	50	30	50
Initial TTC [s]	2.88		0.96	1.44
Min. FOV angle (γ_{min}) [°]	18.23	48.70	12.40	34.51
LTAC	0.76	1.24	0.76	1.24
	Camera system times (ttc [s])			
TFOV	**1.55**	**1.50**	0.59	0.67
detected	**1.35**	**1.30**	0.39	0.48
movement recognized	**1.15**	1.10	0.19	0.28
	P_{MA50} at LTAC			
Camera car-based VRU CAS[1]:	3%	56%	100%	100%
Cooperative VRU CAS[2]:	2%	2%	2%	2%

[1] $\sigma_X = 1\,m$, $\sigma_Y = 0.5\,m$, $\sigma_\Phi = 10.0°$, $\sigma_V = 0.5\,\frac{m}{s}$
[2] $\sigma_{XY} = 0.44\,m$, $\sigma_\Phi = 8.6°$, $\sigma_V = 0.206\,\frac{m}{s}$

Table 7.5: Performance comparison of a (camera) car-based VRU CAS and a cooperative VRU CAS

REFERENCES

[7.1] K. David and A. Flach, 'CAR-2-X and Pedestrian Safety: Innovative Collision Avoidance System', *IEEE Vehicular Technology Magazine*, vol. 5, no. 1, pp. 70–76, 2010. DOI: 10.1109/MVT.2009.935536 (cit. on p. 109).

[7.2] M. Bachmann, M. Morold, S. Engel, J. Götz and K. David, 'Camera vs. Cooperative VRU Collision Avoidance', in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, Antwerp, Belgium, May 2020, pp. 1–5. DOI: 10.1109/VTC2020-Spring48590.2020.9128854 (cit. on p. 113).

8

SUMMARY AND CONCLUSION

This dissertation discussed the concept of a *cooperative* VRU Collision Avoidance System (cooperative VRU CAS) in which Vulnerable Road Users (VRUs) are equipped with some mobile device that is able to exchange movement information with the vehicle. It was argued that a cooperative VRU CAS could address most limitations of the existing car-based VRU Collision Avoidance Systems (VRU CASes): Since in a cooperative VRU CASes the VRUs and vehicles transmit their movement vectors via radio, they are independent of the Line-of-Sight (LOS) condition; a key limitation of car-based VRU CASes.

Another major advantage of a cooperative VRU CAS in contrast to car-based VRU CASes is that not only the driver, but also the VRU could be warned. By considering the State-of-the-Art it was identified that most researchers propose a cooperative VRU CAS on the basis of smartphones. Essentially, mobile devices, like smartphones or smartwatches, provide everything that is necessary for a cooperative VRU CAS: First, to determine the movement of a VRU, Global Navigation Satellite Systems (GNSS), digital compass information, and Inertial Measurement Units (IMUs), i.e. an accelerometer and a gyroscope, can be used. Second, to exchange movement information with vehicles, direct and cellular radio access methods are available. The technical possibilities and the fact that a rising number of people permanently carry smartphones with them, makes such devices a promising platform for cooperative VRU CASes.

8.1 SUMMARY

The research question of this dissertation was "What is the minimum required accuracy for VRU movement recognition on mobiles devices for a certain collision detection performance of a cooperative VRU CAS?". To answer this question two challenges needed to be solved:

CHALLENGE I (C I): How to precisely detect collisions in a cooperative VRU CAS?

CHALLENGE II (C II): How to quantify the collisions detection performance of a VRU CAS independent to a certain architecture?

Those two challenges were addressed by:

C I: A precise Time to Collision (TTC) based collision detection algorithm. The algorithm models real world scenarios in a two-dimensional coordinate system, called Road Space Model (RSM),

and supports point, rectangle, and circle geometries to represent the road users. Within the RSM the algorithm is able to determine both, the time to collision and the impact position at the road users' geometries (Chapter 3).

C II: To quantify the collision detection performance of a VRU CAS the two Key Performance Indicators (KPIs), namely the "probability of a Missed Alarm" (P_{MA}) and the "probability of a False Alarm" (P_{FA}) were introduced (Chapter 4). To precisely determine the KPIs for a cooperative VRU CAS (Section 4.5), an adaption of the precise TTC based collision detection algorithm was used to determine Collision Direction Ranges (Chapter 5).

Based on the precise TTC based collision detection algorithm and the P_{MA} and P_{FA} KPIs, an comprehensive analysis on the required movement recognition accuracy for CPNC-50 scenario from the "Test Protocol for Autonomous Emergency Braking (AEB) VRU Systems" of the European New Car Assessment Programme (Euro NCAP) was conducted: The minimum required accuracies to get P_{MA} and P_{FA} below 10%, 5%, and 1% was analysed in terms of the maximum standard deviation of normally distributed errors for position, direction, and speed for different street crossing angles based on the CPNC-50 scenario (Chapter 6).

Afterwards, a comparison of the collision detection performance of a cooperative VRU CAS and car-based VRU CAS was performed for the CPNC-50 Non-Line-of-Sight (NLOS) scenario from the "Test Protocol for AEB VRU Systems" of the Euro NCAP using the results for the minimum required movement recognition accuracy. The comparison was conducted for two VRU speeds, first a walking pedestrian and second a running pedestrian. It was analysed when the VRU is in the Field of View (FOV) of the car-based VRU CAS, and based on this time, if a collision can still be prevented according to the Latest Time to Avoid Collision (LTAC) (Chapter 7).

8.2 CONCLUSION

Based on the comparison of the collision detection performance of a cooperative VRU CAS and car-based VRU CAS in the CPNC-50 scenario from the mandatory "Test Protocol for AEB VRU Systems" of the European New Car Assessment Programme (Euro NCAP), it was found that only scenarios with a walking pedestrian can be addressed by a car-based VRU CAS. It was shown that a collision with a running pedestrian can not be avoided by any LOS-dependet car-based VRU CAS due to trigonometric restrictions. In detail:

- A walking pedestrian with 5 $\frac{km}{h}$, is in general detectable by a car-based VRU CAS before the LTAC. But, it depends on the

detection and reaction time of the car-based VRU CAS if the collision can actually be avoided.

- Regarding a running pedestrian, with 15 $\frac{km}{h}$, it was shown that the pedestrian is not detectable by a car-based VRU CAS before LTAC. This is independent of both, the field of view angle and the detection time of the camera-based system
- Since a cooperative VRU CAS is not influenced by a NLOS scenario and not restricted to a filed of view angle, the cooperative VRU CAS is able to recognize the presence of a VRU in any case.

Regarding reliable collision detection, i.e. keeping the probability of a missed alarm (P_{MA}) and a false alarm (P_{FA}) under a certain thresholds, a cooperative VRU CAS would required the following minimum accuracies for recognizing the movement of a VRU in terms of maximum standard deviations of the position, speed and direction error:

To keep the probability of a Missed Alarm and False Alarm below 0.1 ($P_{\{MA,FA\}} \leqslant 0.1$) at a time to collision of 2.88 s and for a street crossing angle of $\phi_{sc} = 90°$, a maximum standard deviation below:

- $\pm 0.52\,m$ ($\pm 0.26\,m$ for $P_{\{MA,FA\}} \leqslant 0.01$) for the position error,
- $\pm 16.0°$ ($\pm 11.6°$ for $P_{\{MA,FA\}} \leqslant 0.01$) for direction error, and
- $\pm 0.151\,\frac{m}{s}$ ($\pm 0.104\,\frac{m}{s}$ for $P_{\{MA,FA\}} \leqslant 0.01$) for the speed error is required.

For scenarios in which the pedestrian crosses the street with arbitrary crossing angles between 15° and 165° and for impact positions between 25% and 75%, a maximum standard deviation below:

- $\pm 0.44\,m$ ($\pm 0.20\,m$ for $P_{\{MA,FA\}} \leqslant 0.01$) for the position error,
- $\pm 8.6°$ ($\pm 4.2°$ for $P_{\{MA,FA\}} \leqslant 0.01$) for direction error, and
- $\pm 0.206\,\frac{m}{s}$ ($\pm 0.094\,\frac{m}{s}$ for $P_{\{MA,FA\}} \leqslant 0.01$) for the speed error is required.

Considering smaller distances between the pedestrian and the car, i.e., a decreased time to collision, it was found that the requirements for direction and speed recognition significantly decrease while the required position accuracy remains constant.

While, especially for the standard CPNC-50 scenario from the Euro NCAP, speed and direction errors, are already in the domain of the required accuracy, the position error of smartphones is yet not accurate enough to provide a collision detection performance of $P_{\{MA,FA\}} \leqslant 0.1$. It can be expected that within the next years the position accuracy of smartphone-grade GNSS systems, which are used for getting absolute positioning information, will be further increased. Modern GNSS

receivers in smartphones already provide the possibility to use "dual-frequency" positioning which significantly reduces errors even in urban environments. Thus, in future, the concept of cooperative VRU CAS could become an important application to increase the safety of VRUs, especially in urban environments where car-based VRU CASes have considerable limitations.

APPENDIX

i Road user $i \in \{c, p\}$.

MOVEMENT

x_i: X-coordinate of road user i.

y_i: Y-coordinate of road user i.

v_i: Speed in $\frac{m}{s}$ of road user i.

ϕ_i: Direction of road user i in rad.

t: Time in seconds

$\mathbf{r}_{0_i} = (x_i, y_i)^T$: Current position of road user i

$\mathbf{r}_i(t) = v_i \cdot t \cdot \begin{pmatrix} \sin\phi_i \\ \cos\phi_i \end{pmatrix} + \mathbf{r}_{0_i}$: Linear movement equation for the road user i.

$\mathbf{m}_i = (\mathbf{r}_{0_i}, \phi_i, v_i)$: Linear movement vector of road user i.

ttc: Time to collision in seconds between two road users.

TTC: Initial ttc values in the beginning of a scenario.

t_x: Time at which both road users are the same x-coordinate.

t_y: Time at which both road users are the same y-coordinate.

MOVEMENT RECOGNITION ERRORS

σ_{xy}: Standard deviation for position error symmetrical in the lateral and longitudinal direction of the VRU

σ_x: Standard deviation for position error in the lateral direction of the VRU

σ_y: Standard deviation for position error in the longitudinal direction of the VRU

σ_Φ: Standard deviation for direction error

σ_v: Standard deviation for speed error

$\mathbf{X}$: Bi-variate random variable for position error

Φ: Random variable for direction error

V: Random variable for speed error

COLLISION DIRECTION RANGES

$[\varphi_{k_{min}}, \varphi_{k_{max}}]$, $k \in \{1, 2\}$: Collision direction ranges

φ_k, $k \in \{1, 2\}$: The two possible collision directions as WGS84 azimuth values for point-to-point models

$\bar{\varphi}_j$, $j \in \mathbb{N}$: Pseudo collision direction thresholds

GEOMETRIES

$G_i = \{P, R, C\}$: Geometry of road user i, where as $P = point$, $R = rectangle$, $C = circle$

g_i: Geometry component of geometry i, where as $P = point$, $R = rectangle$, $C = circle$

G_c-TO-G_p: Road user models used in RSM, e.g. rectangle-to-point when road user c is modelled as a rectangle and road user p is modelled as a circle.

L_i: Corner $L \in \{A, B, C, D\}$, of a rectangle geometry from road user i.

r_i: Radius of the circle geometry of the road user i.

w_i: Width of the road user i's rectangle geometry.

l_i: Length of the road user i's rectangle geometry.

ROAD SPACE MODEL TRANSFORMATIONS

$\dot{x}_i, \dot{y}_i$: Coordinates in a normalized or relative RSM

$\ddot{x}_i, \ddot{y}_i$: Coordinates in a rotated relative RSM

$\dot{\phi}_p$: Direction of road user p in a normalized RSM

$\dot{\phi}_c = 90°$: Direction of road user c in a normalized RSM, always 90°

$\tilde{\phi}_p = \tilde{\phi}$: Relative direction of road user p in a (rotated) relative RSM

$\tilde{v}_p = \tilde{v}$: Relative speed of road user p in a (rotated) relative RSM

$[\dot{\varphi}_{k_{min}}, \dot{\varphi}_{k_{max}}]$, $k \in \{1, 2\}$: Collision direction range in a normalized RSM

$[\tilde{\varphi}_{min}, \tilde{\varphi}_{max}]$: Collision direction range in a relative RSM

$\dot{\bar{\varphi}}_j$, $j \in \mathbb{N}$: Pseudo collision direction thresholds in a normalized RSM

$\ddot{\bar{\varphi}}$, $j \in \mathbb{N}$: Pseudo collision direction thresholds in a rotated relative RSM

$\dot{\varphi}_k$, $k \in \{1, 2\}$: The two possible collision directions in a normalized RSM for point-to-point models

$\ddot{\varphi}$: The collision direction in a rotated relative RSM for point-to-point models

$\dot{\theta}$: WGS84 rotation to normalized RSM

$\ddot{\theta}$: Normalized RSM to rotated normal RSM

$\check{\theta}$: Re-rotation for g_i

WGS84 MAPPING

d: Metric distance between two WGS84 coordinates.

φ_{WGS84}: Azimuth in the WGS84 coordinate system.

x_{i_0}: Original x-coordinate of road user i in a non transformed RSM.

y_{i_0}: Original y-coordinate of road user i in a non transformed RSM.

v_{i_0}: Original speed in $\frac{m}{s}$ of road user i in a non transformed RSM.

ϕ_{i_0}: Original direction of road user i in degree in a non transformed RSM.

COLLISION DIRECTION (RANGES)

φ_{p_j}, $j \in \mathbb{N}$: Direction thresholds for p between which the collisions occurs in a scenario.

φ_{rel_k}, $k \in \mathbb{N}$: Direction thresholds for p between which the collisions occurs in a rotated relative scenario.

$[\varphi_{min}, \varphi_{max}]$: A collision range consisting of minimum and maximum φ_{p_j} or φ_{rel_k}, $j, k \in \mathbb{N}$.

ACRONYMS

AEB	Autonomous Emergency Braking
CAM	Cooperative Awareness Message
CAS	Collision Avoidance System
CDR	Collision Direction Range
CPA	Closest Point of Approach
CPNA	Car-to-Pedestrian Nearside Adult
CPNC	Car-to-Pedestrian Nearside Child

cooperative VRU CAS	*cooperative* VRU Collision Avoidance System
D2D	Device-to-Device
DSRC	Dedicated Short Range Communication
Euro NCAP	European New Car Assessment Programme
FN	False Negative
FOV	Field of View
FP	False Positive
GDA	Geographical Destination Area
GIDAS	German In-Depth Accident Study
GNSS	Global Navigation Satellite Systems
HSDPA	High Speed Downlink Packet Access
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IPG	Inter Packet Gap
KPI	Key Performance Indicator
LIDAR	Light Detection and Ranging
LOS	Line-of-Sight
LTAC	Latest Time to Avoid Collision
LTE	Long Term Evolution
MEC	Mobile Edge Computing
NLOS	Non-Line-of-Sight
P2P	Peer-to-Peer
PDR	Packet Delivery Ratio
PIR	Packet Inter-Reception Time
Radar	Radio detection and ranging
RFID	Radio Frequency Identification
RMSE	Root-Mean-Square Error
RSM	Road Space Model
RSSI	Received Signal Strength Indicator
RSU	Road Side Units
SIM	Subscriber Identity Module
SubRec	Subdivision and Reconstruction
T-ITS	Transactions on Intelligent Transportation Systems

TAS	Time Available to Stop
TFOV	Time of entry the Field Of View
TN	True Negative
TP	True Positive
TTC	Time to Collision
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-everything
VANet	Vehicular Ad Hoc Network
VRU	Vulnerable Road User
VUT	Vehicle under Test
VRU CAS	VRU Collision Avoidance System
WGS84	World Geodetic System 1984
WLAN	Wireless Local Area Network

LIST OF FIGURES