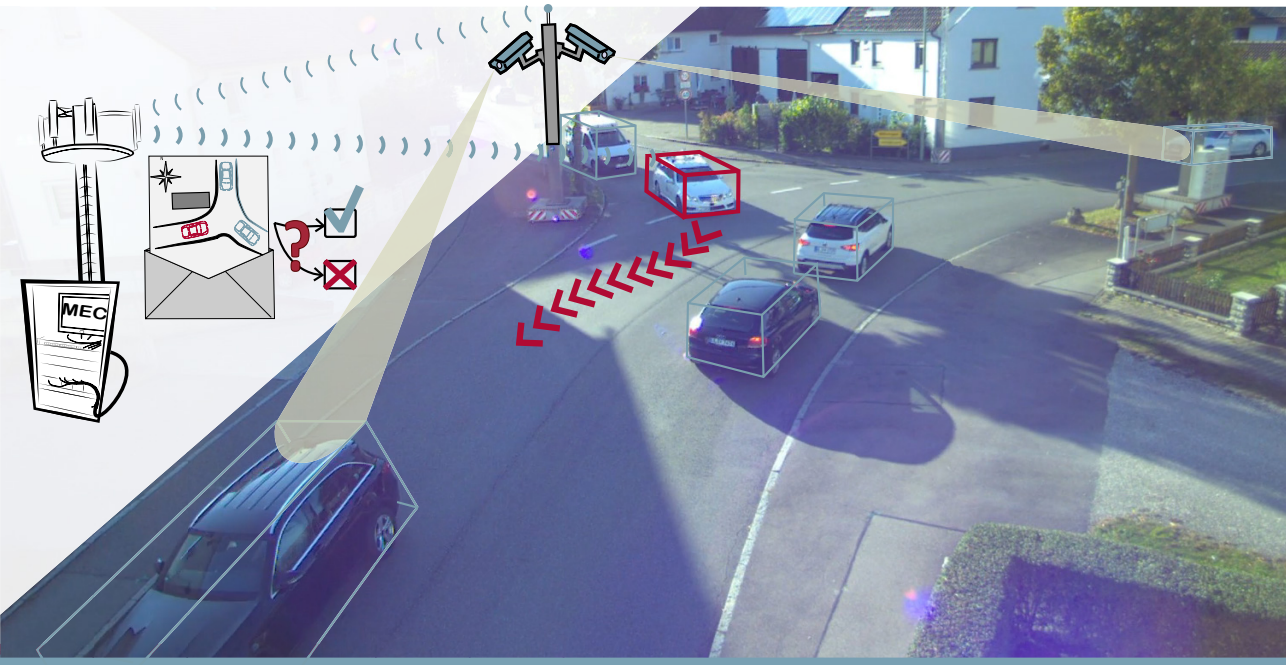
universität ulm



Motion Planning for Connected Automated Vehicles at Occluded Junctions with External Perception

Johannes Christian Müller

Band 41

Schriftenreihe des Instituts für Mess-, Regel- und Mikrotechnik

Johannes Christian Müller

Motion Planning for Connected Automated Vehicles at Occluded Junctions with External Perception

Schriftenreihe des
Instituts für Mess-, Regel- und Mikrotechnik
Universität Ulm

Herausgeber:
Prof. Dr.-Ing. Klaus Dietmayer

Band 41

Johannes Christian Müller

Motion Planning for Connected Automated Vehicles at Occluded Junctions with External Perception

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de abrufbar.

Dissertation, Universität Ulm,
Fakultät für Ingenieurwissenschaften, Informatik und Psychologie, 2021

Impressum

Universität Ulm
Institut für Mess-, Regel- und Mikrotechnik
Prof. Dr.-Ing. Klaus Dietmayer
Albert-Einstein-Allee 41
89081 Ulm
http://www.uni-ulm.de/mrm

Eine Übersicht über alle Bände der Schriftenreihe finden Sie unter
http://www.uni-ulm.de/mrmschriften.

Diese Veröffentlichung ist im Internet auf dem Open-Access-Repositorium der Universität Ulm (https://oparu.uni-ulm.de) verfügbar und dort unter der Lizenz “Standard” publiziert. Details zur Lizenz sind unter http://www.uni-ulm.de/index.php?id=50392 zu finden.

Institut für Mess-, Regel- und Mikrotechnik der Universität Ulm 2022
Print on Demand

ISBN 978-3-941543-63-8 ISBN 978-3-941543-64-5 (E-Book)



ulm university universität
uulm

Motion Planning for Connected Automated Vehicles at Occluded Junctions with External Perception

DISSERTATION

zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEURS
(Dr.-Ing.)

der Fakultät für Ingenieurwissenschaften, Informatik
und Psychologie der Universität Ulm

von

Johannes Christian Müller
aus Friedrichshafen

Gutachter:	Prof. Dr.-Ing. Klaus Dietmayer
	Prof. Dr.-Ing. J. Marius Zöllner
Amtierender Dekan:	Prof. Dr.-Ing. Maurits Ortmanns

Ulm, 16.12.2021

To my parents Rita and Martin and to my brother Manuel

Danksagung

Zuallererst möchte ich mich bei meinem Doktorvater Prof. Dr.-Ing. Klaus Dietmayer für die Betreuung meiner Dissertation bedanken. Wann immer ich etwas gebraucht habe, habe ich es selbst in herausfordernden Zeiten zügig bekommen. Außerdem konnte ich in unseren Gesprächen wichtige Impulse und wertvolles Feedback für meine Forschung mitnehmen. Weiterhin bedanke ich mich bei Prof. Dr.-Ing. J. Marius Zöllner für die Übernahme des Zweitgutachtens und bei Prof. Dr.-Ing. Dr. h.c. Stefan Wesner und Prof. Dr.-Ing. Thomas Walter, die sich als weitere Prüfer meiner Arbeit angenommen haben. Daneben möchte ich Dr.-Ing. Michael Buchholz erwähnen, der die interne Koordination des Projekts MEC-View übernommen und meine Arbeit mit betreut hat. Außerdem bedanke ich mich bei Prof. Dr. rer. nat. Vasileios Belagiannis und Prof. Dr.-Ing. Amr Rizk, die für fachliche und persönliche Fragen immer ein offenes Ohr hatten und mit denen ich mich freundschaftlich verbunden fühle.

Des weiteren möchte ich mich bei meinen Kollegen für die schöne gemeinsame Zeit am Institut bedanken. Mein besonderer Dank gilt Jan Strohbeck, Thomas Griebel, Dr. Roman Geiselhart, Dr.-Ing. Felix Mesmer, Max Mertens, Sebastian Menner, Suvrath Pai, Alexander Tsaregorodtsev, Peter Speth, Marc Forstenhäusler und von Nachbarinstituten André Dürr, Dr.-Ing. Tobias Meuser und Florian Ostermaier. Die vielen fachlichen Diskussionen haben mich weitergebracht und meinen Horizont erweitert. Die zahlreichen schönen gemeinsamen Momente – auch abseits der Arbeit – werden mir in besonderer Erinnerung bleiben. So sind für mich aus Kollegen Freunde geworden und die schönen Momente haben mich immer wieder auch durch anstrengende Zeiten getragen.

Ein großer Dank geht auch an „meine“ Studenten. Die Zusammenarbeit hat mir immer große Freude bereitet. Insbesondere bedanke ich mich bei Robin Dehler, der mich bei vielen Engineering-Arbeiten tatkräftig unterstützt hat.

Außerdem bedanke ich mich bei unseren Technikern Franz Degenhard, Oliver Betz, Martin Nieß, Thomas Löffler und Uwe Kerner für die tatkräftige Unterstützung. Ein weiterer großer Dank geht an unsere Sekretärinnen Claudia Fricke und Diemut Magon, die mich nicht nur bei allerlei Verwaltungstätigkeiten unterstützt haben, sondern auch als gute Seelen des Instituts immer für mich und meine Kollegen da waren.

Daneben möchte ich mich bei meinen MEC-View-Projektpartnern für die gute Zusammenar-

beit bedanken. Besonders bedanke ich mich bei Dr. Benjamin Völz, Michael Gabb, Tobias Müller, Holger Digel, Matthias Maier, Steffen Schulz und unserem Gesamtprojektleiter Dr. Rüdiger Walter Henn. Gemeinsam haben wir geschafft, was unter Pandemie-Bedingungen unmöglich erschien, sodass wir MEC-View trotz allem sehr erfolgreich abschließen konnten.

Zum Schluss möchte ich mich bei meiner Familie und meinen Freunden für die immerwährende Unterstützung bedanken. Besonders bedanke ich mich bei meinen Eltern Rita und Martin, meinem Bruder Manuel, meiner Tante Brigitte Nölle und meinem Onkel Prof. Dr.-Ing. Nikolaus Müller.

Euch allen gilt mein herzlicher Dank!

Abstract

Automated driving represents a key technology for the mobility of the future. However, although automated driving has been studied for decades, complex urban traffic scenarios, where the ego perception is restricted by occlusions, still pose tremendous challenges for automated vehicles. External perception, for example based on infrastructure sensors, can help to resolve such occlusions so that the automated vehicle can efficiently handle even complex traffic situations. However, if the external perception data is unreliable, this can lead to dangerous situations.

Therefore, to ensure the safety of the maneuvers, the reliability of the external perception must first be assessed, which is particularly challenging due to the small amount of data on which the reliability must be estimated. For this purpose, a Subjective-Logic-based method is presented in this thesis that uses two consistency tests and two plausibility tests to automatically evaluate the reliability of the external perception. In addition, another evaluation scheme, also based on the mathematical theory Subjective Logic, assesses the reliability of the vehicle-to-anything (V2X) communication channel.

Based on this, a novel motion planning concept is presented, which, on the one hand, minimizes the residual risk of the planned maneuvers and, on the other hand, maximizes the driving comfort of the passengers. A formalized description of the situation context allows the motion planning to be adapted to a wide variety of traffic situations. The residual risk is estimated by a risk model, which takes into account the interaction with other road users, uncertainties in the perception, and the estimated reliability of the external perception. By additionally limiting the residual risk to a maximum permissible level, the safety of the planned trajectories can always be guaranteed in a probabilistic sense. In case an invariably safe set around the planned trajectory can be constructed based on set-based methods, the safety of the planned trajectory can even be formally guaranteed.

Both, motion planning and reliability estimation are evaluated experimentally using a challenging application scenario at an occluded unsignalized yield junction. For the experiments, infrastructure sensors provide the external perception. The results confirm that both methods work as expected.

Kurzfassung

Automatisiertes Fahren stellt eine Schlüsseltechnologie für die Mobilität der Zukunft dar. Doch obwohl automatisiertes Fahren bereits seit Jahrzehnten erforscht wird, stellen insbesondere komplexe urbane Verkehrsszenarien, in welchen die Fahrzeug-eigene Wahrnehmung durch Verdeckungen eingeschränkt ist, automatisierte Fahrzeuge immer noch vor große Herausforderungen. Externe Perzeption, beispielsweise auf Basis von Infrastruktursensorik, kann helfen, solche Verdeckungen aufzulösen, sodass das automatisierte Fahrzeug komplexe Verkehrssituationen effizient bewältigen kann. Sind die Daten der externen Perzeption jedoch unzuverlässig, kann dies zu gefährlichen Situationen führen.

Um die Sicherheit der Manöver zu gewährleisten, muss daher zunächst die Zuverlässigkeit der externen Perzeption bewertet werden. Dies stellt auf Grund des kleinen Umfangs an Daten, auf Basis dessen die Zuverlässigkeit geschätzt werden muss, eine besondere Herausforderung dar. Dazu wird ein Subjective-Logic-basiertes Verfahren vorgestellt, welches mithilfe von zwei Konsistenz- und zwei Plausibilitätstests die Zuverlässigkeit der externen Perzeption automatisiert bewertet. Darüber hinaus beurteilt ein weiteres Bewertungsschema ebenfalls auf Basis der Subjective Logic die Zuverlässigkeit des Kommunikationskanals.

Darauf aufbauend wird ein neuartiges Bewegungsplanungskonzept vorgestellt, welches einerseits mithilfe eines Risikomodells die Interaktion mit anderen Verkehrsteilnehmern, die Unsicherheiten in der Perzeption und die geschätzte Zuverlässigkeit der externen Perzeption berücksichtigt und das Restrisiko der geplanten Manöver minimiert und andererseits den Fahrkomfort der Insassen maximiert. Indem das Restrisiko zusätzlich auf ein maximal zulässiges Maß begrenzt wird, kann die Sicherheit der geplanten Trajektorien im probabilistischen Sinn immer garantiert werden. Für den Fall, dass auf Basis von mengenbasierten Verfahren eine invariante sichere Zustandsmenge um die geplante Trajektorie konstruiert werden kann, kann die Sicherheit der geplanten Trajektorie sogar formal garantiert werden. Durch eine formalisierte Beschreibung des Situationskontext lässt sich die Bewegungsplanung auf verschiedenste Verkehrssituationen anpassen.

Sowohl die Bewegungsplanung als auch die Zuverlässigkeitsschätzung werden am Beispiel eines anspruchsvollen Anwendungsszenarios an einer verdeckten Vorfahrt-Achten-Kreuzung experimentell validiert. Infrastruktursensoren stellen hierbei die externe Perzeption zur Verfügung. Die Ergebnisse bestätigen, dass beide Verfahren wie erwartet funktionieren.

Contents

1 Introduction **1**
- 1.1 State of the Art for Reliability Estimation . . . 3
- 1.2 State of the Art for Motion Planning . . . 7
- 1.3 Contributions . . . 11

2 Background **13**
- 2.1 Application Scenario . . . 13
- 2.2 Infrastructure-based Traffic Surveillance System . . . 15
- 2.3 Connected Automated Vehicle . . . 18

3 Fundamentals **21**
- 3.1 Subjective Logic . . . 21
  - 3.1.1 Opinion Representation . . . 21
  - 3.1.2 Fusion Operators . . . 23
- 3.2 Optimal Control and Model Predictive Control for Motion Planning . . . 25
- 3.3 Set-based Methods for Motion Planning . . . 29

4 An Urn Model Intuition to Subjective Logic **33**
- 4.1 Motivation and Intuitive Derivation . . . 33
- 4.2 Formalization and Mathematical Proof . . . 40
- 4.3 Summary . . . 51

5 Subjective-Logic-based Reliability Estimation **53**
- 5.1 Reliability Estimation of a Communication Channel . . . 54
  - 5.1.1 Problem Formulation . . . 54
  - 5.1.2 Identification Algorithm . . . 55
  - 5.1.3 Simulative Evaluation . . . 57
  - 5.1.4 Real-World Experiment . . . 58
- 5.2 On-board Reliability Estimation of External Information . . . 62
  - 5.2.1 Prediction Test . . . 63
  - 5.2.2 Map Test . . . 64
  - 5.2.3 Ego Localization Test . . . 66
  - 5.2.4 Ego Perception Test . . . 67
  - 5.2.5 Overall Fusion . . . 69

5.2.6 Experimental Setup . . . 70
5.2.7 Experimental Results . . . 71
5.3 Reliability and Trust in a Vehicular Multi-Agent System . . . 76
5.3.1 System Description and Trade-Off Considerations . . . 77
5.3.2 Attacker Model . . . 79
5.3.3 Trust Building . . . 80
5.3.4 Misbehavior Detection and Trust Revision . . . 80
5.3.5 Simulations of the Generalized Reliability Estimation Scheme for the Application Scenario . . . 83
5.3.6 Simulation Study on Scaling the Approach by the Example of Traffic Jam Detection . . . 85
5.4 Summary . . . 88

6 Motion Planning for CAVs at Occluded Intersections 89
6.1 Context Awareness . . . 91
6.1.1 General Concept . . . 91
6.1.2 Processing of the Situation Context . . . 93
6.1.3 Situation Context for the Application Scenario . . . 94
6.2 Motion Planning . . . 95
6.2.1 Integration of an External Environment Model . . . 96
6.2.2 General Risk Model . . . 98
6.2.3 Risk Model for the Application Scenario . . . 99
6.2.4 Combinde Decision Making and Longitudinal Trajectory Planning . 100
6.2.5 Lateral Trajectory Tracking . . . 104
6.2.6 Planning Algorithm . . . 105
6.3 Simulation Studies for the Application Scenario . . . 107
6.3.1 Comparison Between Jerk Optimal and Time-Weighted Jerk Optimal Solution . . . 107
6.3.2 Monte-Carlo Simulations on Decision Making . . . 108
6.4 Experimental Validation . . . 113
6.4.1 Lane Keeping Properties of the Motion Planning Scheme . . . 114
6.4.2 Maneuver Timing . . . 115
6.5 Summary . . . 118

7 Conclusion and Future Work 119

A Augmented Lagrangian Approach and Numerical Optimization Strategies 121
A.1 Augmented Lagrangian Approach . . . 121
A.2 Numerical Optimization Strategies . . . 123

B Additional Parameters 127

C Supervised Bachelor and Master Theses 131

D Author's Publications **133**

Acronyms **137**

List of Symbols **141**

Bibliography **153**

Chapter 1

Introduction

Automated driving has a huge potential to make the mobility of the future more efficient. With 6.9 billion hours additional traveling time and 11.7 billion liter of fuel wasted in traffic jams in the USA in 2014 alone [AY20], reducing these numbers even by a fraction would contribute to fight the climate change and to improve the quality of life for thousands of commuters. However, despite decades of research, urban automated driving has not fully been achieved so far [YLCT20]. Many of the key challenges on the way to urban automated driving condense in traffic scenarios at unsignalized junctions. Accident statistics [SHP09] indicate that junction scenarios are challenging even among human drivers: they involve strong interaction with other road users, require adequate decision making and maneuver timing over long prediction horizons, and contain significant uncertainties with respect to position, velocity, and future motion of the other road users. Automated vehicles have to handle these challenges in real time, while providing safety, robustness and passenger comfort at the same time [EWS21; HZ19; YLCT20].

In addition, many urban traffic scenarios comprise occluded areas due to other road users or static obstacles such as buildings. This further complicates the respective traffic scenarios. Hence, maintaining safe motion despite occlusions itself is a challenging task [KA20]. In general, three options to handle such occlusions are common. First, the sensor setup of an automated vehicles can be upgraded such that occlusions appear in less traffic scenarios. However, this is associated with high hardware costs and still does not work for all scenarios. Second, sacrificing traffic efficiency, an automated vehicle could slow down such that it can immediately stop if an object appears out of the occluded areas. Third, an automated vehicle could receive external perception information from vehicle-to-anything (V2X) communication to resolve the occlusions and to efficiently navigate through the scenario. Following such a communication approach renders the automated vehicle a connected automated vehicle (CAV). However, when relying on an external perception, it is crucial for the safety of the CAV that the external perception actually is correct [HDLK18; SES18]. Deciding whether a source of external perception is reliable or not gets particularly challenging, since the CAV typically has only few measurements and limited computational power available. Thus, for

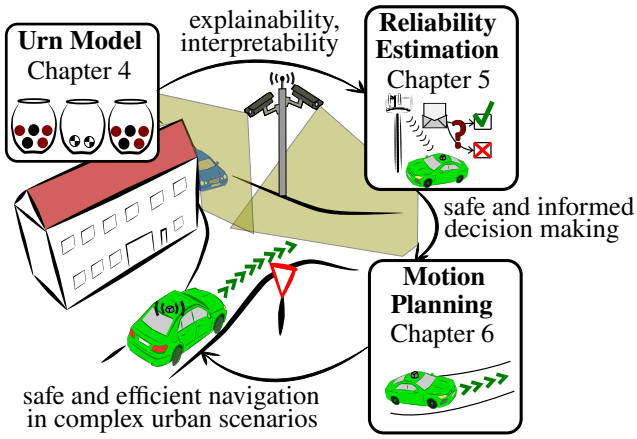


Figure 1.1: Goals of the thesis and chapters addressing the related topics.

safe and informed decision making, two major questions arise, which will be addressed by this thesis. First, how can the reliability of an external perception be estimated in an explainable and interpretable way before relying on it for motion planning? Second, how can a CAV safely and efficiently merge into an occluded junction using external perception? The goals associated with answering the questions and the chapters that address the corresponding topics are shown in Fig. 1.1.

To address the first major question of this thesis, two Subjective Logic (SL)-based reliability estimation schemes are proposed, one to estimate the reliability of the communication channel between the CAV and the external perception, and one to assess the reliability of the external perception itself. Both reliability estimation schemes are designed to operate in real time aboard a CAV. Since SL is designed to operate on insufficient statistics, the estimation scheme can operate on the small sample sizes to which the online decision making is limited. For the channel reliability estimation, the communication channel is modeled as a Markov chain, which then is identified using a novel SL-based identification scheme. The reliability of the external perception is estimated based on two SL consistency tests and two SL plausibility tests, which then are weighted according to the available statistical evidence to yield the overall result. Both reliability estimation schemes are evaluated through experiments. Furthermore, it is shown that the reliability estimates from such estimation schemes can be used to identify and remove misbehaving agents from the V-MAS, while a reputation can be built for well-behaving agents. Thus, trust can be established within a V-MAS. It is demonstrated by simulations that the approach is applicable even for a large scale scenario [MMSB19].

To adress the second major question, an optimization-based motion planning scheme is

presented that uses an external environment model transferred to the CAV via V2X communication to resolve occlusions and thus enhance the safety and performance of the CAV. By sampling over the available behavior options formalized within a representation of the situation context, the proposed approach solves decision making and longitudinal trajectory planning holistically. Thus, each behavior decision results in a drivable trajectory, which is not always given for hierarchical schemes [MLH+19]. In contrast to existing sampling-based planning schemes like [WZKT10], the proposed method considers multiple behavior options at the same time and adds a time-weight to the jerk that can reduce jerky behavior due to replanning effects. To keep the CAV on its lane, an Model Predictive Control (MPC) based lateral trajectory tracking is employed, which makes the CAV follow the longitudinal trajectory along the intended path. For the MPC, an indirect optimization strategy [EVM+19] is employed. As opposed to partially observable Markov decision process (POMDP)-based approaches like [HSX+18] optimizing the interaction between other road users and the CAV, the proposed motion planning considers other road users as a constraint. This significantly reduces the calculation efforts, since it avoids iteratively adapting the CAV's trajectory and the predicted trajectories of the other road users. A risk model then calculates the risk of violating this or another constraint accounting for the uncertainties in the environment model. Formal safety guarantees are inherited from the concept of invariably safe set (ISS) [PKWA17] for the special case of vanishing residual risk, while limiting the residual risk according to the risk model otherwise leads to softened guarantees in a probabilistic sense. For the motion planning, an external environment model can originate from various external perception sources like other connected road users, road side units, or a multi access edge computing (MEC) server in a mobile network. However, instead of using classical T2T fusion to merge the environment models, the environment models are used in parallel. This avoids self-fulfilling prophecies that might occur if the CAV's environment model is passed to further agents or back to the environment model service on the server. The proposed motion planning scheme is evaluated by both simulations and experiments in real traffic. For the experiments, a challenging application scenario at an occluded urban intersection in Ulm-Lehr is selected, since this scenario covers many challenges that occur in urban traffic. However, although the evaluation of the motion planning scheme focuses on the application scenario, the proposed concept is more general and can be applied to many other scenarios. Thus, it is argued, how the proposed approach and the experimental results would transfer to other scenarios.

1.1 State of the Art for Reliability Estimation

In literature, reliability estimation has so far been addressed mostly from four perspectives. First, reliability estimation is related to safety of the intended functionality (SOTIF), a validation method gathering statistical evidence, e.g. by Monte-Carlo simulations and scenario-based testing, to estimate the reliability of a module under test [RPL+20]. However,

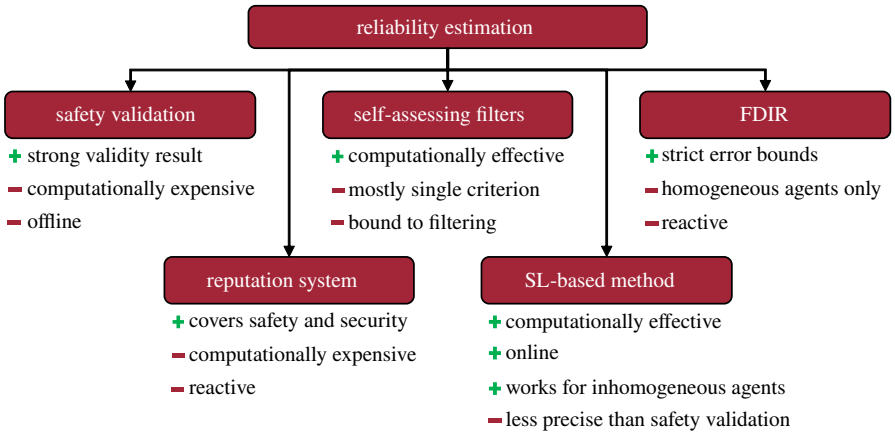


Figure 1.2: Categorization of methods related to reliability estimation and a summary of strengths and weaknesses.

existing SOTIF methods are mostly performed offline and require large amounts of statistical evidence, which is not available online. Second, fault detection, isolation, and recovery (FDIR) methods have been proposed for multi-agent systems (MAS) [DKJ18; ZKA+17] , where reliability is established implicitly by removing non-reliable agents from the MAS. Third, from the field of filtering and information fusion, self-assessment methods have been proposed, mostly based on the normalized innovation squared (NIS) [BFC90; Gib13; Pic16]. However, these methods usually are inherently coupled with the filtering and mostly have the NIS as the only criterion. Finally, for the domain of security, e-commerce, and Internet of Things (IoT), trust management and reputation systems have been proposed [JIB07], from which, recently, misbehavior detection systems for vehicular multi-agent systems (V-MASs) have evolved [HDLK18]. For these systems, SL has already been applied. SL is also used to realize the reliability estimation scheme of Chapter 5. Yet, the existing trust management and misbehavior detection systems can only determine the reliability of an agent a posteriori, whereas the CAV needs the estimate before it relies on the external perception. An overview on these categories and a summary of strengths and weaknesses is given in Fig. 1.2.

The increasing level of automation transfers responsibility from the driver to the automated vehicle. Thus, safety assessment becomes increasingly important [OZZ20]. Particularly, to evolve automated vehicles into autonomous systems, as pointed out in [MMT+21], online self-assessment is vital. This already follows from the literal meaning of the Greek αὐτόνομος (autonomous), which means *having its own laws, being independent*, where independence requires the ability to assure correct functionality and to detect and react to problems on its own. In the field of avionics and navigation SOTIF has been widely adopted for a long time [Pul11]. Conversely, in the context of automated driving, SOTIF became a key stone of safety assessment only recently [RPL+20]. In contrast to functional safety, which

ensures that a technical system is free of failures, SOTIF goes beyond, ensuring that even performing an intended function insufficiently does not induce hazards to an otherwise failure-free system [RPL+20]. In the context of connected automated driving, reliability estimation of an external perception addresses SOTIF, since on the one hand, accurate external information can be used to resolve occlusions and thus to boost the efficiency of motion planning, while, on the other hand, wrong external information can induce hazards to a CAV [MGB19]. Therefore, methods from safety assessment and validation are related to the reliability estimation of Chapter 5, which also addresses SOTIF.

In the survey [RPL+20] of Riedmaier et al., current methods for safety assessment and validation are summarized. Safety assessment strategies can be roughly categorizes as formal verification, functional-based, scenario-based, and real-world testing. The latter, however, is hardly feasible due to the huge amount of test kilometers necessary [WW16]. Proving absolute safety of automated vehicles through formal verification methods, e.g. [KPMA19; PKA19], however, is not possible either since road traffic imposes an open parameter space upon automated driving [RPL+20]. However, while these methods are insufficient on their own, they can be combined with other methods to reduce the testing efforts or to yield results for confined subsystems. Similarly, function-based testing can only supplement other methods, since the required functionality of automated vehicles in any conceivable situation cannot be specified. Thus, only scenario-based testing remains to reach SOTIF in automated driving [RPL+20]. Finally, scenario-based approaches reduce the testing to sequences of significant actions and events, the scenarios [UMR+15]. Thus, testing efforts can be reduced significantly so that testing gets feasible [RPL+20]. Furthermore, if real-world testing for selected scenarios of particular interest is combined with simulation studies that systematically explore the parameter space, results with high expressiveness can be obtained at tractable efforts. For the simulation part of th scenario-based safety assessment, Jesenski et al. proposed quite promising simulation methods [JSSZ19; JTSZ20]. One of the biggest challenges with scenario-based methods is to determine all relevant scenarios while limiting them to a set that remains tractable for testing [RPL+20]. Besides, in contrast to the reliability estimation proposed in this thesis, the safety assessment and validation methods are intended primarily to operate offline and, thus, cannot react to online changes. Furthermore, for offline operation, real-time capability is not relevant, so the few safety assessment methods that can be used online, e.g. [PKA19], tend to be computationally expensive [AGH+20].

In 2018, van der Heijden et al. surveyed the application of trust management and misbehavior detection systems for V-MAS [HDLK18]. Misbehavior in this respect refers to agents that do not comply with their expected behavior either due to system faults or due to an attack. One of the key findings of the survey is that security issues have often been ignored by the community and full reliability of the entities in a V-MAS has often been wrongly assumed. Accordingly, the issue of misbehaving agents has not received sufficient attention so far and an ideal solution for cooperative misbehavior detection does not exist. More recently, Azzedin et al. surveyed the relation between trust and information fusion in the context of IoT [AG19]. In this survey, it is stated that conducting a reliable information fusion is an

open field of research. Like in [HDLK18], it is pointed out that the reliability aspect has not received enough attention so far, while it is emphasized that unreliable data sources have big negative impact on the fusion results. Recently, Liu [Liu20] surveyed trust modelling from a Bayesian perspective and pointed out that trust is an enabling technology for informed decision making. However, he also pointed out that existing trust models, including SL-based trust models, typically come with high computational load. Furthermore, pure trust models themselves do not address how to make informed decisions based on the trust information. Compared to the reliability estimation of this work, trust is mostly inferred from former interactions with respective agents [YSL+13], rather than estimated online while entering the respective interactions. Hence, trust tends to lack behind at least one interaction. More specifically, from the trust perspective, the proposed reliability estimation can be seen as a mean to collect statistical evidence based on which trust can be established. This is shown by the example of the trust management system proposed in Chapter 5.3. The existing approach that comes closest to the proposed misbehavior detection mechanism is that of Dietzel et al. [DHDK14], who used SL for misbehavior detection in a vehicle-to-vehicle communication network. Yet, in contrast to existing methods, see e.g. [AG19], the mechanism of Chapter 5.3 is computationally light weight and scales well to even large scale scenarios.

In the field of filtering and fusion, reliability estimation is often addressed by self-assessment. Classically, the NIS has been applied for this purpose [BFC90]. More recently, further consistency tests, as e.g. proposed by Gibbs [Gib13], have been proposed. Existing online tests for Kalman filter consistency then have been investigated by Piché in 2016 [Pic16]. While most of the tests could be traced back to the NIS, one further test similar to the NIS was found that is able to detect other forms of model errors. Based on consistency tests and Random Finite Sets theory, Stübler et al. developed a self-assessment for localization applications [SRD17]. Other self-assessment methods based on the Dempster-Shafer theory [Sha76], like [NSV+18; NSX+17], or approaches based on neuronal networks, e.g. [HGS+14], have also been proposed. Eventually, Griebel et al. [GMBD20] proposed a Kalman filter approach with a SL based self-assessment. The first difference between these self-assessment methods and the proposed approach is that the self-assessment methods are all inherently coupled with the filtering, whereas the method of this thesis is not. Moreover, the reliability estimation of Chapter 5 can account for different aspects of reliability, whereas most of the discussed self-assessment methods focus on a single criterion.

From the field of MAS and system theory in general, particularly for the application of smart grids, FDIR methods have been investigated [ZKA+17]. These methods often rely on state observers. However, according to the MAS survey of Dorri et al. [DKJ18], FDIR methods tend to suffer from high computational load, often require homogeneous agents, and the fault detection has not received enough attention. More recent approaches [FXWW20; HP20; LXP20; XZLD20] have addressed several FDIR related issues, such as the resource consumption [HP20]. Nonetheless, compared to the reliability estimation proposed in Chapter 5, these methods focus on the system theoretical properties of the MAS, like error bounds, rather than on the statistical properties of the observed statistical evidence.

Methodologically, the reliability estimation is challenging because the CAV has to decide on the basis of only a few measurements. In particular, classical probabilistic methods tend to fail as the statistics corresponding to the available measurements break down. This is, while classical methods can easily model first order uncertainties such as measurement uncertainties in terms of covariances, they do not provide an explicit measure for the statistical uncertainty, i.e. a second order uncertainty, and thus yield results with little expressiveness for small sample sizes. One extension of classical probability theory, which has a representation for the statistical uncertainty and thus can cope with small sample sizes, is the evidence theory by Dampster and Shafer [Sha76]. Therefore, the evidence theory has been employed for reliability estimation [NSV+18; NSX+17]. However, as shown by Zadeh's example [Zad84], the evidence theory may yield unintuitive or even wrong results if the incoming information is highly contradictory. Zadeh's example states: *"Suppose that a patient, P, is examined by two doctors, A and B. A's diagnosis is that P has either meningitis, with probability 0.99, or brain tumor, with probability 0.01. B agrees with A that the probability of brain tumor is 0.01, but believes that it is the probability of concussion rather than meningitis that is 0.99. Applying the Dempster rule to this situation leads to the conclusion that the belief that P has brain tumor is 1.0 - a conclusion that is clearly counterintuitive because both A and B agree that it is highly unlikely that P has a brain tumor"* [Zad84]. In the context of automated driving, Zadeh's example can occur, e.g. if the external perception detects objects with high reliability but with poor accuracy due to a calibration error. In this case, the two criteria detection rate and detection accuracy are contradictory.

More recently, the mathematical theory SL has been proposed. It is an extension of classical probability theory designed to handle small sample sizes [Jøs16; Jøs97]. This theory overcomes the shortcomings of the evidence theory by providing a wide range of fusion operators for different scenarios. However, the community has been quite doubtful about SL [DTHT14], because SL is a complex mathematical theory that is "a bit difficult to follow" [DTHT14]. To provide additional intuition for SL and the processes which take place in the proposed reliability estimation schemes, a novel urn model is presented in Chapter 4. Particularly, the operators from SL used to realize the reliability estimation schemes are represented in the urn model. The urn model is furthermore used to mathematically prove the connection between SL and the Pólya urn scheme [EP23], which models branching phenomena such as contagion.

1.2 State of the Art for Motion Planning

To answer the second major question of this thesis, an adequate motion planning scheme, i.e., a solution for decision making and trajectory planning, is required for the CAV. Existing motion planning schemes can be classified into four broad classes: graph-based methods, planning schemes built on partially observable Markov decision processes (POMDPs), end-

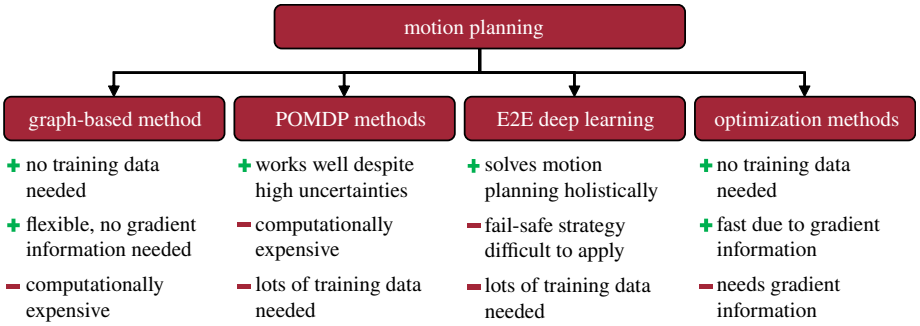


Figure 1.3: Categorization of motion planning methods and a summary of strengths and weaknesses.

to-end (E2E) deep learning techniques, and optimization-based approaches. These method classes as well as their strengths and weaknesses are summarized in Fig. 1.3. The proposed motion planning can be categorized as optimization-based.

Graph-based methods discretize the state and action space and interpret the discretized states as nodes in a graph, where the state transitions induced by the actions represent the edges. Then, graph search techniques, like the A^* search algorithm [HNR68], rapidly exploring random trees (RRTs), or a Monte-Carlo tree search, are employed to find the optimal solution. For example, the A^* search algorithm is used in [FHW16], the work of Du et al. [DML+16] is based on RRTs, and Kurzer et al. [KFZ20; KZZ18] relied on Monte-Carlo tree search. Since the graph-based methods do not rely on gradient information, these methods are very flexible and thus are used e.g. for valet parking [BNKZ17], which requires to navigate through unstructured environment with little gradient information. However, these approaches suffer heavily from the curse of dimensionality and therefore usually contain sophisticated heuristics, e.g. [KFZ20]. Nonetheless, the approaches normally involve high computational cost.

A further development of the graph-based approaches are the methods based on the POMDPs [GHT+16; HSX+18; MLH+19] . POMDPs are particularly well-suited to model uncertainties in the environment of the ego vehicle, the interaction with other road users, and the predicted temporal evolution of the traffic situation, since they allow for latent states, which can only be inferred from measurements, but not measured directly. For example, such latent state can be the intention of another driver. POMDPs can be solved either classically through POMDP solvers, like in [HSX+18], or the solution can be approximated through reinforcement learning [GHT+16]. However, as for the graph-based methods, the POMDP-based methods suffer from the curse of dimensionality. In fact, the complexity of solving POMDPs generally is PSPACE complete [SAR18]. Reinforcement learning can reduce computational effort by learning heuristics for approximate solutions. Nonetheless, this neither solves the principal problem nor is it easily applicable, since it is effortful to train

learning-based methods [YLCT20]. Details on reinforcement learning techniques can be found in the survey of Haydari et al. [HY20].

Similar to most POMDP-based methods, E2E deep learning methods use neural networks to map the sensor input to driving actions [KHJ+19]. However, although these techniques already show promising results, they still need a significant amount of training data and still tend to suffer from poor generalization [YLCT20]. Furthermore, fail-safe measures are more difficult to apply [YLCT20] and safety verification is an unsolved problem [RPL+20].

Classical optimization-based methods are hierarchically structured into decision making and trajectory planning, where the decision making restricts the trajectory planning to the area in the state space, which corresponds to the decision [PCY+16]. Furthermore, the trajectory planning is sometimes divided into longitudinal and lateral planning, which is known as *path velocity decomposition*. Decision making has been solved by finite-state machines, but more recent approaches also used POMDP-based methods [MLH+19], or even set-based methods [GSMA19; KA20; PA18a; PKWA17]. The key idea of set-based methods is to construct reachable sets, i.e. regions in the state space which can possibly be occupied by a road user within the prediction horizon. From these reachable sets, a so-called ISS is derived. An ISS is a region in the state space where the safe navigation of the CAV is formally guaranteed. The formal guarantees are the key benefit of the set-based methods. However, these formal guarantees have to be bought by a significantly restricted solution space for the trajectories. Thus, truly speaking, such motion planning schemes tend to be overly conservative and might not allow to solve at all complex traffic scenarios with high uncertainty. Furthermore, the formal guarantees might pretend a wrong safety owing to failing assumptions. Particularly, the perception system is often assumed to detect all objects in the FOV with 100 % accuracy, which is a questionable assumption. Since the set description of uncertainties is vastly incompatible with classical probabilistic modeling, yet most perception modules work probabilistically, set-based approaches are often difficult to apply. Hence, Ge et al. [GSMA19] themselves admitted that a probabilistic extension of set-based methods would be an important next step. Besides, to approximate the set bounds, set-based methods often require a sampling over the state space, which renders these methods computationally expensive [AGH+20]. In contrast to existing methods, the motion planning scheme of Chapter 6 uses a risk model that combines probabilistic and set-based methods. Hence, for simple situations with a negligible risk of violating the assumptions, the planning scheme can inherit the formal guarantees from the set-based methods, while it still can handle complex situations with high uncertainty, yet at the cost of softening the formal guarantees to guarantees in a probabilistic sense. While risk-minimized planning has also been addressed, e.g., by [KMB+18; WF18], none of these papers showed real-world experiments or had comparable formal guarantees.

For the trajectory planning, an Optimal Control Problem (OCP) is stated and solved using an optimization strategy. Existing optimization strategies can be categorized in three classes: direct approaches, indirect approaches, and sampling-based methods. Direct approaches,

e.g. [ZBDS14], use a discretization to reduce the OCP to a parameter optimization problem and then apply sequential quadratic programming to it to obtain a solution. Indirect methods, like [KG14], which was employed in [GGD18], numerically solve the first order optimality conditions to obtain the solution of the OCP. Finally, sampling-based approaches, like [WZKT10], first solve a simplified optimization problem analytically and then conduct a grid search over the remaining parameters to find the optimal solution for the full OCP. Indirect methods, particularly sub-optimal approaches like [KG14], typically are faster than direct methods and can cope well with nonlinear problems [GGD18]. However, the indirect methods require a good initial solution to profit from their hot start capabilities. Still, with increasing optimization horizon, the computational complexity rises for both, direct and indirect methods. In contrast, by employing an analytical solution, sampling-based methods in principle become independent of the initial solution and the optimization horizon. Their calculation time rather depends on the size of the parameter space they need to search.

A further aspect of the motion planning is the communication type of the CAV. For this, three types of approaches are common in literature. Methods of the first type consider the road users as agents in a MAS, where either a central authority schedules the motion of the individual agents or the agents negotiate their maneuver timings. Since the perception data and the intended trajectories are shared among the MAS, occlusions are resolved and uncertainties can be neglected. This is a major advantage of these methods. However, in contrast to current mixed traffic, these approaches usually assume all road users to be interconnected. Because of this unrealistic assumption, communication-based approaches have not yet proven successful in practice [YLCT20]. Details on these approaches can be found, e.g., in the surveys [RM17] and [PAPP17]. Approaches of the second type, e.g. [CD19], share raw sensor data among a communication network, which then is processed by the individual agents and used for motion planning. Thus, occlusions can be resolved despite some road users being not connected. However, these approaches impose huge amounts of network traffic, where scaling the system to many CAVs would lead to exhaustive bandwidth requirements. Approaches from the third type, like [AAB+19; GDMH19; GMT+16], use collective perception and share preprocessed object lists to save bandwidth. However, in contrast to this thesis, these methods rely on track-to-track (T2T) fusion to merge the object lists [RLNH19]. While T2T fusion may increase the accuracy of the perceived objects, it also requires effortful correction mechanisms [RLNH19]. Moreover, if T2T fusion is used to merge external and ego object lists, the origin of the information gets lost. Thus, if the information is fed back, this may lead to self-fulfilling prophecies, e.g., in the form of persistent clutter objects. This is avoided by using the environment models in parallel rather than merging them, which is proposed in Chapter 6.2.1.

Chapter 2 gives an overview on the background of the thesis. This is, the example scenario and the experimental setup with which the methods of this thesis are evaluated, are presented. Moreover, Chapter 3 briefly describes the fundamentals of this thesis and Chapter 7 gives a conclusion and suggestions for future work.

1.3 Contributions

The Chapters 4-6 describe the contributions of this thesis, which can be summarized as follows:

- An urn model providing an alternative intuition to SL is proposed in Chapter 4. The connection between SL and Pólya's urn model, and further key properties of the urn model are proven mathematically.

- A novel SL-based method to estimate the reliability of the communication link between the CAV and the external percepetion is presented in Chapter 5. This includes a novel SL-based identification method for Markov chains. Simulations and experiments show that the new method has less estimation uncertainty and reacts as quickly as the classical one. Furthermore, a SL-based estimation scheme is proposed to estimate the reliability of an external perception. Experiments confirm a reliability of the proposed reliability estimation scheme of over 90 % for a confidence level of 95 %. Based on the resulting reliability estimates, a trust management system is presented which identifies misbehaving agents or even collaborative attackers and removes them from a V-MAS. At the same time, reputation is built for the well-behaving agents. Hence, trust can be established within a V-MAS. Simulation studies show that the approach scales well even to large-scale systems like the traffic in the city of Cologne.

- A motion planning scheme is proposed in Chapter 6 that optimizes for safety and passenger comfort over all available behavior options across different behavior decisions. For that, the planning scheme comprises a formalized representation of the situation context that allows the motion planning to reduce the calculation efforts and to adapt to various situations. By sampling over the available behavior options for the current situation context the motion planning scheme solves decision making and longitudinal trajectory planning holistically. To generate the longitudinal trajectory corresponding to the available behavior options, a closed form analytical solution is proposed, which is optimal with respect to time-weighted jerk while a point-mass-model is used as system dynamics. It is shown through simulations that the time-weighting can mitigate jerky behavior due to replanning in uncertain environments. A risk model is proposed, which combines set-based methods and probabilistic approaches to compute the residual risk of a planned maneuver. This risk model also takes into account that the external environment model might not be fully reliable. Nonetheless, the planned trajectories come with general safety guarantees in a probabilistic sense. By employing set-based methods, in case of vanishing residual risk, the formal safety guarantees from [PKWA17] are inherited. The motion planning scheme is evaluated by both large-scale Monte-Carlo simulations and experiments in real traffic showing that the approach makes reasonable decisions and yields appropriate trajectories.

Chapter 2

Background

A key challenge for prototype systems, such as those used in this thesis, is to balance the engineering effort, cost, and funding required with the generality of the results obtained. For this reason, the evaluation of the concepts proposed in this thesis focuses on a specific application scenario, while the concepts are formulated in general terms and thus can be applied to many other scenarios as well. The application scenario is presented in Section 2.1, where, furthermore, an overview on the overall system is provided. Moreover, it is discussed, how the results obtained for the application scenario can be applied to other scenarios. Then, in Section 2.2, an overview on the infrastructure-based traffic surveillance system (ITSS) is presented, which is the data source for the external perception in this application scenario. Finally, Section 2.3 introduces the CAV used within this thesis.

2.1 Application Scenario

To efficiently use the available resources and to comply with the funding policy objectives of the MEC-View project, an unsignalized occluded yield junction in Ulm-Lehr [BMH+21] was selected as an application scenario to showcase the results of the proposed methods. The application scenario is illustrated in Fig. 2.1. Its key idea is to achieve an increase of traffic efficiency when merging into an occluded yield junction. Due to a building close to the junction, the main road is occluded for the CAV when it approaches the junction. Hence, without external information, any vehicle has to stop at the yield line and observe the main road before it can merge into it. In contrast, with external information and the proposed motion planning scheme, a CAV can dynamically merge into the junction, since the occlusions are resolved through the external information. However, to safely perform the merging maneuver, the CAV needs to rely on the external information. Thus, estimating the reliability of the external information is important to the decision making of the CAV. Furthermore, the external information has to be integrated into the ego environment model of the CAV to support the motion planning. Dynamically merging into the yield junction

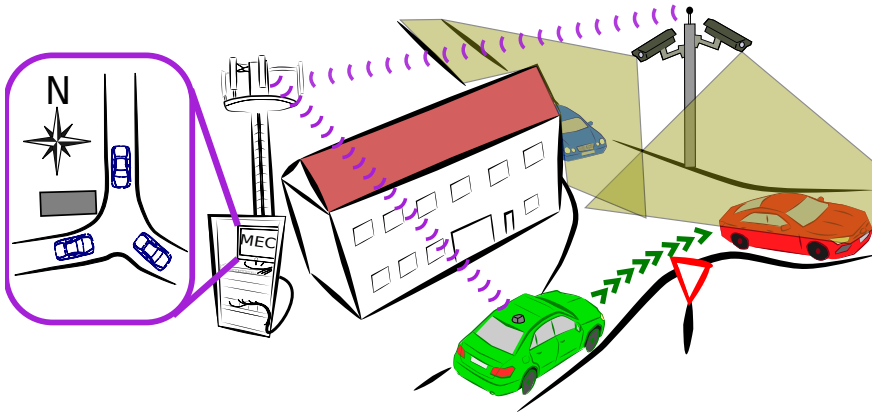


Figure 2.1: Application Scenario. A CAV dynamically merges into an occluded yield junction using the data of an ITSS as external perception.

involves further challenges: If several other road users travel on the main road, the CAV has to decide which traffic gap it uses when merging. If, due to the traffic situation, dynamically merging is not possible at all, the CAV has to detect this and stop at the yield line. Moreover, the merging maneuver requires precise timing and the CAV needs to react to the motion of the other road users. Since the junction contains a narrow S-shaped curve, the CAV has to account for the lateral control of the vehicle when navigating through the junction. Particularly, this imposes velocity constraints to the longitudinal motion planning. Furthermore, the necessary slow down of the CAV imposes additional challenges to the maneuver planning, since the maneuver needs a significant amount of time, even in the best case. The CAV must therefore plan over a long time horizon (up to 10 s), which involves two major challenges: First, the long time horizons are difficult to the numerical optimization used in motion planning schemes, since the computational complexity for a fixed discretization time grows with an increasing horizon. Secondly, the planning scheme needs to deal with high uncertainties in the predicted states of the other road users.

For the application scenario, the overall system consists of four main components [BMH+21]. First, a multi-sensor network observes the junction, where sensor processing units (SPUs) are used to preprocess the sensor data and to forward the detection lists. Second, a low-latency mobile network is used to transmit the data. Third, a MEC server tracks the objects in the scene and predicts their future trajectory. The sensor network and the MEC server together are referred to as infrastructure-based traffic surveillance system (ITSS). Finally, there are the CAVs that use the external data for their motion planning. The system overview, the data flow and the modules of the respective subsystems are illustrated in Fig. 2.2. The reliability estimation and the motion planning are in the focus of this thesis and thus are highlighted.

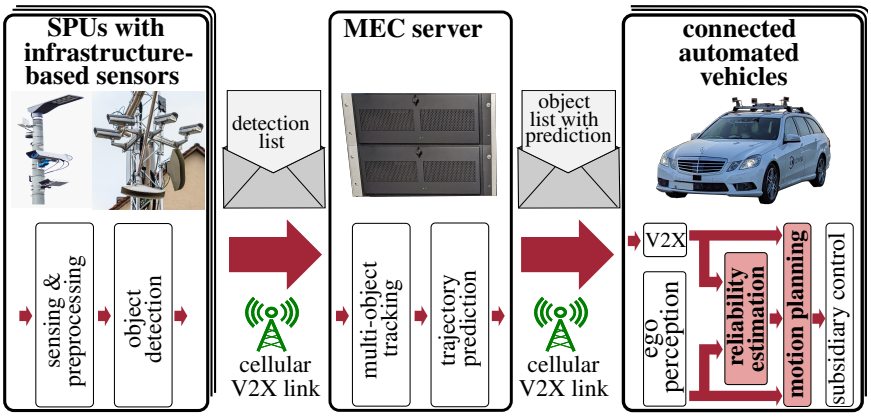


Figure 2.2: System overview with data flow and modules.

2.2 Infrastructure-based Traffic Surveillance System

The infrastructure-based multi-sensor network and the MEC server together form an ITSS. In this ITSS, the detections measured by the distributed sensor network are transmitted to the MEC server. For time reference, the detection lists contain a timestamp referring to GPS time. On the MEC server, the detections are fed into a Labeled Multi-Bernoulli (LMB) filter [HMSB19; HPS+20] to synchronize and track the objects at the junction. Moreover, the future trajectories of the objects are predicted. In the simplest case, these trajectory predictions are calculated through a Kalman filter and a respective kinematic model, e.g. a constant velocity (CV) model. Yet, this approach gets inaccurate for predictions over some seconds. Thus, for better prediction more recent neural network approaches, e.g. [CRC+19; SBM+20], can be used. The objects and their predicted trajectories then are assembled in an environment model and transmitted to the CAVs, using the low-latency mobile network.

The distributed sensor network consists of two subsystems: a high-price reference measurement system to provide ground truth data and a low-cost subsystem for which the components could be obtained cheaply if optimized for low cost and mass production. Since building up such ITSS requires a lot of efforts, the work was shared within the MEC-View project. At the time of the experiments, the low-cost system was operated by a project partner. In contrast, the reference measurement system was designed, built, and operated by the author and some of his colleagues from Ulm University. For both subsystems, the sensor measurements are preprocessed on sensor processing units (SPUs) yielding detection lists that are sent to the MEC server. The low-cost system consists of eight fixed-beam VMS-16/XX-116B LIDAR sensors from Ingeniuerbüro Spies and seven mono cameras mounted on five street light poles [MEC20; OSR20]. For the cameras, neural network based approaches [AG17; BFG18] are

used in combination with homographies [HZ04, p. 33] to detect the objects in the scene. In turn, the fixed-beam LIDAR sensors calculate detections directly from the reflection profiles of the road users. Since the low-cost system comprises many sensors, a good coverage in a wide field of view (FOV) on and around the junction is achieved. However, by principle, the homographies have a limited, position dependend accuracy, and the single layer fixed-beam LIDAR sensors only provide the object extents in one direction. Thus, the detections only achieve a medium accuracy. Furthermore, no raw data were available, since the subsytem was operated by a project partner. Therefore, the low-cost system is not well-suited to generate ground-truth data. For this reason, the low-cost system is supplemented by the high-precision reference measurement system.

The reference measurement system can be subdivided further into a LIDAR subsystem and a stereo vision subsystem. The LIDAR subsystem consists of four high-price SICK LD-MRS 8 layer laser scanners mounted on two poles which feature measurement accuracy in the decimeter range. With laser scanners, raw data is available and, compared to camera images, the individual scans occupy little storage space. However, distant objects often yield not enough measurement points to identify a point cluster as an object. Furthermore, the sensor FOV is far smaller than that of a camera. Hence, the coverage of the LIDAR subsystem is limited. In turn, the stereo vision subsystem consists of four stereo camera pairs mounted on the same poles as the LIDAR subsystem. The coverage of the stereo vision subsystem is better than that of the LIDAR subsystem. However, its accuracy is lower and depth dependent, because, objects far away from the stereo cameras can be resolved only with limited accuracy. Furthermore, the raw data, i.e. the images, occupy a lot of disk space which limits the length of the recordings. Figure 2.3 sketches the mounting positions and the coverage of the low-price and the LIDAR subsystem, while images from the stereo vision subsystem are used to illustrate its FOV.

To calibrate the reference measurement system, a new automatic calibration method [MHS+19] is used. The method identifies the affine transform from the sensor to the world frame by matching the coordinates of a CAV, as measured in the measurement frame, with the coordinates of the CAV in the world frame provided by a high precision localization of the CAV. The object detection in the sensor frame is conducted in three steps: first, the background is subtracted from the scene, then a DBSCAN [EKSX+96] clustering is used to extract the objects, and finally, the extents and center positions of the objects are estimated through a box fitting. For the LIDAR subsytem, the box detection is computed in about 6 ms [MHS+19] and hence is used for online object detection. In contrast, for the stereo vision subsystem, approaches based on neural networks are used. The calibration accuracy is in the range of the sensor accuracy [MHS+19], and hence sufficient for the task.

On the MEC server, an LMB tracking approach is carried out to compute the external environment model on the basis of reported detections [HMSB19]. To study the LMB tracking, on the MEC server, the data from the subsystems can be switched on and off as required. This, however, goes beyond the scope of this thesis.

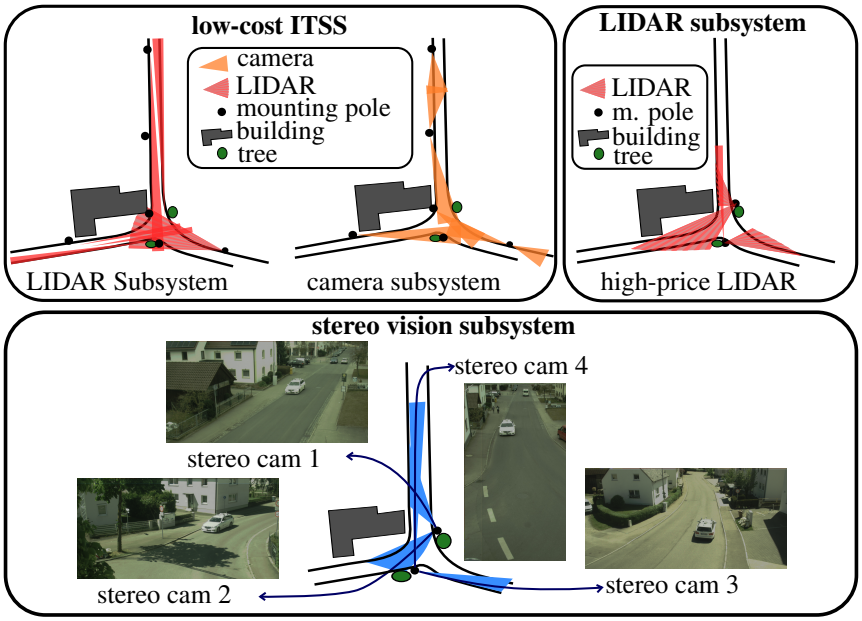


Figure 2.3: Field of views and mounting positions of the sensors of the low-cost and the LIDAR subsystems. The stereo vision subsystem is visualized through sample images.

For this thesis, only two configurations are relevant: first, the reliability estimation is evaluated using the LIDAR subsystem only. Using only LIDAR data allows for long recordings of raw data where a reduced detection rate is observed in the weakly covered areas. Thus, the reliability estimation can be tested for several runs in a row and the recordings can later be split into sequences for evaluation. Furthermore, due to the reduced detection rate in the weakly covered areas, the reliability estimation is confronted with an increased number of detection errors. Therefore, more statistical evidence on the sensitivity of the reliability estimation can be obtained. Moreover, since the LIDAR subsystem provides high accurate raw data, high-quality ground truth data can be derived from the recordings by manual labeling. Second, the motion planning is evaluated using only data from the low-cost ITSS. Thus, it is demonstrated that the proposed planning scheme can deal with the medium accuracy of the low-cost subsystem. Furthermore, since the low-cost system is designed to use only production-ready technologies, it is expected that systems with similar characteristics will reach market maturity in the near future. Therefore, the exclusive use of data from the low-cost system demonstrates that the motion planning requirements for the external perception are not only technically feasible, but also economically viable. Still, the LIDAR subsystem may serve as an additional reference, while the stereo vision subsystem may contribute visuals for illustration.

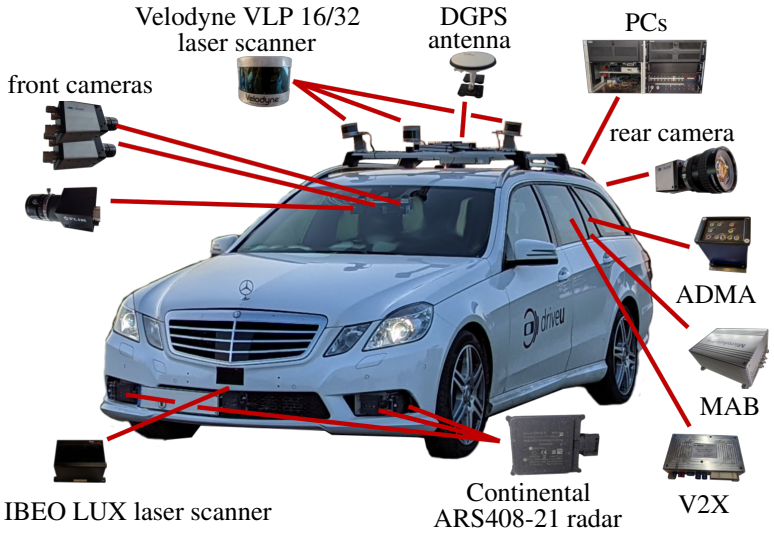


Figure 2.4: Hardware setup of the CAV.

2.3 Connected Automated Vehicle

The system architecture of the CAV consists of six major subsystems: the sensors, the ego perception, the V2X communication, the reliability estimation, the motion planning, and the subsidiary control. Figure 2.4 shows the hardware setup of the CAV. To sense the 360° medium range around the CAV, the CAV is equipped with two Velodyne VLP-32 LIDAR sensors and two Velodyne VLP-16 LIDAR sensors. Furthermore, for the medium range, an IBEO LUX 4 layer laser scanner is mounted in the front of the CAV, providing a 120° front view. Visual cues on the CAV's environment are provided by three front and one rear camera. To complement the sensing, four ARS 408-21 Premium radar sensors from Continental are mounted in front of the CAV, where the radar targets are preprocessed with a neural network based object classifier [DGBD19]. For localization, the CAV has an GeneSys ADMA real time kinematic (RTK) system on board. For the MEC-View project, the CAV is additionally equipped with a V2X communication module, which communicates via a cellular channel.

The sensor raw data from the sensor layer are processed in the perception layer consisting of object detectors, a LMB multi-object tracking (MOT) [RVVD14], and a localization and ego state estimation module. The processing result of the perception layer is assembled in an environment model and forwarded to the motion planning. In parallel, the communication layer handles the V2X communication with other agents, such as the ITSS, maps the received external information to the internal data format of the CAV, and provides the external

environment model to the motion planning and the reliability estimation. The reliability estimation determines the reliability of the external environment model based on consistency and plausibility tests (see Chapter 5.2) and provides it to the motion planning. Classical approaches like [GDMH19] would merge the ego environment model and the external environment model through T2T fusion before using the merged environment model in the motion planning. However, in this thesis, a motion planning scheme is proposed that handles the ego and the external environment model in parallel (see Chapter 6).

From the motion planning layer, the subsidiary control layer receives a reference trajectory. While tracking this reference, the subsidiary control layer stabilizes the CAV on the road and interfaces with the actuators to set the requested set values. For low level stabilization, a control concept of Kehl [Keh07] is adopted [KNW+15]. Besides, the subsidiary control layer implements the safety functions such as the fall-back to the safety driver. To guarantee safety, the subsidiary control layer is implemented on a separate computer, a MicroAutoBox (MAB) 2 real time system [KNW+15].

To comply with legal regulations, the CAV has a test vehicle approval, which demands the CAV to be operated according to a safety concept. The safety concept demands the CAV to be driven by a specially skilled safety driver. Furthermore, a co-driver is required, who continuously monitors the systems of the prototype while carrying out test drives. Since neither driver nor co-driver can see into the occluded junction of the application scenario, the co-driver additionally has to be in steady communication with additional track marshals, who observe the occluded areas and report oncoming road users. Thus, the co-driver can detect possible malfunctions in the external perception. With these safety measures, a full-human fail-safe layer is implemented.

Chapter 3

Fundamentals

In this chapter, the fundamentals of this thesis are summarized. First, the basics of SL are presented. SL is used to estimate the reliability of the external information that serves as input for the motion planning scheme. Subsequently, the fundamentals of optimal control and model predictive control are discussed, which constitute the basis of the proposed motion planning scheme. Finally, the key ideas of set-based methods are summarized. The concepts of set-based methods are employed in the motion planning scheme, since they allow the planning scheme to inherit their safety guarantees.

3.1 Subjective Logic

SL is an emerging mathematical theory that is designed to cope with small sample sizes. It is an extension of classical probability theory that explicitly models the statistical uncertainty within a probabilistic statement. The brief introduction given in this section covers the fundamentals of this thesis. It is primarily based on the text book [Jøs16] by A. Jøsang, the driving force behind the development of the theory.

3.1.1 Opinion Representation

The key element of SL is the so-called opinion, as defined in Definition 3.1. It comprises three main elements: first, the belief mass $\boldsymbol{b}_A$, which reflects the statistical evidence supporting the opinion, second, the statistical uncertainty u_A, which models the lack of knowledge within an opinion, and third, the statistical prior $\boldsymbol{a}_A$, which contains a prior information.

Definition 3.1 (SL Opinion [Jøs16, p. 30])
Let $\mathbb{X}$ *be a domain and* $W = \text{card}\{\mathbb{X}\} \geq 2$*, where* $\text{card}\{\cdot\}$ *is the cardinality. Let X further*

be a random variable in $\mathbb{X}$ and let A be the observing information source. A SL opinion (opinion in short) of A on X is an ordered triple $\omega_A = (\boldsymbol{b}_A, u_A, \boldsymbol{a}_A)$ with

$$\boldsymbol{a}_A : \mathbb{X} \mapsto [0,1]^W, \qquad 1 = \sum_{i=1}^{W} a_{A,i}\,, \tag{3.1a}$$

$$\boldsymbol{b}_A : \mathbb{X} \mapsto [0,1]^W, \qquad 1 = u_A + \sum_{i=1}^{W} b_{A,i}\,. \tag{3.1b}$$

With Theorem 3.1, the so-called bijective mapping theorem, an SL opinion can be represented in terms of a Dirichlet distribution

$$\mathrm{Dir}(\boldsymbol{p}, \boldsymbol{r}_A, \boldsymbol{a}_A) = \frac{\Gamma\left(\sum_{i=1}^{W}(r_{A,i} + a_{A,i}W)\right)}{\prod_{i=1}^{W}\Gamma(r_{A,i} + a_{A,i}W)} \prod_{i=1}^{W} p_i^{\,r_{A,i}+a_{A,i}W-1}\,, \tag{3.2}$$

where $\Gamma(\,\cdot\,)$ is the gamma function [BSMM07], $\boldsymbol{r}$ corresponds to the statistical evidence, and $\boldsymbol{p}$ is the free variable of the distribution. The so-called projected probability

$$\boldsymbol{P}_{A,\perp} = \boldsymbol{b}_A + u_A\,\boldsymbol{a}_A\,, \tag{3.3}$$

is an ordinary probability which is equivalent to the expectation of the corresponding Dirichlet distribution. According to [Bis06, p. 76], the conjugate prior of the Dirichlet distribution is

$$\mathrm{Mult}(\boldsymbol{r}|\boldsymbol{p}, N) = \frac{N!}{\prod_{i=1}^{W} r_i!} \prod_{i=1}^{W} p_i^{r_i}, \tag{3.4}$$

i.e., the multinomial distribution, which describes an urn filled with balls of multiple colors from which N balls are drawn and returned. From the single drawing probabilities $\boldsymbol{p}$, the multinomial distribution yields a vector $\boldsymbol{r}$ counting the drawn balls of the respective colors.

Theorem 3.1 (Bijective Mapping (cf. [Jøs16, pp.36-37]))
Let $\omega_A = (\boldsymbol{b}_A, u_A, \boldsymbol{a}_A)$ be a SL opinion and $\mathrm{Dir}(\boldsymbol{p}, \boldsymbol{r}_A, \boldsymbol{a}_A)$ a Dirichlet distribution over the same $X \in \mathbb{X}$ with $W = \mathrm{card}\{\mathbb{X}\}$. Then, the equivalent mapping

$$\left.\begin{aligned} \boldsymbol{b}_A &= \tfrac{\boldsymbol{r}_A}{W + \sum_{i=1}^{W} r_{A,i}} \\ u_A &= \tfrac{W}{W + \sum_{i=1}^{W} r_{A,i}} \end{aligned}\right\} \iff \left\{\begin{aligned} \boldsymbol{r}_A &= \tfrac{W\boldsymbol{b}_A}{u_A} \\ 1 &= u_A + \textstyle\sum_{i=1}^{W} b_{A,i} \end{aligned}\right., \tag{3.5}$$

transforms the Dirichlet distribution into the SL opinion and vice versa.

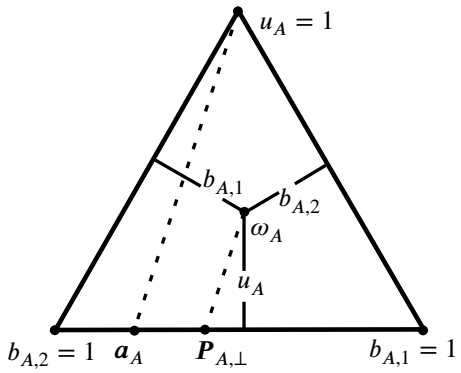


Figure 3.1: Barycentric Triangle (cf. [Jøs16, p. 25]).

Proof. see [Jøs16, p. 36]. □

An important special case of SL opinions is the so-called binomial opinion, for which $W = 2$ holds. In this case, the Dirichlet distribution reduces to a β-distribution and $b_{A,2}$ is called *disbelief*. Opinions with $W > 2$ are called multinomial opinions. Binomial opinions can be visualized through a *barycentric triangle*, a triangle that has belief $b_{A,1}$, disbelief $b_{A,2}$ and the statistical uncertainty u_A at its corners. Due to (3.1), each point within the triangle corresponds to a specific opinion. A barycentric triangle is illustrated in Fig. 3.1.

To measure similarity between two opinions, SL provides the Degree of Conflict (DC)

$$DC(\omega_A, \omega_B) = \frac{1}{2} \sum_{i=1}^{W} \left| P_{A,\perp,i} - P_{B,\perp,i} \right| (1 - u_A)(1 - u_B) \in [0, 1]\,. \tag{3.6}$$

Here, similarity is measured between the opinions ω_A and ω_B by comparing their projected probabilities weighted with the amount of statistical evidence the opinions come with. Thus, the less statistical uncertainty is included in the opinions, the higher is the DC for the same absolute difference of the projected probabilities. In turn, the DC decreases with decreasing absolute difference of the projected probabilities.

3.1.2 Fusion Operators

To accumulate knowledge, different opinions need to be combined through fusion operators. In fact, one of the key strengths of SL is that it provides numerous fusion operators to combine opinions under various scenarios. A selection of such SL operators used through this thesis

is presented in the following. The first important operator is the aleatory cummulative belief fusion (CBF) operator defined in Definition 3.2. It is intended to combine statistical independent information.

Definition 3.2 (Aleatory Cummulative Belief Fusion (cf. [Jøs16, p. 226]))
Let ω_A and ω_B be sources A's and B's respective opinions over the same variable X on domain $\mathbb{X}$. Then, the operator $\oplus$ in

$$\omega_A \oplus \omega_B = \begin{cases} \boldsymbol{b} &= \frac{\boldsymbol{b}_A u_B + \boldsymbol{b}_B u_A}{u_A + u_B - u_A u_B} \\ u &= \frac{u_A u_B}{u_A + u_B - u_A u_B} \\ \boldsymbol{a} &= \frac{\boldsymbol{a}_A u_B + \boldsymbol{a}_B u_A - (\boldsymbol{a}_A + \boldsymbol{a}_B) u_A u_B}{u_A + u_B - 2u_A u_B} \end{cases}, \tag{3.7}$$

where $0 < u_A < 1$ and $0 < u_B < 1$, is called aleatory cummulative belief fusion. For accumulating multiple opinions from a set S_ω with N elements, $\bigoplus_{\forall \omega \in S_\omega} \omega = \omega_1 \oplus \cdots \oplus \omega_N$ is used as shorthand.

In contrast, if the pieces of statistical evidence are correlated, the weighted average belief fusion (WABF) operator is more appropriate. Its definition, see Definition 3.3, as proposed in [MB21], is a generalization of the definition in [Jøs16] that includes the average belief fusion operator as well as the uncertainty-weighted average belief fusion operator.

Definition 3.3 (Weighted Average Belief Fusion (cf. [Jøs16, pp. 230-232]))
Let ω_A and ω_B be opinions over the same variable X in the domain $\mathbb{X}$. Then, for the weights c_a, c_B and $u_A \neq 0$, $u_B \neq 0$, the operator $\tilde{\oplus}$ in

$$\omega_A \underset{c_A, c_B}{\tilde{\oplus}} \omega_B = \begin{cases} \boldsymbol{b}_{A\tilde{\oplus}B} = \frac{\boldsymbol{b}_A c_A u_B + \boldsymbol{b}_B c_B u_A}{c_B u_A + c_A u_B} \\ u_{A\tilde{\oplus}B} = \frac{(c_A + c_B) u_A u_B}{c_B u_A + c_A u_B} \\ \boldsymbol{a}_{A\tilde{\oplus}B} = \frac{\boldsymbol{a}_A c_A + \boldsymbol{a}_B c_B}{c_A + c_B} \end{cases} \tag{3.8}$$

is called weighted average belief fusion operator. If $c_A = c_B = 1$ holds, the operator is called aleatory average belief fusion (ABF), denoted $\underline{\oplus}$. For $c_A = (1 - u_A)$ and $c_B = (1 - u_B)$, the operator $\tilde{\oplus}$ is called uncertainty weighted average belief fusion, denoted $\hat{\oplus}$. Similar to Definition 3.2, the shorthand $\tilde{\bigoplus}_{\omega \in S_\omega} \omega$ is used to merge multiple opinions from a set S_ω.

Finally, the trust discounting operator specified in Definition 3.4 uses a discount probability p_d to transform belief mass into statistical uncertainty. This way, information degradation can be modeled.

Definition 3.4 (Trust Discounting (cf. [Jøs16, p. 256]))
Let $\hat{\omega}_A = (\hat{\boldsymbol{b}}_A, \hat{u}_A, \hat{\boldsymbol{a}}_A)$ be an opinion and let $p_d \in [0,1]$ be a probability. Then, the trust discounting operator $\mathcal{D}(\hat{\omega}_A, p_d)$ is defined as

$$\omega_{td} = \mathcal{D}(\hat{\omega}_A, p_d) = \begin{cases} \boldsymbol{b}_{td} & = p_d \, \hat{\boldsymbol{b}}_A \\ u_{td} & = 1 - p_d \sum_{j=1}^{W} \hat{b}_{A,j} \\ \boldsymbol{a}_{td} & = \hat{\boldsymbol{a}}_A \end{cases} . \tag{3.9}$$

3.2 Optimal Control and Model Predictive Control for Motion Planning

In literature, several categories of motion planning approaches have been proposed so far [PCY+16]. One of these categories are methods based on Model Predictive Control (MPC). Since the vehicle dynamics can be described using comparatively simple, differentiable physical models with sufficient accuracy and the problem of moving a vehicle from its current state to a target state, i.e., an intended final state, offers several degrees of freedom, methods based on optimal control or MPC appear particularly well-suited for the task. The principle idea behind these approaches is to formulate the deviation from the goals of the vehicle in terms of a cost function and to use mathematical optimization to generate trajectories that fulfill the goals of the vehicle as well as possible while meeting its constraints. The constraints comprise the system dynamics of the model, equality and inequality constraints, and terminal equality and inequality constraints. In this section, the fundamentals on optimal control and MPC used for the motion planning scheme proposed in this thesis are summarized, mainly from [EVM+19; GGD18; Kir04; KS12; LVS12; SK16].

Consider the general, nonlinear system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \, , \; \boldsymbol{x}(0) = \boldsymbol{x}_0 \, , \tag{3.10}$$

with $\boldsymbol{f}(\cdot)$ the nonlinear system dynamics function of the states $\boldsymbol{x}(t)$, the control variable $\boldsymbol{u}(t)$, the time t, and $\boldsymbol{x}_0$ being the initial state at initial time $t_0 = 0$. For the system, a trajectory

$$\mathcal{T} \, : \, t \in [0, t_f] \mapsto \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{u}(t) \end{bmatrix} \tag{3.11}$$

is a solution of (3.10) for a specific time interval $t \in [0, t_f]$ and a specific time course of the control variable $\boldsymbol{u}(t)$, where t_f denotes the final time. The goal of an optimal control is to determine the optimal control variable $\boldsymbol{u}^*(t)$ and thus the optimal trajectory $\mathcal{T}^*$ that moves the system from its initial state $\boldsymbol{x}_0$ to its optimal final state $\boldsymbol{x}_f^*$ at the optimal final time t_f^*

while minimizing the cost function

$$J(\boldsymbol{x}(t), \boldsymbol{u}(t), t_f) = V(\boldsymbol{x}(t_f), t_f) + \int_0^{t_f} l(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\mathrm{d}t\,. \tag{3.12}$$

At the same time, the equality constraints $\boldsymbol{g}(\,\cdot\,)$, inequality constraints $\boldsymbol{h}(\,\cdot\,)$, and terminal constraints $\boldsymbol{g}_T(\,\cdot\,)$ and $\boldsymbol{h}_T(\,\cdot\,)$, respectively, have to be fulfilled. In (3.12), the terminal cost function $V(\cdot)$ describes the cost of the final state at the final time, while the integral cost $l(\cdot)$ describes its accumulating dynamic cost. Mathematically, this can be described as a general Optimal Control Problem (OCP)

$$\boldsymbol{u}^*(t) = \arg\min_{\boldsymbol{u}(t),\, t_f} \{J(\boldsymbol{x}(t), \boldsymbol{u}(t), t_f)\} \quad \text{subject to} \tag{3.13a}$$

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\,,\ \boldsymbol{x}(0) = \boldsymbol{x}_0\,, \tag{3.13b}$$

$$\boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), t),\ \boldsymbol{0} = \boldsymbol{g}_T(\boldsymbol{x}(t_f), t_f)\,, \tag{3.13c}$$

$$\boldsymbol{0} \geq \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t), t),\ \boldsymbol{0} \geq \boldsymbol{h}_T(\boldsymbol{x}(t_f), t_f)\,, \tag{3.13d}$$

$$t_f \in [t_{f,\min}\,,\, t_{f,\max}]\,. \tag{3.13e}$$

For the proposed motion planning scheme, two system models are used to describe the vehicle's dynamics: a point-mass-model for the longitudinal planning (see Section 6.2.4) and a nonlinear kinematic model for lateral trajectory tracking (see Section 6.2.5). The 1D-point-mass model

$$\dot{\boldsymbol{x}}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} s \\ v \\ a \end{bmatrix}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} s \\ v \\ a \end{bmatrix}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t)\,, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0\,, \tag{3.14a}$$

$$\boldsymbol{0} = \boldsymbol{x}(t_f) - \boldsymbol{x}_f\,, \tag{3.14b}$$

$$\boldsymbol{0} \geq \begin{bmatrix} v(t) - v_{\max}(t), & -v(t), & a(t) - a_{\max}, & a_{\min} - a(t) \end{bmatrix}^T\,, \tag{3.14c}$$

$$t_f \in [0, t_{f,\max}]\,, \tag{3.14d}$$

comprises the longitudinal jerk $u(t)$, the state $\boldsymbol{x}$, which consists of the longitudinal position s, velocity v, and acceleration a, the final state $\boldsymbol{x}_f$, the maximum longitudinal velocity $v_{\max}$, the longitudinal acceleration bounds $a_{\min}$ and $a_{\max}$, and the maximum optimization horizon $t_{f,max}$. Note that the control variable, i.e. the longitudinal jerk $u(t)$, is not constrained in (3.14). Hence, emergency braking, which instantly sets the acceleration to $a_{\min}$ and thus produces $u(t) \to -\infty$ at this instant of time, still yields a valid solution. Nonetheless, since the cost function depends on the squared longitudinal jerk, such solution has infinite cost (see Chapter 6.2). The model is used if, due to a subordinate control, the lateral dynamics can be neglected.

If, on the other hand, the effects of lateral dynamics must be taken into account, the nonlinear vehicle model [SK16, pp. 1238-1242]

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}_{\text{NLVM}}\big(\boldsymbol{x}(t), \boldsymbol{u}(t), t\big), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0\,, \tag{3.15a}$$

$$\begin{aligned}\mathbf{0} \geq \big[&x_5 - v_{\max}(t),\ -x_5,\ a_\perp^2(t) - a_{\perp,\max}^2,\ x_4^2 - x_{4,\max}^2,\\ &u_{1,\min} - u_1(t),\ u_1(t) - u_{1,\max},\ a_{\min} - u_2(t),\ u_2(t) - a_{\max}\big]^T\end{aligned} \tag{3.15b}$$

is applied, where the vehicle dynamics $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}_{\text{NLVM}}\big(\boldsymbol{x}(t), \boldsymbol{u}(t), t\big)$ are described by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}_{\text{NLVM}}\big(\boldsymbol{x}(t), \boldsymbol{u}(t), t\big) = \begin{cases} \dot{x}_1 = x_5\ \cos(x_3)\,, & (3.16a)\\ \dot{x}_2 = x_5\ \sin(x_3)\,, & (3.16b)\\ \dot{x}_3 = \dfrac{x_5 \cdot x_4 \cdot v_{\text{char}}^2}{l_{\text{veh}}\,(v_{\text{char}}^2 + x_5^2)}\,, & (3.16c)\\ \dot{x}_4 = u_1\,, & (3.16d)\\ \dot{x}_5 = u_2\,, & (3.16e)\\ \dot{x}_6 = x_5\ \sin(x_3 - x_7)\,, & (3.16f)\\ \dot{x}_7 = \dfrac{x_5 \cdot \cos(x_3 - x_7) \cdot \kappa_{\text{ref}}(x_8)}{1 - x_6 \cdot \kappa_{\text{ref}}(x_8)}\,, & (3.16g)\\ \dot{x}_8 = \dfrac{x_5 \cdot \cos(x_3 - x_7)}{1 - x_6 \cdot \kappa_{\text{ref}}(x_8)}\,. & (3.16h) \end{cases}$$

In (3.15) and (3.16), x_1 and x_2 describe the position of the vehicle in Cartesian coordinates, x_3 the course angle of the vehicle, x_4 the wheel angle, x_5 the longitudinal velocity, x_6 the lateral deviation from the reference path, x_7 the reference course angle, x_8 the longitudinal position along the reference path $\mathcal{P}$, and κ_{ref} the curvature of $\mathcal{P}$ at position x_8. The lateral acceleration is calculated by

$$a_\perp = \frac{x_5^2 \cdot x_4 \cdot v_{\text{char}}^2}{l_{\text{veh}}\,(v_{\text{char}}^2 + x_5^2)}\,, \tag{3.17}$$

where l_{veh} is the vehicle length and v_{char} is the characteristic velocity, i.e. the velocity of maximum steering sensitivity. The maximum acceptable lateral acceleration is $a_{\perp,\max}$. Moreover, the control variables are the steering wheel angle rate u_1 and the longitudinal acceleration u_2. Figure 3.2 illustrates the nonlinear model (3.16).

The nonlinear model (3.16) can be subdivided into two parts: the upper part, (3.16a) - (3.16e), describes the actual vehicle dynamics, while the lower part, (3.16f) - (3.16h), comprises an observer structure that allows to estimate the vehicle's position in a Frenet frame. Thus, calculation-intense transformations of the vehicle's position between Frenet coordinates

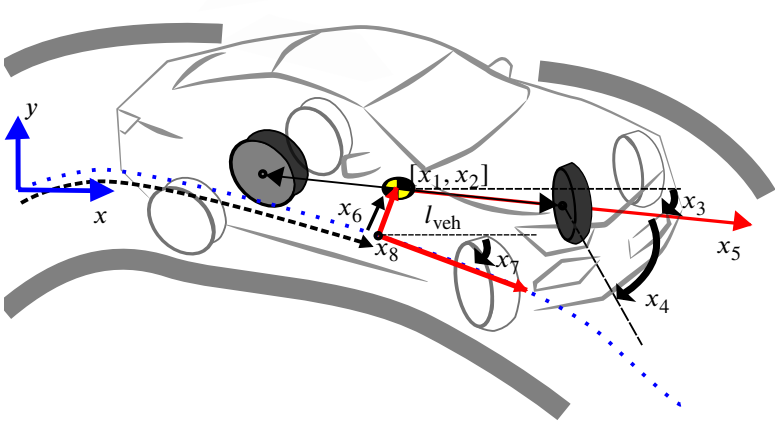


Figure 3.2: Illustration of the nonlinear vehicle model (cf.[GGD18]).

and Cartesian coordinates can be avoided during the optimization. This, however, comes at the cost that the model needs the reference path $\mathcal{P}$ the vehicle is intended to follow in terms of the longitudinal position along $\mathcal{P}$ and its curvature $\kappa_{\text{ref}}(s)$ rather than in Cartesian (x, y)-coordinates. The respective transformation is calculated with the *curvature MPC* from [GGD18], which solves the OCP

$$u(s)^* = \arg\min_u \left\{ \int_0^{s_f} w_1 \left(x(s) - x_{\text{ref}}\right)^2 + w_2 \left(y(s) - y_{\text{ref}}\right)^2 + w_3 \, u^2(s) \, \mathrm{d}s \right\} \text{ s.t.} \tag{3.18a}$$

$$\boldsymbol{x}'(s) = \frac{\delta}{\delta s}[x,\, y,\, \varphi]^T(s) = [\cos(\varphi),\, \sin(\varphi),\, u]^T(s),\; \boldsymbol{x}(0) = [x_{\text{ref},0},\, y_{\text{ref},0},\, 0]^T\,, \tag{3.18b}$$

where w_1, w_2, w_3 are weights, and the curvature is used as control variable $u = \kappa_{\text{ref}}$. Furthermore, $[s_{x,\text{ref}},\, s_{y,\text{ref}}]^T(s)$ are the reference positions determined from the path $\mathcal{P}$ according to

$$\begin{bmatrix} s_{x,\text{ref}} \\ s_{y,\text{ref}} \end{bmatrix}(s) = \begin{bmatrix} x_{j-1} \frac{s_j - s}{s_j - s_{j-1}} + x_j \frac{s - s_j}{s_j - s_{j-1}} \\ y_{j-1} \frac{s_j - s}{s_j - s_{j-1}} + y_j \frac{s - s_j}{s_j - s_{j-1}} \end{bmatrix}. \tag{3.19}$$

In (3.19), j describes the subscripts of the points in $\mathcal{P}$. A specific of the curvature MPC is that it is formulated over the 1-D position s along $\mathcal{P}$ rather than over time.

To solve the OCP (3.13), there are basically two ways: the first one is to turn the OCP into a static parameter optimization problem through discretization and solving the high-dimensional problem using numerical methods. This approach is taken e.g. by ACADO [HFD11]. Alternatively, the OCP (3.13) can be solved indirectly by determining a solution for the optimality conditions. The indirect approach has the advantage that it allows to

find exact analytical solutions for simple OCPs, while it can still be backed by tools like GRAMPC [EVM+19] to solve more complex OCPs numerically. Further details on the numerical optimization strategies can be found in Appendix A.2.

To derive first-order optimality conditions, the Hamiltonian [KS12]

$$H(\boldsymbol{x}(t),\, \boldsymbol{u}(t),\, \lambda(t),\, t) = l(\boldsymbol{x}(t), \boldsymbol{u}(t), t) + \lambda(t)^T\, \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \tag{3.20}$$

with co-states $\boldsymbol{\lambda}$ is introduced for a reduced OCP that contains neither inequality constraints (3.13d) nor equality constraints (3.13c).

Subsequently, the optimality conditions

$$\dot{\boldsymbol{x}}(t) = \frac{\delta H}{\delta \boldsymbol{\lambda}}(\boldsymbol{x},\, \boldsymbol{u},\, \lambda,\, t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\,, \quad \boldsymbol{x}^*(0) = \boldsymbol{x}_0\,, \tag{3.21a}$$

$$\dot{\lambda}(t) = -\frac{\delta H}{\delta \boldsymbol{x}}(\boldsymbol{x},\, \boldsymbol{u},\, \lambda,\, t)\,, \quad \lambda^*(t_f) = \left.\frac{\delta V}{\delta \boldsymbol{x}}\right|_{t=t_f}, \quad t_f \in [t_{f,\min}, t_{f,\max}]\,, \tag{3.21b}$$

$$\boldsymbol{u}^*(t) = \arg\min_{\boldsymbol{u}} \{ H(\boldsymbol{x}^*(t),\, \boldsymbol{u}(t),\, \lambda^*(t),\, t) \}\,, \quad \boldsymbol{u}(t) \in [\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}]\,, \tag{3.21c}$$

follow with Pontryagin's Maximum Principle [KS12, p. 219]. In any case, the set of differential equations (3.21), also called canonical equations [Gra15], in combination with the initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$ and the transversality condition $\lambda(t_f) = \frac{\delta V}{\delta t}(t_f)$ or a terminal condition $\boldsymbol{x}(t_f) - \boldsymbol{x}_f = \boldsymbol{0}$ form a Boundary Value Problem (BWP). Under the assumption that the solution of the OCP is unique in an environment around the initial solution, the solution of the BWP then yields the optimal trajectory $\mathcal{T}^*$. At this point, it should be noted that the uniqueness assumption corresponds to the assumption that a determined local optimum actually is the global optimum and thus is not specific to the indirect approach. With the augmented Lagrangian approach, optimality conditions for the constraint optimization problem (3.13) can be derived. Details on the augmented Lagrangian approach can be found in Appendix A.1.

3.3 Set-based Methods for Motion Planning

Compared to probabilistic approaches, such as SL, set-based methods use a completely different way to express uncertainty within a system. The key idea of set-based methods is to assemble all possible outcomes of a random experiment in sets, where bigger sets correspond to higher uncertainty [MKBH18; PKWA17]. Since all possible outcomes of the random experiment are covered by the output set, the description of the system becomes deterministic and formal guarantees can be derived. Yet, the results are usually very conservative and computationally expensive to calculate [AGH+20].

In this thesis, set-based methods are adopted for safe motion planning. Particularly, the concepts Point of No Return (PNR) and Point of Guaranteed Arrival (PGA) are adopted from the set-based approach [PKWA17]. This section briefly summarizes these concepts. Both, the PNR and the PGA settle upon an ISS as defined in Definition 3.5.

Definition 3.5 (Invariably Safe Set [PKWA17])
Consider the general, nonlinear dynamic system (3.10)*, for which* $\mathcal{X}(t_f,\, \boldsymbol{x}_0, \boldsymbol{u}(t))$ *with* $t \in [t_0, t_f]$ *denotes the solution of* (3.10)*. Let furthermore* $\mathcal{S}^t_{safe}$ *denote the maximum set of safe, i.e. collision-free, states at time t. Then, a feedback control law* $\boldsymbol{\Phi}_{safe}$ *for the given reference* $\mathcal{T}_{ref}$

$$\boldsymbol{\Phi}_{safe} \,:\, \mathcal{X}(t_f, \boldsymbol{x}_0, \boldsymbol{\Phi}_{safe}(\boldsymbol{x}(t), \mathcal{T}_{ref})) \in \mathcal{S}_{safe}(t),\ t \in [t_0, t_f] \tag{3.22}$$

is called safe feedback control law. *For such a safe feedback control law* $\boldsymbol{\Phi}_{safe}$ *and a given instance in time t, the set*

$$\mathcal{S}_{ISS}(t) = \left\{ \boldsymbol{x}(t) \in \mathcal{S}_{safe}(t) \,\middle|\, \forall \tau > t \,:\, \mathcal{X}(\tau, \boldsymbol{x}(t), \boldsymbol{\Phi}_{safe}(\boldsymbol{x}(\tilde{\tau}), \mathcal{T}_{ref})) \in \mathcal{S}_{safe}(t);\ \tilde{\tau} \in [t, \tau] \right\} \tag{3.23}$$

is termed invariably safe set.

The ISSs are constructed such that the safety of a trajectory is formally guaranteed if the whole trajectory is included in such an ISS [PKWA17]. Nevertheless, if a trajectory is not fully included in an ISS, this does not necessarily mean the trajectory leads to a collision. It rather tells that the trajectory contains states with non-zero residual risk of ending up in a collision, since the trajectory might comprise temporarily safe states $\boldsymbol{x}(t) \in \mathcal{S}_{\text{safe}}(t) \setminus \mathcal{S}_{\text{ISS}}(t)$. Note that, in practice, it is hardly possible to determine an ISS $\mathcal{S}_{\text{ISS}}(t)$ exactly, since the ISS might be of complex shape and its calculation requires to know the maximum set of safe states $\mathcal{S}_{\text{safe}}(t)$ in the first place, which as well is impracticable. Particularly, to determine $\mathcal{S}_{\text{safe}}(t)$ exactly, the system would need to be free of any uncertainty, because otherwise it is not possible to tell which states at a given time are safe and which are not. Hence, these sets are under-approximated by estimates $\hat{\mathcal{S}}_{\text{ISS}}(t) \subset \mathcal{S}_{\text{ISS}}(t)$ and $\hat{\mathcal{S}}_{\text{safe}}(t) \subset \mathcal{S}_{\text{safe}}(t)$. To find these sets $\hat{\mathcal{S}}_{\text{ISS}}(t)$ and $\hat{\mathcal{S}}_{\text{safe}}(t)$, reachable sets, as defined in Definition 3.6, are used.

Definition 3.6 (Reachable Set [PKWA17])
The reachable set $\mathcal{S}_{reach}(t)$ *of a general nonlinear system* (3.10) *is the set of states which are reachable at time t from a set of initial states* $\mathcal{S}_{\boldsymbol{x}_0}$ *at time* t_0 *and subject to the set of possible inputs* $\boldsymbol{u}(\tau) \in [\boldsymbol{u}_{min}, \boldsymbol{u}_{max}]$*:*

$$\mathcal{S}_{reach}(t) = \left\{ \boldsymbol{x}_0 + \int_{t_0}^{t} \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau) d\tau \,\middle|\, \boldsymbol{x}_0 \in \mathcal{S}_{\boldsymbol{x}_0},\ \boldsymbol{u}(\tau) \in [\boldsymbol{u}_{min}, \boldsymbol{u}_{max}] \right\}. \tag{3.24}$$

First, the set of static safe states $\hat{\mathcal{S}}_{\text{safe}}^{\text{stat}}$ is estimated comprising all states that satisfy the time-invariant constraints of the nonlinear system. Then, the reachable set for all relevant dynamic objects $o \in \mathcal{L}_{\text{rel}}$ is calculated and subtracted from $\hat{\mathcal{S}}_{\text{safe}}^{\text{stat}}$ to get $\hat{\mathcal{S}}_{\text{safe}}$ and $\hat{\mathcal{S}}_{\text{ISS}}(t)$, i.e.

$$\hat{\mathcal{S}}_{\text{safe}}(t) = \hat{\mathcal{S}}_{\text{safe}}^{\text{stat}} \setminus \bigcup_{\forall o \in \mathcal{L}_{\text{rel}}} \mathcal{S}_{\text{reach}}(o,t)\,, \tag{3.25}$$

$$\hat{\mathcal{S}}_{\text{ISS}}(t) = \left\{ \boldsymbol{x}(t) \in \hat{\mathcal{S}}_{\text{safe}}(t) | \forall \tau > t \,:\, \mathcal{X}(\tau, \boldsymbol{x}(t), \boldsymbol{\Phi}_{\text{safe}}(\boldsymbol{x}(\tilde{\tau}), \mathcal{T}_{\text{ref}})) \in \hat{\mathcal{S}}_{\text{safe}}(t);\, \tilde{\tau} \in [t,\tau] \right\}. \tag{3.26}$$

If a trajectory ends up in an ISS, it is guaranteed that some Φ_{safe} keeps the system in the ISS. Thus, for safety reasons, each planned trajectory must at least end up in an ISS. From that, it can be concluded that each trajectory that starts in an ISS but is not fully included in it comprises a PNR, a PGA, and a set of states in between, termed *safety-critical passageway*. In such case, the trajectory leaves the initial ISS at the PNR and enters an ISS again at the PGA. The PNR and PGA are formally defined in Definition 3.7 and Definition 3.8, respectively.

Definition 3.7 (Point of No Return [PKWA17])
The Point of No Return (PNR) is the state $\boldsymbol{x}(t_{PNR})$, $t_{PNR} \in [t_0, t_f]$ along the state trajectory $\boldsymbol{x}([t_0, t_f])$ from which returning to the initial ISS $\mathcal{S}_{ISS}^{init}(t)$ is ultimately possible, i.e.

$$\forall t \in [t_0, t_{PNR}] \,|\, \exists \boldsymbol{u}([t,\tilde{t}_f]) \,:\, \mathcal{X}(\tilde{t}_f, \boldsymbol{x}(t), \boldsymbol{u}([t,\tilde{t}_f]) \in \mathcal{S}_{ISS}^{init}(\tilde{t}_f)\,, \tag{3.27a}$$

$$\forall t \in (t_{PNR}, t_f] \,|\, \nexists \boldsymbol{u}([t,\tilde{t}_f]) \,:\, \mathcal{X}(\tilde{t}_f, \boldsymbol{x}(t), \boldsymbol{u}([t,\tilde{t}_f]) \in \mathcal{S}_{ISS}^{init}(\tilde{t}_f)\,, \tag{3.27b}$$

where $\tilde{t}_f$ is the final time of a trajectory returning to $\mathcal{S}_{ISS}^{init}(\tilde{t}_f)$.

Definition 3.8 (Point of Guaranteed Arrival [PKWA17])
The Point of Guaranteed Arrival (PGA) is the state $\boldsymbol{x}(t_{PGA})$, $t_{PGA} \in [t_0, t_f]$ along the state trajectory $\boldsymbol{x}([t_0, t_f])$, from which the existence of a trajectory ending in its corresponding ISS is guaranteed, i.e.

$$\forall t \in [t_{PGA}, t_f] \,|\, \exists \boldsymbol{u}([t_{PGA},\tilde{t}_f]) \,:\, \mathcal{X}(\tilde{t}_f, \boldsymbol{x}(t), \boldsymbol{u}([t,\tilde{t}_f]) \in \mathcal{S}_{ISS}^{fin}(\tilde{t}_f)\,. \tag{3.28}$$

These concepts are employed in Chapter 6 to inherit formal safety guarantees for the motion planning.

Chapter 4

An Urn Model Intuition to Subjective Logic

In this chapter, based on the author's pre-publication [MB21], a novel urn model is presented, which provides an alternative, more intuitive, interpretation of SL than the original derivation of SL by Jøsang [Jøs97]. Furthermore, the urn model establishes a link between SL and the Pólya urn scheme [EP23], which models branching phenomena such as contagion. The urn model is presented in two steps. First, the urn model is motivated and intuitively derived. Then, the urn model and its drawing schemes are formalized and the key properties of the urn model are mathematically proven.

4.1 Motivation and Intuitive Derivation

Most phenomena that contain uncertainties can be modeled by classical probability theory. For example, the outcome of drawing balls from an urn, the state of an object recovered from noisy measurements, or the reliability of an external perception can be described by probabilities. The respective probabilities are determined empirically, by repeatedly drawing from the urn, measuring a reference object, or comparing the information from an external perception with ground truth data. The law of large numbers [BSMM07, p. 764] then guarantees that the probabilities, which are empirically determined from large amounts of statistical evidence, describe the respective phenomena sufficiently well. However, in many cases, the required amount of statistical evidence can hardly be obtained. For example, empirically proving the accident probability of automated vehicles being sufficiently low is infeasible [RPL+20]. Similarly, in the application scenario (see Section 2.1), the few measurements obtained aboard the CAV while approaching the junction are insufficient to empirically estimate the reliability of the external perception using classical methods. This is due to the fact that classical probability theory ignores the statistical evidence that led to the result, i.e., the statistical uncertainty [Jøs97]. Rather, a statistical uncertainty of zero is implicitly assumed.

To illustrate this, consider a classical urn $\mathcal{U}_1$ filled with red and black balls, symbolized as • and •, respectively, from which balls are drawn. In this thesis, drawing a ball from an urn generally means, a ball is randomly picked from the set of balls in the urn, its type is noted, the ball is returned to the urn and a ball of the same type is emitted for further use. Drawing balls from $\mathcal{U}_1$ hence represents a random experiment over the domain $\mathbb{X} = \{•, •\}$, which has the dimension $W = \text{card}\{\mathbb{X}\} = 2$. Each drawn ball corresponds to a measurement, i.e., a piece of statistical evidence obtained from the system. To estimate the probability p_{red} of drawing a red ball, a set $\mathcal{S}_A$ of N_A balls are drawn from the urn, and the number of red balls $N_{A,\text{red}}$ in $\mathcal{S}_A$ is counted. From this, the maximum likelihood (ML) estimation is obtained by

$$\hat{p}_{A,\text{red,ML}} = \arg\max_{p_{\text{red}}} \left\{ p(\mathcal{S}_A | p_{\text{red}}) \right\} = \frac{N_{A,\text{red}}}{N_A}, \quad \hat{\boldsymbol{p}}_{A,\text{ML}} = \begin{bmatrix} \hat{p}_{A,\text{red,ML}} \\ 1 - \hat{p}_{A,\text{red,ML}} \end{bmatrix}. \tag{4.1}$$

If prior knowledge $\boldsymbol{a} = [a_{\text{red}}, a_{\text{black}}]^T$ and the normalization constant $p(\mathcal{S}_A)$ is available, the result can be improved using the maximum a posteriori (MAP)

$$\hat{p}_{A,\text{red,MAP}} = \frac{\hat{p}_{A,\text{red,ML}}\, a_{\text{red}}}{p(\mathcal{S}_A)}, \quad \hat{\boldsymbol{p}}_{A,\text{ML}} = \begin{bmatrix} \hat{p}_{A,\text{red,MAP}} \\ \hat{p}_{A,\text{black,MAP}} \end{bmatrix} = \begin{bmatrix} \hat{p}_{A,\text{red,MAP}} \\ 1 - \hat{p}_{A,\text{red,MAP}} \end{bmatrix}. \tag{4.2}$$

However, for small sample sizes, neither the ML, nor the MAP estimation yield correct results. For example, if there is an equal number of red and black balls in the urn, i.e., the actual distribution is $\boldsymbol{p}^*_{\mathcal{U}1} = [0.5, 0.5]^T$, and three balls are drawn from the urn, only one of the four values $\hat{p}_{A,\text{red,ML}} \in \{0, 1/3, 2/3, 1\}$ can be estimated. Hence, $\boldsymbol{p}^*_{\mathcal{U}1}$ can never be concluded, even if the correct probabilities are assumed as prior $\boldsymbol{a} = \boldsymbol{p}^*_{\mathcal{U}1}$ for the MAP estimator. Thus, for small sample sizes, the question with which probability $\boldsymbol{p}_{\mathcal{U}1}$ red and black balls are drawn from the urn cannot be answered appropriately. In fact, after drawing $\mathcal{S}_A$ from the urn, still multiple discrete probability distributions $\boldsymbol{p}_A = [p_{A,\text{red}}, 1 - p_{A,\text{red}}]^T$ are likely. So, the question arises of how likely is which $\boldsymbol{p}_A$?

To answer this, some sort of upsampling is needed. At this point two perspectives have to be distinguished. Objectively, each draw from this urn is independent, so the actual probability $\boldsymbol{p}^*_{\mathcal{U}}$ of drawing a red or black ball is always the same. Subjectively, however, the individual draws from the urn are not independent. Since the actual probability $\boldsymbol{p}^*_{\mathcal{U}}$ is unknown, it must be estimated. Thus, the estimate $\hat{\boldsymbol{p}}_{\mathcal{U}1}$ changes with each draw. Particularly, as can be seen from the ML estimator (4.1) and the MAP estimator (4.2), the estimated probability $\hat{p}_{\text{red}}$ of drawing a red ball increases with the number of red balls which have been drawn already. To model this, a draw scheme with reinforcement is needed. The simplest draw scheme with reinforcement is the Pólya urn scheme [EP23], where for each drawn ball an additional ball of the same type is added to the urn. However, the Pólya urn scheme per se neither answers the question about the likelihood of the $\boldsymbol{p}_A$ nor contains a measure for the statistical uncertainty nor supports the use of prior knowledge. Hence, the urn model has to be extended.

First, the objective perspective and the subjective perspective are separated. For that, a second urn $\mathcal{U}_2$ is added to the model. While the first urn continues to objectively describe the actual system, the second urn reflects the knowledge about the system, for which the subjective perspective is taken. Furthermore, a third urn $\mathcal{U}_3$ is added, which represents the prior knowledge on the system. Hence, $\mathcal{U}_3$ contains balls of the same color as the balls in $\mathcal{U}_1$, i.e., red and black balls. In contrast to $\mathcal{U}_1$, for which the distribution of the balls is unknown and has to be estimated, the balls in $\mathcal{U}_3$ are distributed according to the prior $\boldsymbol{a}$, i.e. are known. Since the statistical uncertainty is not applicable to the prior knowledge, which is represented by $\mathcal{U}_3$, the number of balls in $\mathcal{U}_3$ is arbitrary. To model the lack of knowledge, an additional type of balls, the checkered ball, symbolized as ⊕, is introduced. Initially, the second urn $\mathcal{U}_2$ is filled with W of these checkered balls indicating complete ignorance.

To gather information about the system, a small set of balls $\mathcal{S}_A$ is drawn from the first urn $\mathcal{U}_1$ and put into the second urn $\mathcal{U}_2$. Then, the set of balls $\mathcal{S}_2$ in the second urn $\mathcal{U}_2$ is upsampled using a modified Pólya urn scheme. For that, in each upsampling step, a ball is drawn from $\mathcal{U}_2$. If either a red or black ball is drawn, a second ball of this type is added to $\mathcal{U}_2$. Otherwise, if a checkered ball is drawn from $\mathcal{U}_2$, a further ball is drawn from $\mathcal{U}_3$ and added to $\mathcal{U}_2$. Hence, either the number of red balls or the number of black balls in $\mathcal{U}_2$ grows after each upsampling step, while the number of checkered balls remains constant. This process is later on illustrated at an example (see Fig.4.1). The upsampling is continued for N_{up} steps.

By adding further colors to $\mathbb{X}$, the urn model can be extended to the multi-dimensional case ($W > 2$). In this case, the number of checkered balls in $\mathcal{U}_2$ has to be adjusted to the new W, and balls of the respective colors have to be added to $\mathcal{U}_1$ and $\mathcal{U}_3$, such that the balls in $\mathcal{U}_3$ are distributed according to the new prior. For an infinite number of upsampling steps, i.e. for $N_{\text{up}} \to \infty$, the Pólya urn scheme yields a Dirichlet distribution [Pou14]. Here, the likelihood of $\boldsymbol{p}_A$ follows $\text{Dir}(\boldsymbol{p}_A, \boldsymbol{r}_A, \boldsymbol{a}_A)$. This second order probability can be imagined as the likelihood that the probabilities of drawing a respective ball from $\mathcal{U}_1$ actually are given by $\boldsymbol{p}_A$. Moreover, $\boldsymbol{r}_A$ describes how many balls of the respective colors from $\mathbb{X}$ are in $\mathcal{S}_A$.

Since $\boldsymbol{r}_A$ is the measured number of respectively colored balls in $\mathcal{S}_A$, $\boldsymbol{r}_A$ reflects the available statistical evidence on the system. Hence, in terms of evidence theory, $\boldsymbol{r}_A$ referred to the total number of balls $N_2 = N_A + W$ in $\mathcal{U}_2$ forms the belief mass $\boldsymbol{b}_A$, while the fraction of checkered balls referred to N_2, is the statistical uncertainty u_a. This yields the SL opinion

$$\omega_A = (\boldsymbol{b}_A, u_A, \boldsymbol{a}_A), \quad \boldsymbol{b}_A = \frac{\boldsymbol{r}_A}{W + N_A}, \quad u_A = \frac{W}{W + N_A}. \tag{4.3}$$

Since the proposed upsampling process links the opinion ω_A with the Dirichlet distribution $\text{Dir}(\boldsymbol{p}_A, \boldsymbol{r}_A, \boldsymbol{a}_A)$ it realizes the equivalent mapping theorem, Theorem 3.1.

By replacing the checkered balls by balls drawn from $\mathcal{U}_3$ the SL opinion can be projected to a classical probability. Here, the checkered balls reflect the lack of knowledge and the balls

in $\mathcal{U}_3$ the prior knowledge. The expected number of respectively colored balls in $\mathcal{U}_2$ referred to the total number of balls in $\mathcal{U}_2$ then yields the projected probability of ω_A, which is

$$\boldsymbol{P}_{A,\perp} = \frac{\boldsymbol{r}_A}{N_A + W} + \frac{W\,\boldsymbol{a}_A}{N_A + W} = \boldsymbol{b}_A + u_A\,\boldsymbol{a}_A\,. \tag{4.4}$$

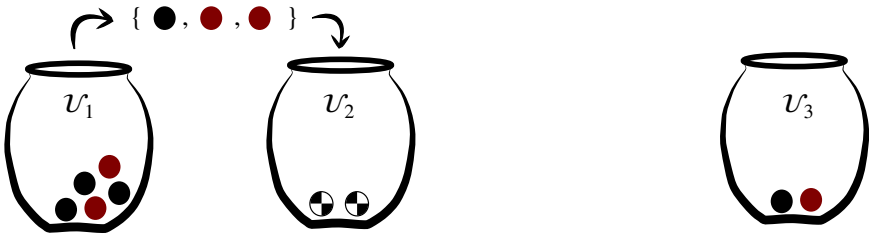
Figure 4.1 illustrates the urn model at the example of a binomial opinion over the domain $\mathbb{X} = \{\bullet, \bullet\}$, which is determined for the set of $N_A = 3$ balls $\mathcal{S}_A = \{\bullet, \bullet, \bullet\}$ drawn from $\mathcal{U}_1$. Without special prior knowledge, the prior $\boldsymbol{a}_A = [0.5,\ 0.5]^T$ is assumed. This corresponds to having the set of balls $\mathcal{S}_3 = \{\bullet, \bullet\}$ in the third urn. With the $W = 2$ checkered balls initially included in $\mathcal{U}_2$, the overall set of balls in the second urn is $\mathcal{S}_2 = \{\bullet, \bullet, \bullet, \textcircled{\bullet}, \textcircled{\bullet}\}$. From this, the belief mass $\boldsymbol{b}_A = [0.4,\ 0.2]^T$ and the statistical uncertainty $u_A = 0.4$ are calculated, which eventually yields the SL opinion $\omega_A = ([0.4,\ 0.2]^T,\ 0.4,\ [0.5,\ 0.5]^T)$. Then the upsampling is carried out according to the modified Pólya urn scheme. For $N_{\text{up}} \to \infty$ upsampling steps, the likelihood of $\hat{\boldsymbol{p}}_A$ converges to the Dirichlet distribution $\text{Dir}\left(\hat{\boldsymbol{p}}_A, [2, 1]^T, [0.5, 0.5]^T\right)$. Furthermore, the projected probability $\boldsymbol{P}_{A,\perp} = [0.6, 0.4]^T$ results.

At this point, the question arises, how knowledge on the system can be combined. For this, two sets of balls $\mathcal{S}_A$ and $\mathcal{S}_B$, consisting of N_A and N_B balls, respectively, are independently drawn from $\mathcal{U}_1$. The most intuitive way to combine $\mathcal{S}_A$ and $\mathcal{S}_B$ is to put both sets together in the second urn. However, since the priors $\boldsymbol{a}_A$ and $\boldsymbol{a}_B$ might differ for $\mathcal{S}_A$ and $\mathcal{S}_B$, $\boldsymbol{a}_A$ and $\boldsymbol{a}_B$ need to be combined, as well. An intuitive way of doing so is to calculate the weighted average with respect to the sample sizes N_A and N_B, i.e.,

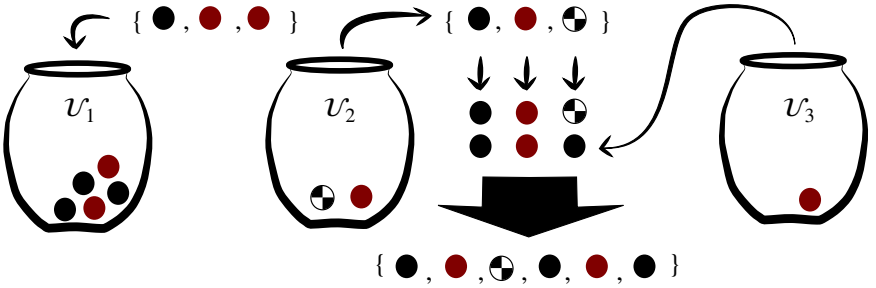
$$\boldsymbol{a}_{A\oplus B} = \frac{N_A\,\boldsymbol{a}_A + N_B\,\boldsymbol{a}_B}{N_A + N_B}\,. \tag{4.5}$$

As will be shown with Theorem 4.2, this leads to the CBF operator from SL.

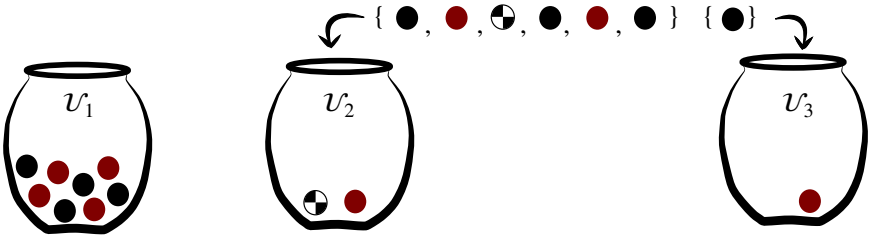
So far, the color of the balls drawn from the first urn have been evident, i.e., full observability has been assumed for the urn model. However, due to measurement noise, the assumption of full observability is questionable in many practical cases. Therefore, to represent partial observability, the urn model is extended by a fourth urn $\mathcal{U}_4$. This fourth urn is also filled with balls that have the colors from $\mathbb{X}$. Additionally $\mathcal{U}_4$ contains white balls, symbolized as $\circ$. Like for $\mathcal{U}_3$, only the relative fractions are relevant for $\mathcal{U}_4$. In the partial observable case, estimating $\boldsymbol{p}^*_{\mathcal{U}'1}$ is not possible, but has to be infered from balls drawn from $\mathcal{U}_4$ and $\mathcal{U}_1$. A set of such drawn balls $\tilde{\mathcal{S}}_A$ is referred to as observation. $\tilde{\mathcal{S}}_A$ is obtained by drawing N_A balls from $\mathcal{U}_4$. Whenever a white ball is drawn, a further ball is drawn from $\mathcal{U}_1$ and added to $\tilde{\mathcal{S}}_A$. Otherwise, the drawn ball is directly added to $\tilde{\mathcal{S}}_A$. The sampling process is hidden, so only $\tilde{\mathcal{S}}_A$ is known.



(a) Step 1: A set of balls S_A is drawn from $\mathcal{U}_1$ and put into $\mathcal{U}_2$, which is already filled with $W = 2$ checkered balls.



(b) Step 2: As part of drawing S_A is returned to $\mathcal{U}_1$. Furthermore, balls are drawn from $\mathcal{U}_2$. For each checkered ball a further ball is drawn from $\mathcal{U}_3$, while for each colored ball another one with the same color is added. The sample is upsampled according to the Pólya urn scheme.



(c) Step 3: The sets of balls are put back to $\mathcal{U}_2$ and $\mathcal{U}_3$, respectively, and the upsampling is continued with Step 2.

Figure 4.1: Basic urn model based on the Pólya urn scheme [EP23] illustrating SL opinions. The probability estimate $\hat{p}_A$ is obtained by repeating the upsampling Steps 2 and 3 N_{up} times (cf. [MB21]).

Figure 4.2 illustrates the draw scheme for partial observability by the example of drawing $N_A = 3$ balls $\{\circ, \bullet, \circ\}$ from $\mathcal{U}_4$. For the example, the extended set of colors in $\mathcal{U}_4$ is $\mathbb{X}_{\text{po}} = \{\bullet, \bullet, \circ\}$. One of the drawn balls is a colored one, i.e. $\bullet$, the other two balls are white balls $\circ$. Thus, two balls from $\mathcal{S}_A$ are added to $\tilde{\mathcal{S}}_A$, while the third ball is not. Here, the balls $\{\bullet, \bullet\}$ are drawn from $\mathcal{U}_1$, which yields $\tilde{\mathcal{S}}_A = \{\bullet, \bullet, \bullet\}$. The corresponding opinion is $\tilde{\omega}_A = (\tilde{\boldsymbol{b}}_A, \tilde{u}_A, \boldsymbol{a}_A)$ with $\tilde{\boldsymbol{b}}_A = [0.2, 0.4]^T$, $\tilde{u}_A = 0.4$, and $\boldsymbol{a}_A = [0.5, 0.5]^T$.

Apparently, the imperfect observation leads to an underestimation of the statistical uncertainty $\tilde{u}_A$ and an overestimation of the belief mass $\tilde{\boldsymbol{b}}_A$. This is because all balls in $\tilde{\mathcal{S}}_A$ are put into $\mathcal{U}_2$, while only the balls drawn from $\mathcal{U}_1$ represent statistical evidence. More specifically, the expected number of balls that are actually drawn from $\mathcal{U}_1$ is $N_{A,D} = p_{\text{white}}\, \tilde{N}_A$, where p_{white} denotes the probability of drawing a white ball from $\mathcal{U}_4$ and $\tilde{N}_A$ is the number of balls in $\tilde{\mathcal{S}}_A$. Thus, for the estimated statistical uncertainty to equal the expected value of the actual statistical uncertainty, $\tilde{\mathcal{S}}_A$ must be reduced to $N_{A,D}$ balls. As will be stated in Theorem 4.3, this can be realized in a probabilistic way by randomly selecting and removing balls from $\tilde{\mathcal{S}}_A$. In terms of evidence theory, this corresponds to multiplying the belief mass $\tilde{\boldsymbol{b}}_A$ by a discount probability p_d, i.e.

$$\boldsymbol{b}_{A,D} = p_d\, \tilde{\boldsymbol{b}}_A\,, \qquad u_{A,D} = 1 - p_d \sum_{i=1}^{W} \tilde{b}_{A,i} = 1 - p_d + p_d\, \tilde{u}_A\,, \tag{4.6}$$

where a specific choice of p_d reduces the expected number of balls in $\tilde{\mathcal{S}}_A$ to the expected number of balls drawn from $\mathcal{U}_1$. In terms of SL, (4.6) corresponds to the trust discounting operator (3.9).

Apart from merging statistically independent information, for which the CBF operator has been introduced, in many situations, statistically dependent information has to be merged. One particular example is multi-criterion reliability estimation, where the same data are evaluated with respect to different criteria. This corresponds to observing the same set of balls $\mathcal{S}_A$ drawn from $\mathcal{U}_1$ multiple times. For such statistically dependent information, Bayesian inference and thus the CBF operator cannot be used. Instead, a further type of fusion is applied, which increases the accuracy rather than reducing the statistical uncertainty.

For example, if $\mathcal{S}_A$ is drawn from $\mathcal{U}_1$, the observation process may produce the two observations $\tilde{\mathcal{S}}_A^A$ and $\tilde{\mathcal{S}}_A^B$ with the corresponding statistical evidence $\boldsymbol{r}_A^A$ and $\boldsymbol{r}_A^B$. Moreover, the priors $\boldsymbol{a}_A^A$ and $\boldsymbol{a}_A^B$ belong to the observations $\tilde{\mathcal{S}}_A^A$ and $\tilde{\mathcal{S}}_A^B$, respectively. The basic idea of merging two observations $\tilde{\mathcal{S}}_A^A$ and $\tilde{\mathcal{S}}_A^B$ is to compute the weighted average of the corresponding statistical evidence $\boldsymbol{r}_A^A$ and $\boldsymbol{r}_A^B$ on the one hand, and weighted average of the priors on the

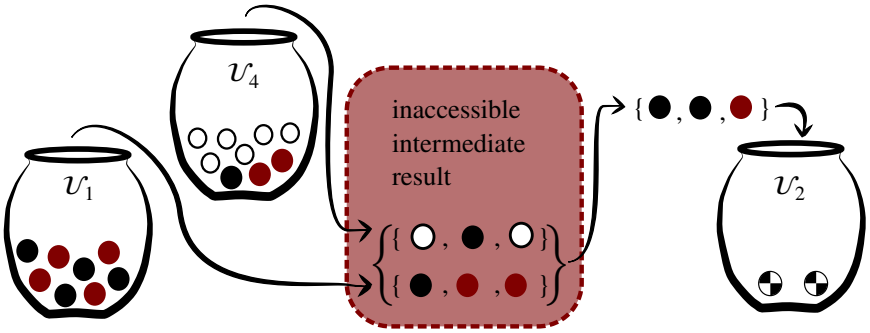


Figure 4.2: Extension of the urn model by a fourth urn. The drawing of $\tilde{S}_A$ is illustrated, which replaces Step 1 in Fig. 4.1. The other steps from Fig. 4.1 remain the same (cf. [MB21]).

other hand. With the importance weights c_A^A and c_A^B, this yields

$$\boldsymbol{r}_{A\tilde{\oplus}B} = \frac{c_A^A \boldsymbol{r}_A^A + c_A^B \boldsymbol{r}_A^B}{c_A^A + c_A^B}, \text{ and } \boldsymbol{a}_{A\tilde{\oplus}B} = \frac{c_A^A \boldsymbol{a}_A^A + c_A^B \boldsymbol{a}_A^B}{c_A^A + c_A^B}, \tag{4.7}$$

which corresponds to the weighted ABF operator. Since the number of balls in $\mathcal{U}_3$ is arbitrary, the fusion of the priors can be easily represented in the urn model. For that, $\boldsymbol{a}_{A\tilde{\oplus}B}$ is computed and $\mathcal{U}_3$ is filled with colored balls such that the probability of drawing the respective colors is $\boldsymbol{a}_{A\tilde{\oplus}B}$. In contrast, the number of balls in $\tilde{S}_A^A$ and $\tilde{S}_A^B$ is not arbitrary, but reflects the included amount of statistical evidence. Hence, obtaining $\boldsymbol{r}_{A\tilde{\oplus}B}$ is slightly more complex and requires three steps. First, the balls in $\tilde{S}_A^A$ and $\tilde{S}_A^B$ are replaced by multiple copies of them such that $\boldsymbol{r}_{A,\text{up}}^A$ and $\boldsymbol{r}_{A,\text{up}}^B$, which count the occurrences of the colors from $\mathbb{X}$ in the upscaled sets $\tilde{S}_{A,\text{up}}^A$ and $\tilde{S}_{A,\text{up}}^B$, are nonnegative integer numbers. This is,

$$\boldsymbol{r}_{A,\text{up}}^A = \tilde{\alpha} \frac{c_A^A}{c_A^A + c_A^B} \boldsymbol{r}_A^A \in \mathbb{N}_0^W, \qquad \boldsymbol{r}_{A,\text{up}}^B = \tilde{\alpha} \frac{c_A^B}{c_A^A + c_A^B} \boldsymbol{r}_A^B \in \mathbb{N}_0^W. \tag{4.8}$$

Then, the upscaled sets are joint, i.e., $S_{A,\text{up}}^{A\cup B} = \tilde{S}_{A,\text{up}}^A \cup \tilde{S}_{A,\text{up}}^B$. Finally, $S_{A,\text{up}}^{A\cup B}$ is scaled down again. This is realized in a probabilistic way, by including each ball from $S_{A,\text{up}}^{A\cup B}$ in the down scaled set $S_{A\tilde{\oplus}B}$ with the probability $p_{\text{down}} = 1/\tilde{\alpha}$. Hence, the expected number with which the colors from $\mathbb{X}$ occur in $S_{A\tilde{\oplus}B}$ is $\boldsymbol{r}_{A\tilde{\oplus}B}$.

The observation process adds noise to the observations, which can be mitigated but not fully removed by the trust discounting or the weighted ABF. Therefore, it is reasonable to estimate the inconsistency of the opinions first, and merge them only, if the inconsistency between

the opinions is sufficiently low. One way to think about inconsistency is the probability of observing a difference that cannot be explained by a lack of knowledge. In SL, this inconsistency measure is computed by the DC, as will be illustrated at the urn model.

For this, consider two observations $\tilde{S}_A$ and $\tilde{S}_B$ with their statistical evidence $\tilde{\boldsymbol{r}}_A$, $\tilde{\boldsymbol{r}}_B$, as well as their priors $\boldsymbol{a}_A$ and $\boldsymbol{a}_B$. This corresponds to two opinions $\tilde{\omega}_A$ and $\tilde{\omega}_B$. In contrast to the SL fusion operators described so far, the DC operator compares the information in $\tilde{S}_A$ and $\tilde{S}_B$ rather than merging it. Hence, one has to think of two instances of the urn model, where $\tilde{S}_A$ is put into the second urn of the first urn model instance $\mathcal{U}_{2,A}$ and $\tilde{S}_B$ is put into the second urn of the second urn model instance $\mathcal{U}_{2,B}$, respectively, and now contains colored and checkered balls. To compare the full knowledge included in the opinions, i.e., the observed and the prior knowledge, the projected probabilities are used for comparison. For that, the sets $\tilde{S}_{A,\perp}$ and $\tilde{S}_{B,\perp}$ are constructed such that they contain $\tilde{\boldsymbol{r}}_{A,\perp}$ and $\tilde{\boldsymbol{r}}_{B,\perp}$ balls of the respective colors from $\mathbb{X}$, where

$$\tilde{\boldsymbol{r}}_{A,\perp} = \tilde{\alpha}_A \tilde{\boldsymbol{r}}_A + \tilde{\alpha}_A W \boldsymbol{a}_A \in \mathbb{N}_0^W, \quad \tilde{\boldsymbol{r}}_{B,\perp} = \tilde{\alpha}_B \tilde{\boldsymbol{r}}_B + \tilde{\alpha}_B W \boldsymbol{a}_B \in \mathbb{N}_0^W, \tag{4.9a}$$

$$\tilde{N}_{A,\perp} = \sum_{i=1}^{W} \tilde{\boldsymbol{r}}_{A,\perp,i} = \sum_{i=1}^{W} \tilde{\boldsymbol{r}}_{B,\perp,i}, \; \tilde{\boldsymbol{P}}_{A,\perp} = \frac{\tilde{\boldsymbol{r}}_{A,\perp}}{\tilde{N}_{A,\perp}}, \; \tilde{\boldsymbol{P}}_{B,\perp} = \frac{\tilde{\boldsymbol{r}}_{B,\perp}}{\tilde{N}_{B,\perp}}. \tag{4.9b}$$

In essence, this means that $\tilde{S}_{A,\perp}$ and $\tilde{S}_{B,\perp}$ contain the same number of balls $\tilde{N}_{A,\perp}$ and are distributed according to their corresponding projected probabilities $\tilde{\boldsymbol{P}}_{A,\perp}$ and $\tilde{\boldsymbol{P}}_{B,\perp}$. Intuitively, if the two opinions are fully consistent $\tilde{S}_{A,\perp}$ and $\tilde{S}_{B,\perp}$ should be the same. Hence, to measure inconsistencies, only the differences between $\tilde{S}_{A,\perp}$ and $\tilde{S}_{B,\perp}$ provide new information. To reflect this, the balls contained in both sets, i.e., $S_{\perp,A\cap B} = \tilde{S}_{A,\perp} \cap \tilde{S}_{B,\perp}$, are each replaced by checkered balls and the modified sets are joined. This yields the set S_{PD}, which is comprised of $2\,\tilde{N}_{A,\perp}$ balls from which $2\,N_{\perp,A\cap B} = 2\,\mathrm{card}\{S_{\perp,A\cap B}\}$ balls are checkered balls. The set S_{PD} is put into a further urn $\mathcal{U}_{PD}$. Then, a ball is drawn from each of the three urn $\mathcal{U}_{2,A}$, $\mathcal{U}_{2,B}$, and $\mathcal{U}_{PD}$. Since the checkered balls represent a lack of knowledge, drawing three balls that are not checkered can be interpreted as observing a difference between $\tilde{\omega}_A$ and $\tilde{\omega}_B$ that cannot be explained by a lack of knowledge. It is shown in Theorem 4.5 that this probability is equivalent to the DC from SL.

4.2 Formalization and Mathematical Proof

In the next step, the urn model derived in the previous Section 4.1 is formalized and the key properties are proven. For this, first, the basic urn model is formalized in Definition 4.1.

Definition 4.1 (Basic Urn Model)
Consider an urn model of three urns, $\mathcal{U}_1$, $\mathcal{U}_2$, and $\mathcal{U}_3$. Let $\mathbb{X}$ be a finite set of colors and let ◓ describe an additional type of balls that has neither of the colors in $\mathbb{X}$, i.e. $\{◓\} \cap \mathbb{X} = \emptyset$.

Balls of the same type cannot be distinguished from each other. Then, draw a ball *is defined by the function* DrawBall($\mathcal{S}$)*, where* DrawBall($\mathcal{S}$) *randomly selects one ball from a set of balls $\mathcal{S}$, notes the type of the ball, puts the ball back, and outputs a ball of the same type. Correspondingly, the function* DrawBall($\mathcal{S}, k$) *returns a set $\mathcal{S}_{draw}$ of $k \in \mathbb{N}_0$ balls drawn from $\mathcal{S}$. Furthermore, let $\boldsymbol{r} \in \mathbb{N}_0^W$ denote the vector which for each color from $\mathbb{X}$ indicates how many balls from $\mathcal{S}$ have that color. Moreover, let* NoteColors($\mathcal{S}, \mathbb{X}$) *be the function that takes the set of balls $\mathcal{S}$ and the set of colors $\mathbb{X}$ as its input and returns the vector $\boldsymbol{r}$. If $\mathcal{S}$ contains balls with a color that is not included in $\mathbb{X}$, the respective balls are ignored. The first urn $\mathcal{U}_1$ is filled with the set of balls $\mathcal{S}_1$, where each ball has one of the colors from $\mathbb{X}$. Moreover, $\boldsymbol{p}^*_{\mathcal{U}_1}$ denotes the unknown probability vector of drawing a ball with a respective color included in $\mathbb{X}$ from $\mathcal{U}_1$ which has to be determined. The second urn $\mathcal{U}_2$ is initially filled with a set of $W = \text{card}\{\mathbb{X}\}$ checkered balls. The third urn $\mathcal{U}_3$ is filled with the set of balls $\mathcal{S}_3$, where each ball in $\mathcal{S}_3$ has one of the colors included in $\mathbb{X}$. $\mathcal{S}_3$ is chosen such that the probability vector $\boldsymbol{a}$ describes the probability of drawing a ball with a respective color included in $\mathbb{X}$ from $\mathcal{U}_3$. Here, $\boldsymbol{a}$ denotes a statistical prior of an SL opinion. This urn model is called the basic urn model.*

With the basic urn model from Definition 4.1, Theorem 4.1 introduces the drawing scheme for full observability, Algorithm 4.1. Algorithm 4.1 takes a set of balls $\mathcal{S}_A$ drawn from the first urn $\mathcal{U}_1$, the set of colors $\mathbb{X}$, and the prior distribution $\boldsymbol{a}_A$ corresponding to $\mathcal{S}_A$ as its input. From this, it computes W, the sample size N_A, and the statistical evidence $\boldsymbol{r}_A$ included in $\mathcal{S}_A$. With $\boldsymbol{r}_A$ and the prior $\boldsymbol{a}_A$, the algorithm computes the SL opinion ω_A and the projected probability $\boldsymbol{P}_{A,\perp}$. Then, the upsampling process starts, which is an iteration over N_{up} upsampling steps. In each upsampling step, a modified Pólya's urn scheme is applied. This means, balls are drawn from $\mathcal{U}_2$. Whenever a ball that has a color from $\mathbb{X}$ is drawn, a second ball of this color is added to $\mathcal{U}_2$. However, if a checkered ball is drawn, a further ball is drawn from $\mathcal{U}_3$ and the ball is added to $\mathcal{U}_2$. Finally, the occurrences of each color from $\mathbb{X}$ in $\mathcal{U}_2$ are referred to the total number of balls in $\mathcal{S}_2$, which yields the estimated probability $\hat{\boldsymbol{p}}_A$ of drawing the respective color from $\mathcal{U}_1$. The algorithm returns ω_A, $\hat{\boldsymbol{p}}_A$, and $\boldsymbol{P}_{A,\perp}$. It is shown that the algorithm returns an SL opinion ω_A and the projected probability $\boldsymbol{P}_{A,\perp}$, and furthermore realizes the bijective mapping theorem.

Theorem 4.1 (Urn Model Representation of SL Opinions)
Consider the basic urn model as defined in Definition 4.1. Let $\mathbb{X}$ be a domain corresponding to the set of $W = card\{\mathbb{X}\} \geq 2$ colors. Further let X be a random variable in $\mathbb{X}$, and let $\boldsymbol{a}_A$ be a prior of an SL opinion. Furthermore, let $\mathcal{S}_A \leftarrow$ DrawBall($\mathcal{S}_1, N_A$) be a set of N_A balls drawn from the first urn $\mathcal{U}_1$. Moreover, let N_{up} denote the number of iterations, i.e. the upampling steps, carried out during the upsampling process of Algorithm 4.1. Algorithm 4.1 is carried out with $\mathcal{S}_A$, $\boldsymbol{a}_A$, $\mathbb{X}$ as its inputs. Then, ω_A returned by Algorithm 4.1 is a SL opinion on X. Moreover, $\boldsymbol{P}_{A,\perp}$ is the projected probability of ω_A, which is furthermore the minimum mean square error (MMSE) estimate of $\boldsymbol{p}^*_{\mathcal{U}_1}$. $\boldsymbol{P}_{A,\perp}$ also is equal to the expected probability that a ball randomly picked from the set $\mathcal{S}_{A,\perp} \leftarrow \mathcal{S}_A \cup$ DrawBall($\mathcal{S}_3, W$) has a

Algorithm 4.1 Drawing Scheme for Full Observability

Input: Set of balls $\mathcal{S}_A$ drawn from the first urn, set of colors $\mathbb{X}$, number of upsampling steps N_{up}, and prior distribution $\boldsymbol{a}_A$

Output: Opinion ω_A, upsampled probability estimate $\hat{\boldsymbol{p}}_A$, and projected probability $\boldsymbol{P}_{A,\perp}$

$W \leftarrow \text{card}\{\mathbb{X}\},\ N_A \leftarrow \text{card}\{\mathcal{S}_A\}$
$\boldsymbol{r}_A \leftarrow \text{NoteColors}\ (\mathcal{S}_A, \mathbb{X})$
$\omega_A \leftarrow \left(\boldsymbol{b}_A = \frac{1}{N_A+W}\boldsymbol{r}_A, u_A = \frac{W}{N_A+W}, \boldsymbol{a}_A\right)$
$\boldsymbol{P}_{A,\perp} \leftarrow \frac{1}{N_A+W}\left(\boldsymbol{r}_A + W\,\boldsymbol{a}_A\right)$
$\mathcal{S}_2 \leftarrow \mathcal{S}_2 \cup \mathcal{S}_A$ *% remember: $\mathcal{S}_2$ contains W checkered balls*
for $k = 1$ **to** N_{up} **do**
  $\mathcal{S}_{\text{draw}} \leftarrow \text{DrawBall}(\mathcal{S}_2)$
  if $\mathcal{S}_{\text{draw}} \cap \{\text{🌍}\} = \emptyset$ **then**
    $\mathcal{S}_2 \leftarrow \mathcal{S}_2 \cup \mathcal{S}_{\text{Draw}}$ *% Apply Pólya's urn scheme*
  else
    $\mathcal{S}_{\text{draw}} \leftarrow \text{DrawBall}(\mathcal{S}_3)$
    $\mathcal{S}_2 \leftarrow \mathcal{S}_2 \cup \mathcal{S}_{\text{Draw}}$ *% Apply Pólya's urn scheme*
  end if
end for
} Upsampling process
$\hat{\boldsymbol{p}}_A \leftarrow \text{NoteColors}\ (\mathcal{S}_2, \mathbb{X})$
$\hat{\boldsymbol{p}}_A \leftarrow \frac{1}{W+\sum_{i=1}^{W}\hat{\boldsymbol{p}}_{A,i}}\hat{\boldsymbol{p}}_A$
return $\omega_A, \hat{\boldsymbol{p}}_A, \boldsymbol{P}_{A,\perp}$

specific color from $\mathbb{X}$. For $N_{up} \to \infty$, $\hat{\boldsymbol{p}}_A$ obtained from Algorithm 4.1 follows the Dirichlet distribution $Dir(\hat{\boldsymbol{p}}_A, \boldsymbol{r}_A, \boldsymbol{a}_A)$. Hence, the upsampling step of Algorithm 4.1 implements the bijective mapping between ω_A and its corresponding Dirichlet distribution $\text{Dir}(\hat{\boldsymbol{p}}_A, \boldsymbol{r}_A, \boldsymbol{a}_A)$.

Proof. According to Definition 3.1, the elements of ω_A have to fulfill the following conditions:

$$\boldsymbol{b}_A : \mathbb{X} \mapsto [0,1]^W , \tag{4.10a}$$

$$u_A : u_A + \sum_{i=1}^{W} b_{A,i} = 1 , \tag{4.10b}$$

$$\boldsymbol{a}_A : \mathbb{X} \mapsto [0,1]^W \wedge \sum_{i=1}^{W} a_{A,i} = 1 . \tag{4.10c}$$

By interpreting the events of drawing a specifically colored ball from the first urn as elements of $\mathbb{X}$, $\boldsymbol{b}_A$ calculated by Algorithm 4.1 is the result of a mapping $\mathbb{X} \mapsto [0,1]^W$. Particularly,

it follows directly from the definition of NoteColors($\mathcal{S}, \mathbb{X}$) that $r_{A,i} \geq 0$, $i = 1, \ldots, W$, and $\sum_{i=1}^{W} r_{A,i} = N_A$. Hence, the values of $b_{A,i}$ are always between zero and one and condition (4.10a) is fulfilled. Moreover, condition (4.10b) can be rephrased as

$$u_A + \sum_{i=1}^{W} b_{A,j} = \frac{W}{W + N_A} + \frac{\sum_{i=1}^{W} r_{A,i}}{W + N_A} = \frac{W + N_A}{W + N_A} = 1\,. \tag{4.11}$$

Hence, (4.10b) is also fulfilled for the proposed urn model. Since $\boldsymbol{a}_A$ was defined to be a SL prior, condition (4.10c) holds by definition. Therefore, all three conditions hold and ω_A returned by Algorithm 4.1 is a SL opinion.

During the upsampling process in Algorithm 4.1, the number of colored balls in $\mathcal{U}_2$ increases by one per step, while the number of checkered balls remains constant. Therefore, the probability of drawing a checkered ball vanishes in the limit of infinite steps

$$\tilde{u}_A = \frac{W}{W + N_A + N_{\text{up}}} \overset{N_{\text{up}} \to \infty}{\longrightarrow} 0\,. \tag{4.12}$$

Furthermore, for $N_{\text{up}} \to \infty$, the law of large numbers [BSMM07, p. 764] applies to the balls drawn from $\mathcal{U}_3$. Hence, in average $W\ \boldsymbol{a}_A$ balls of the respective colors are added to $\mathcal{U}_2$ in each step. Moreover, it is proven in [Pou14] that for $N_{\text{up}} \to \infty$ the order of draws does not influence the result obtained by the Pólya urn scheme.

Accordingly, the upsampling process of Algorithm 4.1 in the limit $N_{\text{up}} \to \infty$ yields the same $\boldsymbol{p}_A$ as if the Pólya urn scheme was applied to an urn filled with $\tilde{\boldsymbol{r}}_A = \boldsymbol{r}_A + W\ \boldsymbol{a}_A$ balls of the respective colors from $\mathbb{X}$. As shown in [Pou14], $\boldsymbol{p}_A$ obtained from the Pólya urn scheme follows a Dirichlet distribution, which for the particular case is equivalent to

$$\begin{aligned} \operatorname{Dir}(\hat{\boldsymbol{p}}_A, \tilde{\boldsymbol{r}}_A) &= \frac{\Gamma\left(\sum_{i=1}^{W} \tilde{r}_{A,i}\right)}{\prod_{i=1}^{W} \Gamma \tilde{r}_{A,i}} \prod_{i=1}^{W} \hat{p}_{A,i}^{\tilde{r}_{A,i}-1} = \frac{\Gamma\left(\sum_{i=1}^{W} (r_{A,i} + a_{A,i} W)\right)}{\prod_{i=1}^{W} \Gamma(r_{A,i} + a_{A,i} W)} \prod_{i=1}^{W} \hat{p}_{A,i}^{r_{A,i} + a_{A,i} W - 1} \\ &= \operatorname{Dir}(\hat{\boldsymbol{p}}_A, \boldsymbol{r}_A, \boldsymbol{a}_A)\,. \end{aligned} \tag{4.13}$$

Moreover, Algorithm 4.1 defines $\boldsymbol{r}_A$, $\boldsymbol{b}_A$, and u_A such that

$$N_A = \sum_{i=1}^{W} r_{A,i},\ \boldsymbol{b}_A = \frac{\boldsymbol{r}_A}{N_A + W},\ u_A = \frac{W}{N_A + W} \iff \boldsymbol{r}_A = \boldsymbol{b}_A (N_A + W) = \boldsymbol{b}_A \frac{W}{u_A} \tag{4.14}$$

is fulfilled. Hence, for $N_{\text{up}} \to \infty$, the upsampling process in Algorithm 4.1 maps the opinion ω_A to the Dirichlet distribution $\operatorname{Dir}(\hat{\boldsymbol{p}}_A, \boldsymbol{r}_A, \boldsymbol{a}_A)$ realizing the bijective mapping (3.5).

The probability $\boldsymbol{P}_{A,\perp}$ computed by Algorithm 4.1 is the projected probability for ω_A, since

$$\boldsymbol{P}_{A,\perp} = \frac{\boldsymbol{r}_A}{N_A + W} + \frac{W}{N_A + W}\,\boldsymbol{a}_A \overset{(3.5)}{=} \boldsymbol{b}_A + u_A\,\boldsymbol{a}_A\,. \tag{4.15}$$

According to [Pis14, pp. 526-528], the expectation of an a posteriori distribution in general yields the MMSE estimate. To obtain such an a posteriori distribution, Bayesian inference has to be applied. Therefore, Algorithm 4.1 is first carried out for the special case $\mathcal{S}'_A = \emptyset$, which yields the so-called vacuous opinion $\omega_\emptyset$ and the corresponding Dirichlet distribution $\mathrm{Dir}(\hat{\boldsymbol{p}}_\emptyset, \boldsymbol{0}, \boldsymbol{a}_A)$. Second, the distribution of drawing $\mathcal{S}_A$ from $\mathcal{U}_1$ is determined. For this, $\boldsymbol{r}_A$ is interpreted as the random variable of drawing the respectively colored balls in $\mathcal{S}_A$ from $\mathcal{U}_1$. Therefore, $\boldsymbol{r}_A$ follows the multinomial distribution $\mathrm{Mult}\left(\tilde{\boldsymbol{r}}_A \middle| \boldsymbol{p}^*_{\mathcal{U}_A}, N_A\right)$. This is is denoted by $\sim$, i.e., $\boldsymbol{r}_A \sim \mathrm{Mult}\left(\tilde{\boldsymbol{r}}_A \middle| \boldsymbol{p}^*_{\mathcal{U}_A}, N_A\right)$. Using Bayesian inference, the a posterior distribution

$$\boldsymbol{p}_{A,\mathrm{post}}(\hat{\boldsymbol{p}}|\boldsymbol{r}_A, \boldsymbol{a}_A) = \frac{\mathrm{Mult}\left(\tilde{\boldsymbol{r}}_A \middle| \hat{\boldsymbol{p}}, N_A\right)\,\mathrm{Dir}(\hat{\boldsymbol{p}}, \boldsymbol{0}, \boldsymbol{a}_A)}{\int_{-\infty}^{\infty} \mathrm{Mult}\left(\tilde{\boldsymbol{r}}_A \middle| \boldsymbol{p}', N_A\right)\,\mathrm{Dir}(\boldsymbol{p}', \boldsymbol{0}, \boldsymbol{a}_A)\mathrm{d}\boldsymbol{p}'} \tag{4.16}$$

is computed. This can be simplified to $\boldsymbol{p}_{A,\mathrm{post}}(\hat{\boldsymbol{p}}|\boldsymbol{r}_A, \boldsymbol{a}_A) = \mathrm{Dir}(\hat{\boldsymbol{p}}, \boldsymbol{0} + \boldsymbol{r}_A, \boldsymbol{a}_A)$ using the results from [Bis06, p. 77]. Hence, adding the set of drawn balls $\boldsymbol{S}_A$ to the vacuous opinion $\omega_\emptyset$ yields the Dirichlet distribution that corresponds to ω_A. This Dirichlet distribution $\mathrm{Dir}(\hat{\boldsymbol{p}}, \boldsymbol{0} + \boldsymbol{r}_A, \boldsymbol{a}_A)$ can be interpreted as an a posteriori distribution, which has the expectation $\boldsymbol{P}_{A,\perp} = \mathbb{E}\left\{\mathrm{Dir}(\hat{\boldsymbol{p}}, \boldsymbol{0} + \boldsymbol{r}_A, \boldsymbol{a}_A)\right\}$. Hence, $\boldsymbol{P}_{A,\perp}$ is the MMSE estimate of $\boldsymbol{p}^*_{\mathcal{U}_1}$.

Finally, it is shown that $\boldsymbol{P}_{A,\perp}$ also represents the expected probability that a ball randomly picked from the set $\mathcal{S}_{A,\perp} \leftarrow \mathcal{S}_A \cup \mathrm{DrawBall}(\mathcal{S}_3, W)$ has a specific color from $\mathbb{X}$. Intuitively, the set $\mathcal{S}_{A,\perp}$ is obtained by replacing the W checkered balls in $\mathcal{U}_2$ with balls drawn from $\mathcal{U}_3$. For this, let $\tilde{\boldsymbol{r}}_W$ be a random variable describing the number of balls per color after drawing W balls from $\mathcal{U}_3$. Then, $\tilde{\boldsymbol{r}}_W$ follows the multinomial distribution $\mathrm{Mult}\left(\tilde{\boldsymbol{r}}_W \middle| \boldsymbol{a}_A, W\right)$. Since $\tilde{\mathcal{S}}_2$ contains $\mathrm{card}\{\tilde{\mathcal{S}}_2\} = N_A + W$ balls in any case, a ball of a specific color is drawn with

$$\tilde{\boldsymbol{P}}_{A,\perp} = \frac{\boldsymbol{r}_A + \tilde{\boldsymbol{r}}_W}{N_A + W} \sim \frac{\boldsymbol{r}_A + \mathrm{Mult}\left(\tilde{\boldsymbol{r}}_W \middle| \boldsymbol{a}_A, W\right)}{N_A + W}\,. \tag{4.17}$$

Hence, the expectation $\mathbb{E}\{\cdot\}$ of drawing a ball of this color is

$$\mathbb{E}\{\,\tilde{\boldsymbol{P}}_{A,\perp}\,\} = \mathbb{E}\left\{\frac{\boldsymbol{r}_A}{N_A + W}\right\} + \mathbb{E}\left\{\frac{\tilde{\boldsymbol{r}}_W}{N_A + W}\right\} = \frac{\boldsymbol{r}_A + W\,\boldsymbol{a}_A}{N_A + W}\,. \tag{4.18}$$

Thus, the expectation of drawing a ball of a respective color from $\mathcal{S}_2$ is equal to $\boldsymbol{P}_{A,\perp}$. □

Apparently, joining two sets of balls S_A and S_B drawn independently from $\mathcal{U}_1$ corresponds to merging the corresponding opinions ω_A and ω_B using Bayesian inference. However, in contrast to the proof of Theorem 4.1, where a vacuous opinion was extended with statistical evidence, for the fusion of two opinions ω_A and ω_B the priors $\boldsymbol{a}_A$ and $\boldsymbol{a}_B$ might differ. Hence, the priors $\boldsymbol{a}_A$ and $\boldsymbol{a}_B$ must be weighted when merging ω_A and ω_B. Intuitively, the weight of a prior should increase with the sample size, since a bigger sample size corresponds to having more information. Therefore, the priors are weighted with the sample sizes N_A and N_B. This leads to the urn model representation of the CBF operator, as stated in Theorem 4.2.

Theorem 4.2 (Urn Model Representation of the CBF Operator)
Consider the urn model from Definition 4.1. Let S_A and S_B be the sets of balls drawn from $\mathcal{U}_1$. Moreover, let $\boldsymbol{a}_A$ and $\boldsymbol{a}_B$ be priors such that S_A, $\boldsymbol{a}_A$, and S_B, $\boldsymbol{a}_B$ correspond to the opinions ω_A and ω_B, respectively. Then, the result of the CBF operator $\omega_C = \omega_A \oplus \omega_B$ is obtained by joining S_A and S_B, i.e. $S_C = S_A \cup S_B$, and weighting the priors $\boldsymbol{a}_A$ and $\boldsymbol{a}_B$ according to the sample sizes N_A and N_B. Furthermore, the CBF operator corresponds to Bayesian inference.

Proof. Since Definition 4.1 only specifies the relative occurrences of the respectively colored balls from the third urn $\mathcal{U}_3$, but not the absolute number, the set S_3 can be scaled arbitrarily with the factor $\alpha \in \mathbb{R}$. The only condition is that the number of occurrences of each color from $\mathbb{X}$ in S_3 remains a natural number. Thus, $S_{C,3}$ can be obtained by filling $\mathcal{U}_3$ with

$$\boldsymbol{r}_{C,3} = \alpha\, N_A\, \boldsymbol{a}_A + \alpha\, N_B\, \boldsymbol{a}_B \tag{4.19}$$

balls. The factor α is chosen such that $\boldsymbol{r}_{C,3} \in \mathbb{N}^W$ holds. The probability $\boldsymbol{a}_C$ of drawing a specifically colored ball from $\mathcal{U}_3$ is

$$\begin{aligned}\boldsymbol{a}_C &= \frac{\alpha(N_A\,\boldsymbol{a}_A + N_B\,\boldsymbol{a}_B)}{\alpha(N_A + N_B)} \overset{(3.5)}{=} \frac{\boldsymbol{a}_A \frac{W}{u_A}(1-u_A) + \boldsymbol{a}_B \frac{W}{u_B}(1-u_B)}{\frac{W}{u_A}(1-u_A) + \frac{W}{u_B}(1-u_B)} \\ &= \frac{\boldsymbol{a}_A\, u_B + \boldsymbol{a}_B\, u_A + (\boldsymbol{a}_A + \boldsymbol{a}_B)\, u_A\, u_B}{u_A + u_B - 2u_A\, u_B} = \boldsymbol{a}_{A\oplus B}\,. \end{aligned} \tag{4.20}$$

The uncertainty of ω_C can be calculated with the number of balls in the joint S_C, i.e. $N_C = \mathrm{card}\{S_C\} = N_A + N_B$. With this, one obtains

$$u_C = \frac{W}{W + N_A + N_B} \overset{(3.5)}{=} \frac{W}{W + \frac{W}{u_A}(1-u_A) + \frac{W}{u_B}(1-u_B)} = \frac{u_A\, u_B}{u_A + u_B - u_A\, u_B}\,. \tag{4.21}$$

As can be seen from (3.7), u_C is equal to $u_{A\oplus B}$. In the next step, the belief mass of ω_C is

derived. Here, $\mathcal{S}_C = \mathcal{S}_A \cup \mathcal{S}_B$ implies $\boldsymbol{r}_C = \boldsymbol{r}_A + \boldsymbol{r}_B$. Thus, it can be concluded that

$$\boldsymbol{b}_C = \frac{u_C}{W}(\boldsymbol{r}_A + \boldsymbol{r}_B) \overset{(3.5)}{=} \frac{u_C}{W}\left(\frac{W\boldsymbol{b}_A}{u_A} + \frac{W\boldsymbol{b}_B}{u_B}\right) = \frac{\boldsymbol{b}_A u_B + \boldsymbol{b}_B u_A}{u_A + u_B - u_A u_B}\,. \tag{4.22}$$

Therefore, according to (4.22), $\boldsymbol{b}_C$ is equal to $\boldsymbol{b}_{A\oplus B}$. In analogy to the proof of Theorem 4.1, the Dirichlet distribution corresponding to ω_C can be expressed in terms of the Dirichlet and the multinomial distribution, which corresponds to ω_A and ω_B, respectively. This yields

$$\mathrm{Dir}(\hat{\boldsymbol{p}}, \boldsymbol{r}_C, \boldsymbol{a}_C) = \frac{\mathrm{Mult}\left(\boldsymbol{r}_B \middle| \hat{\boldsymbol{p}}, N_B\right)\,\mathrm{Dir}(\hat{\boldsymbol{p}}, \boldsymbol{r}_A, \boldsymbol{a}_C)}{\int_{-\infty}^{\infty}\mathrm{Mult}\left(\boldsymbol{r}_B \middle| \boldsymbol{p}', N_B\right)\,\mathrm{Dir}(\boldsymbol{p}', \boldsymbol{r}_A, \boldsymbol{a}_C)\mathrm{d}\boldsymbol{p}'}\,. \tag{4.23}$$

Thus, it is proven that the CBF operator from SL corresponds to Bayesian inference. □

In the next step, the urn model of Definition 4.1 is extended to the partial observable case, which also accounts for imperfect observation. For this, the draw scheme for partial observability, Algorithm 4.2, is defined.

Definition 4.2 (Urn Model Representation of Subjective Logic for Partial Observability) *Consider the urn model in Definition 4.1. Let $\mathbb{X}_{po}$ be an extended set of colors which additionally contains white balls ∘, i.e. $\mathbb{X}_{po} = \mathbb{X} \cup \{\circ\}$, $\{\circ\} \cap \mathbb{X} = \emptyset$. Furthermore, let $\mathcal{U}_4$ be a fourth urn, filled with a set of balls $\mathcal{S}_4$. Let each of these balls be of one of the colors from $\mathbb{X}_{po}$. Moreover, let $\tilde{\mathcal{S}}_A$ be a set of balls gathered by drawing balls according to Algorithm 4.2 for $\mathcal{S}_1$, $\mathcal{S}_4$, and the sample size N_A. $\tilde{\mathcal{S}}_A$ is called the observation of $\mathcal{S}_A$. Finally, let $\tilde{\omega}_A$ be an SL opinion obtained by carrying out Algorithm 4.1 for $\tilde{\mathcal{S}}_A$, $\mathbb{X}$, and $\boldsymbol{a}_A$. This urn model is called urn model representation of SL for partial observability.*

Algorithm 4.2 generates an observation, by drawing balls from an additional, fourth urn. Whenever a white ball is drawn, a further ball is drawn from the first urn and added to the observation. If, however, a colored ball is drawn, this ball is added to the observation, which reflects noise. Thus, it becomes apparent that carrying out Algorithm 4.1 for the observation $\tilde{\mathcal{S}}_A$ results in an underestimation of the statistical uncertainty. This can be corrected by the trust discounting operator from SL. It is stated in Theorem 4.3, that the trust discounting operator can be represented in the urn model by Algorithm 4.3. The algorithm draws for each ball in $\tilde{\mathcal{S}}_A$ a ball from the fourth urn $\mathcal{U}_4$. If the ball drawn from $\mathcal{U}_4$ is not white, the corresponding ball in $\tilde{\mathcal{S}}_A$ is not counted in $\hat{\boldsymbol{r}}_{A,D}$. Hence, the statistical evidence $\hat{\boldsymbol{r}}_{A,D}$ estimated by Algorithm 4.3 is discounted with respect to $\tilde{\boldsymbol{r}}_A \leftarrow \mathrm{NoteColors}(\tilde{\mathcal{S}}_A, \mathbb{X})$. Furthermore, $\hat{u}_{A,D}$ and $\hat{\boldsymbol{b}}_{A,D}$ are corrected.

Algorithm 4.2 Draw Scheme for Partial Observability

Input: Sets of balls $\mathcal{S}_1$, $\mathcal{S}_4$ in the first and fourth urn, and sample size N_A
Output: Observed set of balls $\tilde{\mathcal{S}}_A$

```
1: S̃_A ← ∅, S_A ← ∅
2: for k = 1 to N_A do
3:    S_noise ← DrawBall(S_4)
4:    S_draw ← DrawBall(S_1)
5:    S_A ← S_A ∪ S_draw
6:    if S_noise = {∘} then
7:       S̃_A ← S̃_A ∪ S_draw
8:    else
9:       S̃_A ← S̃_A ∪ S_noise
10:   end if
11: end for
12: return S̃_A
```

Theorem 4.3 (Urn Model Representation of the Trust Discounting Operator)
Consider the extended urn model in Definition 4.2. Let p_{white} be the probability of drawing a white ball $\circ$ from $\mathcal{U}_4$. Moreover, let $\mathcal{S}_{A,D} = \mathcal{S}_A \cap \tilde{\mathcal{S}}_A$ denote the subset of $\tilde{\mathcal{S}}_A$, which contains the statistical evidence. Let $\omega^*_{A,D} = (\boldsymbol{b}^*_{A,D}, u^*_{A,D}, \boldsymbol{a}_A)$ be the corresponding opinion. Furthermore, let $\hat{\omega}_{A,D}$ be the result of Algorithm 4.3 for the input $\tilde{\mathcal{S}}_A$. Then, the observed uncertainty $\tilde{u}_A$ is related to $u^*_{A,D}$ by

$$\mathbb{E}\left\{u^*_{A,D}\right\} = \frac{\tilde{u}_A}{1 + p_{white} - \tilde{u}_A\, p_{white}}\,. \tag{4.24}$$

Moreover, in expectation, $\hat{\omega}_{A,D}$ is equivalent to the result of the trust discounting operator (3.9) *and the uncertainty $\mathbb{E}\{\hat{u}_{A,D}\} = u^*_{A,D}$ represents the statistical evidence in $\tilde{\mathcal{S}}_A$.*

Proof. Since $\tilde{\mathcal{S}}_A$ is generated with Algorithm 4.2, the number of balls in $\tilde{\mathcal{S}}_A$, which are drawn from $\mathcal{U}_1$ follows a multinomial distribution with the expectation $\mathbb{E}\{N_{A,D}\} = p_{\text{white}}\, N_A$. Therefore, the expected statistical uncertainty in $\tilde{\mathcal{S}}_A$ is

$$\mathbb{E}\left\{u_{A,D^*}\right\} = \frac{W}{W + p_{\text{white}}\, N_A} \overset{(3.5)}{=} \frac{\tilde{u}_A}{1 + p_{\text{white}} - \tilde{u}_A\, p_{\text{white}}}\,. \tag{4.25}$$

In Algorithm 4.3, the balls in $\tilde{\mathcal{S}}_A$ are counted only if a white ball is drawn for them. Thus,

Algorithm 4.3 Urn Model Representation of the Trust Discounting Operator

Input: Sets of drawn balls $\tilde{S}_A$, set of balls in fourth urn S_4, and set of colors $\mathbb{X}$
Output: Corrected opinion $\hat{\omega}_{A,D}$

$\hat{\boldsymbol{r}}_{A,D} \leftarrow \mathbf{0}, \quad \hat{N}_{A,D} \leftarrow 0$
$\tilde{\boldsymbol{r}}_A \leftarrow \text{NoteColors}(\tilde{S}_A, \mathbb{X})$
for $i = 1$ **to** W **do**
  for $j = 1$ **to** $\tilde{r}_{A,i}$ **do**
    $S_{\text{noise}} \leftarrow \text{DrawBall}(S_4)$
    if $S_{\text{noise}} = \{\circ\}$ **then**
      $\hat{\boldsymbol{r}}_{A,D,i} \leftarrow \hat{\boldsymbol{r}}_{A,D,i} + 1$
      $\hat{N}_{A,D} \leftarrow \hat{N}_{A,D} + 1$
    end if
  end for
end for
$\hat{u}_{A,D} \leftarrow \frac{W}{\hat{N}_{A,D}+W}$
$\hat{\boldsymbol{b}}_{A,D} \leftarrow \frac{\hat{\boldsymbol{r}}_{A,D}}{W+\hat{N}_{A,D}}$
return $\hat{\omega}_{A,D} \leftarrow (\hat{\boldsymbol{b}}_{A,D}, \hat{u}_{A,D}, \boldsymbol{a}_A)$

the expectations of $\hat{\boldsymbol{r}}_{A,D}$, $\hat{\boldsymbol{b}}_{A,D}$, and $\hat{u}_{A,D}$ are

$$\mathbb{E}\left\{\hat{\boldsymbol{r}}_{A,D}\right\} = p_{\text{white}}\,\tilde{\boldsymbol{r}}_A,\ \mathbb{E}\left\{\hat{\boldsymbol{b}}_{A,D}\right\} = \frac{p_{\text{white}}\,\tilde{\boldsymbol{r}}_A}{p_{\text{white}}\,N_A + W},\ \mathbb{E}\{\hat{u}_{A,D}\} = \frac{W}{p_{\text{white}}\,N_A + W}\,. \quad (4.26)$$

Hence, in expectation, the uncertainty $\hat{u}_{A,D}$ reflects the statistical evidence in $\tilde{S}_A$.

In the next step, the opinion $\hat{\omega}_{A,D}$ obtained from Algorithm 4.3 is compared to the result of the trust discounting operator (3.9). It can be seen from Algorithm 4.3 that the calculation of $\hat{\boldsymbol{b}}_{A,D}$ slightly differs from the calculation of the belief mass in (3.9). In contrast to (3.9), the belief mass is calculated with $\hat{N}_{A,D}$ instead of N_A. This is to conserve the property $\mathbb{E}\{\hat{u}_{A,D}\} = \mathbb{E}\{u_{A,D}\}$. Hence, $p_d \neq p_{\text{white}}$ has to be adjusted to

$$p_d = \mathbb{E}\left\{\frac{\tilde{\boldsymbol{b}}_A}{\hat{\boldsymbol{b}}_{A,D}}\right\} = 1 - \frac{(1-p_{\text{white}})\tilde{u}_A}{(1-p_{\text{white}})\tilde{u}_A + p_{\text{white}}}\,. \quad (4.27)$$

For this p_d, the belief masses $\mathbb{E}\{\hat{\boldsymbol{b}}_{A,D}\} = \boldsymbol{b}_{A,D}$ and the uncertainty fit in expectation, i.e.,

$$\mathbb{E}\left\{\hat{u}_{A,D}\right\} = 1 - \frac{p_d\,\sum_{i=1}^{W} r_{A,i}}{N_A + W} = \frac{W}{p_{\text{white}}\,N_A + W} = u_{A,D}\,. \quad (4.28)$$

In conclusion, if p_d is chosen according to (4.27), Algorithm 4.3 in expectation yields the same result as the trust discounting operator. □

Note that in most practical cases, the noise rate is low, so p_{white} is close to one. Furthermore, in many situations, the statistical uncertainty is small. Thus, $p_d \approx p_{\text{white}}$ holds and Algorithm 4.3 approximates the trust discounting operator even without the correction (4.27). Another option would be to adjust the definition of the operator. This way, trust discounting would become more intuitive.

Besides trust discounting, the observation accuracy can be improved by averaging multiple observations of the same set S_A drawn from the first urn. This leads to the weighted ABF operator and its representation in terms of the urn model.

Theorem 4.4 (Urn Model Representation of the ABF Operators)
Consider the urn model for partial observability in Definition 4.2. Let $\tilde{S}_A^A$ and $\tilde{S}_A^B$ be observations of the same set of balls S_A, which is drawn from $\mathcal{U}_1$. Moreover, let S_3^A and S_3^B be the corresponding prior sets for $\mathcal{U}_3$. Furthermore, let $c_A^A, c_A^B \in \mathbb{R}^+$ be importance weights reflecting the observation quality of $\tilde{S}_A^A$ and $\tilde{S}_A^B$, and let $\alpha, \tilde{\alpha} \geq 1$ be scaling factors. In addition, let $\tilde{S}_{A,up}^A$ and $\tilde{S}_{A,up}^B$ denote upscaled sets of $\tilde{S}_A^A$ and $\tilde{S}_A^B$, where each ball in $\tilde{S}_A^A$ is replaced by $\alpha\, c_A^A \in \mathbb{N}$ copies, and each ball in $\tilde{S}_A^B$ is replaced by $\alpha\, c_A^B \in \mathbb{N}$ copies of itself. Analogously, let $S_{3,up}^A$ and $S_{3,up}^B$ be the sets obtained by upscaling S_3^A and S_3^B with

$$\frac{\tilde{\alpha} c_A^A}{\text{card}\{S_3^A\}} \in \mathbb{N}, \quad \frac{\tilde{\alpha} c_A^B}{\text{card}\{S_3^B\}} \in \mathbb{N}. \tag{4.29}$$

Furthermore, let $\tilde{S}_{A,up}^{A\cup B} = \tilde{S}_{A,up}^A \cup \tilde{S}_{A,up}^B$ and $S_3^{A\oplus B} = S_{3,up}^A \cup S_{3,up}^B$ denote the joined sets of the upscaled observations and the upscaled prior sets, respectively. Moreover, let $S_{A\oplus B}$ be the set yielded by randomly drawing balls from $\tilde{S}_{A,up}^{A\cup B}$, where each ball is drawn with the probability $p_{down} = \alpha^{-1}(c_A^A + c_A^B)^{-1}$. Finally, let $\omega_{A\oplus B}$ denote the opinion yielded by Algorithm 4.1 for the set $S_{A\oplus B}$, and the prior set $S_3^{A\oplus B}$. Then, the expectation of $\omega_{A\oplus B}$ is equivalent to the result of the weighted ABF operator (3.8)*. For the special choice $c_A^A = c_A^B$, the result is equivalent to the result of the ABF operator. Moreover, for $c_A^A = (1 - u_A^A)$, $c_A^B = (1 - u_A^B)$, the same result as from the uncertainty-weighted ABF operator is obtained.*

Proof. The upscaling yields

$$\boldsymbol{r}_{A,\text{up}}^{A\cup B} = \alpha \left(c_A^A \boldsymbol{r}_A^A + c_A^B \boldsymbol{r}_A^B \right) \tag{4.30}$$

balls in $\tilde{S}^{A\cup B}_{A,\text{up}}$. Hence, the expected numbers of balls per color in $S_{A\tilde{\oplus}B}$ are

$$\mathbb{E}\left\{\boldsymbol{r}_{A\tilde{\oplus}B}\right\} = \frac{\alpha\left(c_A^A\,\boldsymbol{r}_A^A + c_A^B\,\boldsymbol{r}_A^B\right)}{\alpha(c_A^A + c_A^B)} = \frac{c_A^A\,\boldsymbol{r}_A^A + c_A^B\,\boldsymbol{r}_A^B}{c_A^A + c_A^B}\,. \tag{4.31}$$

Due to the choice of the upscaling (4.29), the prior sets contain $\text{card}\{S^A_{3,\text{up}}\} = \tilde{\alpha}\,c_A^A$ and $\text{card}\{S^B_{3,\text{up}}\} = \tilde{\alpha}\,c_A^B$ balls, respectively. Thus, the prior set $S_3^{A\tilde{\oplus}B}$ yields the prior

$$\boldsymbol{a}_{A\tilde{\oplus}B} = \frac{\tilde{\alpha}(c_A^A\,\boldsymbol{a}_A + c_A^B\,\boldsymbol{a}_B)}{\tilde{\alpha}(c_A^A + c_A^B)} = \frac{c_A^A\,\boldsymbol{a}_A + c_A^B\,\boldsymbol{a}_B}{c_A^A + c_A^B}\,. \tag{4.32}$$

Since Theorem 4.1 confirms the validity of bijective mapping, (3.5) can be used for the proof. Hence, with Algorithm 4.1 and (3.5), $\boldsymbol{r}_{A\tilde{\oplus}B}$ can be rephrased as

$$\mathbb{E}\left\{\boldsymbol{b}_{A\tilde{\oplus}B}\right\} = \frac{\mathbb{E}\left\{\boldsymbol{r}_{A\tilde{\oplus}B}\right\}}{W(c_A^A + c_A^B) + c_A^A\,N_A^A + c_A^B\,N_A^B} \overset{(3.5)}{=} \frac{c_A^A\,u_A^B\,\boldsymbol{b}_A^A + c_A^B\,u_A^A\,\boldsymbol{b}_A^B}{c_A^B\,u_A^A + c_A^A\,u_A^B}\,, \tag{4.33a}$$

$$\mathbb{E}\left\{u_{A\tilde{\oplus}B}\right\} = \frac{W}{W(c_A^A + c_A^B) + c_A^A\,N_A^A + c_A^B\,N_A^B} \overset{(3.5)}{=} \frac{(c_A^A + c_A^B)\,u_A^A\,u_A^B}{c_A^B\,u_A^A + c_A^A\,u_A^B}\,. \tag{4.33b}$$

Hence, for $S_{A\tilde{\oplus}B}$ and $S_3^{A\tilde{\oplus}B}$, the expected opinion obtained by Algorithm 4.1 is

$$\mathbb{E}\left\{\omega_{A\tilde{\oplus}B}\right\} = \left(\mathbb{E}\left\{\boldsymbol{b}_{A\tilde{\oplus}B}\right\}, \mathbb{E}\left\{u_{A\tilde{\oplus}B}\right\}, \boldsymbol{a}_{A\tilde{\oplus}B}\right) \overset{(3.8)}{=} \omega_A \,\tilde{\oplus}\, \omega_B\,. \tag{4.34}$$

□

As illustrated by the urn model for partial observability in Definition 4.2, the imperfect observation process adds noise to the observations. This noise, once injected, can hardly be removed from the result any more. Therefore, it is important to check the opinions for consistency before merging them. The opinions can be compared using the DC as the metric. Theorem 4.5 states, how this metric can be represented in terms of the urn model.

Theorem 4.5 (Urn Model Representation of the Degree of Conflict)
Consider two instances of the urn model from Definition 4.2, where the quantities corresponding to the first instance are marked with A and to the second instance are marked with B. Let ω_A and ω_B be opinions on the same domain $\mathbb{X}$, which is interpreted as set of colors. Furthermore, let $\tilde{S}_A$ and $\tilde{S}_B$ be the corresponding observations, i.e., sets of drawn balls for which $\tilde{\boldsymbol{r}}_A$, $\tilde{\boldsymbol{r}}_B$, $\boldsymbol{a}_A$, and $\boldsymbol{a}_B$ denote their evidence vectors and priors, respectively. The set S_A is put in the urn $\mathcal{U}_{2,A}$, and $\tilde{S}_B$ is put into $\mathcal{U}_{2,B}$, respectively. Furthermore, the set $\tilde{S}_{A,\perp}$ is constructed

such that the probability of drawing a ball of specific color from $\tilde{\mathcal{S}}_{A,\perp}$ is the projected probability $\tilde{\boldsymbol{P}}_{A,\perp}$. Similarly, $\tilde{\mathcal{S}}_{B,\perp}$ is constructed with respect to $\tilde{\boldsymbol{P}}_{B,\perp}$. Moreover, $\tilde{\mathcal{S}}_{A,\perp}$ and $\tilde{\mathcal{S}}_{B,\perp}$ are required to contain the same number of balls, i.e., $N_{A,\perp} = \mathrm{card}\{\tilde{\mathcal{S}}_{A,\perp}\} = \mathrm{card}\{\tilde{\mathcal{S}}_{B,\perp}\}$. For these sets, the intersection set $\mathcal{S}_{\perp,A\cap B} = \tilde{\mathcal{S}}_{A,\perp} \cap \tilde{\mathcal{S}}_{B,\perp}$ is replaced by checkered balls, each. Moreover, the modified sets are put into an additional urn $\mathcal{U}_{PD}$. This urn then contains the set $\mathcal{S}_{PD}$. A ball is drawn from each of the three urns $\mathcal{U}_{PD}$, $\mathcal{U}_{2,A}$ and $\mathcal{U}_{2,B}$. Then, the probability that none of the three drawn balls is a checkered ball is equivalent to the DC from SL.

Proof. Let p_{DC} denote the probability that none of the three drawn balls is checkered. Since the balls are drawn independently from the urns, p_{DC} can be split into

$$p_{DC} = p_{PD}\, p_{2,A}\, p_{2,B}\,,\ p_{2,A} = \frac{N_A}{N_A + W} = (1 - u_A)\,,\ p_{2,B} = \frac{N_B}{N_B + W} = (1 - u_B)\,, \quad (4.35)$$

where the probabilities $p_{2,A}$ and $p_{2,B}$ can be computed directly from the cardinalities N_A and N_B of $\tilde{\mathcal{S}}_A$ and $\tilde{\mathcal{S}}_B$. When calculating the probability distance

$$PD = \frac{1}{2}\sum_{i=1}^{W}\left|P_{A,\perp,i} - P_{B,\perp,i}\right| = \sum_{i=1}^{W}\frac{|r'_{A,\perp,i} + r_{A\cap B,i} - r_{A\cap B,i} - r'_{B,\perp,i}|}{2\,N_{A,\perp}} = \frac{N_{\mathrm{pd}}}{2\,N_{A,\perp}}, \quad (4.36)$$

the evidence vectors $\tilde{\boldsymbol{r}}_{A,\perp}$ and $\tilde{\boldsymbol{r}}_{B,\perp}$ corresponding to $\tilde{\mathcal{S}}_{A,\perp}$ and $\tilde{\mathcal{S}}_{B,\perp}$ can be split into the mutual part $\tilde{\boldsymbol{r}}_{A\cap B} \leftarrow \mathrm{NodeColors}(\mathcal{S}_{\perp,A\cap B}, \mathbb{X})$ and a rest $\tilde{\boldsymbol{r}}'_{A,\perp}$, $\tilde{\boldsymbol{r}}'_{B,\perp}$, where the mutual part cancels out. Moreover, for each $i = 1, \ldots, W$, either $r'_{A,\perp,i}$ or $r'_{B,\perp,i}$ has to be zero. Hence, N_{pd} corresponds to the number of balls in $\mathcal{S}_{\mathrm{PD}} = \left(\tilde{\mathcal{S}}_{A,\perp}/\mathcal{S}_{\perp,A\cap B}\right) \cup \left(\tilde{\mathcal{S}}_{B,\perp}/\mathcal{S}_{\perp,A\cap B}\right)$, i.e., the number of balls in $\mathcal{U}_{PD}$ that are not checkered balls. Thus, PD is the probability of drawing a ball from $\mathcal{U}_{PD}$ that has a color from $\mathbb{X}$. Therefore, the probability that none of the three balls drawn from $\mathcal{U}_{2,A}$, $\mathcal{U}_{2,B}$, and $\mathcal{U}_{PD}$ is checkered is

$$DC(\tilde{\omega}_A, \tilde{\omega}_B) = \frac{1}{2}\sum_{i=1}^{W}\left|\tilde{\boldsymbol{P}}_{A,\perp} - \tilde{\boldsymbol{P}}_{B,\perp}\right|\,(1 - \tilde{u}_A)\,(1 - \tilde{u}_B)\,. \quad (4.37)$$

□

4.3 Summary

In this section, a novel urn model representation of SL has been proposed. First, the urn model was motivated and derived intuitively. Then, the model was formalized, and its key features were proven mathematically. The urn model links the Pólya's urn model [EP23] and SL. Furthermore, the connection between Bayesian inference and the CBF operator from SL

was proven. With the urn model, the otherwise complex theory can be illustrated. This may support the community's in-depth understanding of SL.

Chapter 5

Subjective-Logic-based Reliability Estimation

In many challenging driving scenarios, cooperative information can be helpful for a CAV to efficiently master the respective driving task. However, it is vital for the CAV that the cooperative information is reliable. For example in the application scenario, if the CAV fully relies on the external environment model, but the ITSS overlooks a road user on the main road occluded to the CAV, in the worst case, the CAV could collide with that road user, since the CAV might brake too late. To reduce the risk of such an incidence, the CAV can independently estimate the reliability of the external information before relying on it. In case the estimated reliability of the data is not sufficient, the CAV can discard the information and fall back to a fail-safe mode, where performance is sacrificed in favor of safety. For the application scenario, this would mean stopping gently at the yield line and checking that the junction is clear with own sensors before merging.

In this chapter, an SL-based reliability estimation scheme is presented to support decision making, as described in Section 6.2.2. For that, an SL opinion on the parameters of the communication channel is identified from which the reliability of the V2X link can be inferred. Furthermore, four tests are formulated to check that the semantic of the received data is consistent and plausible. Both the reliability estimation of the V2X link and that of the communicated data are evaluated through real-world experiments. In a further step, the reliability estimation is generalized to operate on environment models shared among CAVs.

One particular challenge in relying on information received from other CAVs is the trustworthiness of the transmitting CAV. To address this, a trust management and misbehavior detection mechanism is proposed to assure trustworthiness among a vehicular multi-agent system consisting of a network of CAVs, infrastructure units and MEC devices. Since many state-of-the-art trust management systems suffer from bad scalability due to high computational cost and the inability to distribute the computational load over the multi-agent system, a large-scale traffic simulation is performed to study the scaling properties of the mechanism.

5.1 Reliability Estimation of a Communication Channel

Consider the CAV dynamically merging according to the application scenario from Section 2.1. Then, the communication link breaks and renders the CAV blinded, which may lead to a dangerous situation. To prevent such situation, the reliability of the communication channel can be estimated and used as an early warning indicator. In this way, although the CAV cannot prevent the link to break, it is warned and can adjust its behavior. For that, the reliability can be considered as a constraint in the situation context of the motion planning (see Section 6.1.1). In this section, based on the jointly pre-published work [MGGB19] of the author and Thomas Griebel, an SL-based method is presented to estimate the reliability of a communication channel. While the author of this thesis developed the basic idea of the method focusing on the safety of the CAV, conducted the simulations and experiments, Thomas Griebel contributed in formalizing the work [MGGB19]. A common way to model communication channels is the Markov-chain-based Gilbert-Elliot model [Ell63; Gil60], which is still frequently used, e.g. in [IFE17]. The parameters of this model then are identified using the new SL-based method. With the identified model, the reliability and the expected delay of the channel can be inferred [ZTW+19]. The advantage of the presented method over existing ones such as [ZTW+19] is that, in addition to the parameters, it estimates the statistical uncertainty of the identification result, which provides additional information on the inferred reliability.

5.1.1 Problem Formulation

The communication channel in the Gilbert-Elliot model is described by a Markov chain, which allows to infer the reliability of the communication link. Accordingly, the principal goal of the method is to identify this Markov chain. A Markov chain $X(t_k)$ is a time and value discrete stochastic process for which the probability for the next value only depends on the current one [BSMM07, p. 766]. This means, a Markov chain $X(t_k) \in \{1, \dots, W\}$ is a random variable which can only take one of the W values, the states, at the discrete times t_k, $k \in \mathbb{N}$. A realization, i.e. a measurement of the state at t_k is termed *observation* o_k. Furthermore, the probability vector $\boldsymbol{\pi}_k$ describes for each state the probability of being in this state at time t_k. Accordingly, $\boldsymbol{\pi}_{k+1}$ is the probability vector for the next state. It is computed with the transition matrix $\boldsymbol{P}(t_k, t_{k+1})$, i.e.,

$$\boldsymbol{\pi}_{k+1} = \boldsymbol{P}(t_k, t_{k+1})\, \boldsymbol{\pi}_k \,. \tag{5.1}$$

where the entries p_{ij} of $\boldsymbol{P}(t_k, t_{k+1})$ describe the probability of getting into the state j at time t_{k+1}, if the state i is measured at t_k. Hence, the rows of $\boldsymbol{P}(t_k, t_{k+1})$ sum to one. Thus, in essence, $\boldsymbol{P}(t_k, t_{k+1})$ has to be identified from a sequence of observations $\langle o_m \rangle_{m=k-l_w,\dots,k}$, where $l_w \in \mathbb{N}$ denotes the length of a time window. This is, $\boldsymbol{P}(t_k, t_{k+1})$ varies over time,

but is assumed constant for l_w time steps. In addition, a reliability measure has to provide information on the evidence-based certainty of the result.

5.1.2 Identification Algorithm

To estimate the transition matrix $\boldsymbol{P}(t_k, t_{k+1})$, the proposed identification algorithm, Algorithm 5.1, is based on the classical idea of grouping the incoming stream of observations into mini-batches, counting the state transitions, and identifying the state transition probabilities from it. However, in contrast to classical approaches, SL rather than classical probability theory is used to estimate the transition probabilities. Thus, the algorithm returns a vector of SL opinions $\boldsymbol{\Omega}(t_k)$ rather than just probabilities, so that additional information on the statistical uncertainty within the estimation is provided. Since the rows of $\boldsymbol{P}(t_k, t_{k+1})$ sum to one, an SL opinion $\omega_i(t_k)$ can be formulated for each row of $\boldsymbol{P}(t_k, t_{k+1})$. The SL opinions are stacked into $\boldsymbol{\Omega}(t_k) = [\omega_i(t_k)]_{i=1,\dots,W}$ and the required priors are stacked into the prior matrix $\boldsymbol{A} = [\boldsymbol{a}_i]$. The opinion vector $\boldsymbol{\Omega}(t_k)$ is initialized with the vacuous opinions $\omega_{\emptyset,i} = (\boldsymbol{0}, 1, \boldsymbol{a}_i)$, i.e. $\boldsymbol{\Omega}(0) = [\omega_{\emptyset,i}]_{i=1,\dots,W}$. Then, the identification algorithm is carried out. It takes the prior matrix $\boldsymbol{A}$, the discount probabilities $\boldsymbol{p}_d$ and $\tilde{\boldsymbol{p}}_d$, the consistency threshold θ_{DC}, the opinion vector $\boldsymbol{\Omega}(t_{k-1})$ of the last step, and the incoming sequence $\langle o_m \rangle_{m=k-l_w,\dots,k}$ of observations as its input and empirically determines the transition matrix $\boldsymbol{P}(t_k, t_{k+1})$ and the opinion vector $\boldsymbol{\Omega}(t_{k+1})$ from it. The sequences $\langle o_m \rangle_{m=k-l_w,\dots,k}$, i.e. the mini-batches, are obtained by buffering l_w observations.

The choice of l_w comes with a trade-off: on the one hand, small windows will enable the system to quickly react to changes in the parameters allowing for high dynamics in the time-varying parameters and furthermore contribute to the system's stability by featuring a short dead time. On the other hand, big values of l_w result in lower statistical uncertainty and higher robustness towards statistical outliers. In practice, the choice of l_w will mainly depend on the maximum acceptable dead time, the necessary tracking precision, and the expected transition probabilities, since very rare events require big windows for a realistic chance to observe them.

The algorithm basically performs three steps: the preprocessing, the consistency test, and the update of the opinions. During preprocessing, the state transitions from state i to the other states $j = 1, \dots, W$ that occur in the sequence of observations $\langle o_m \rangle_{m=k-l_w,\dots,k}$ are counted. Depending on what is actually measured, a classifier is needed to determine how many state transitions and which of them have been observed within $\langle o_m \rangle_{m=k-l_w,\dots,k}$. The result is stashed into the evidence vectors $\tilde{\boldsymbol{r}}_i(t_k)$ and the corresponding belief masses $\tilde{\boldsymbol{b}}_i(t_k)$, the statistical uncertainties $\tilde{u}_i(t_k)$, and the SL opinions $\tilde{\omega}_i$ are calculated using the mapping (3.5). Furthermore, the generated opinions $\tilde{\omega}_i(t_k)$ and the opinions from the last step $\omega_i(t_{k-1})$ are trust discounted with the discount probabilities $\tilde{p}_{d,i}$ and $p_{d,i}$, respectively.

Algorithm 5.1 SL-Based Identification of Markov Chains [MGGB19]

Input: Prior matrix $\boldsymbol{A} = \left[\boldsymbol{a}_i\right]_{1,\dots,W}$, last result $\boldsymbol{\Omega}(t_{k-1})$, sequence of observations $\langle o_m \rangle_{m=k-l_w,\dots,k}$, consistency threshold θ_{DC}, discount probabilities $\boldsymbol{p}_d, \tilde{\boldsymbol{p}}_d \in [0,1]^W$, number of states W,
Output: Estimated transition matrix $\boldsymbol{P}(t_k, t_{k+1})$, corresponding opinion vector $\boldsymbol{\Omega}(t_k)$

1: isConsistent $\leftarrow$ **true**
2: **for** $i = 1$ **to** W **do**
3: $\quad \tilde{\boldsymbol{r}}_i(t_k) \leftarrow \text{TransitionStatistics}\big(\langle o_m \rangle_{m=k-l_w,\dots,k}\,,\, i\big)$
4: $\quad \tilde{\boldsymbol{b}}_i(t_k) \leftarrow \frac{\boldsymbol{r}_i}{W+\sum_{j=1}^{W} r_{ij}}$
5: $\quad \tilde{u}_i(t_k) \leftarrow \frac{W}{W+\sum_{j=1}^{W} r_{ij}}$
6: $\quad \tilde{\omega}_i \leftarrow (\tilde{\boldsymbol{b}}_i,\, \tilde{u}_i,\, \boldsymbol{a}_i)$
7: $\quad \tilde{\omega}_{D,i}(t_k) \leftarrow \mathcal{D}\big(\,\tilde{\omega}_i(t_k),\, \tilde{p}_{d,i}\,\big)$
8: $\quad \omega_{D,i}(t_{k-1}) \leftarrow \mathcal{D}\big(\omega_i(t_k), p_{d,i}\big),\ \omega_i(t_{k-1}) \in \boldsymbol{\Omega}(t_{k-1})$
9: $\quad DC_i \leftarrow DC\big(\tilde{\omega}_{D,i}(t_k), \omega_{D,i}(t_{k-1})\big)$
10: $\quad$ **if** $DC_i \leq \theta$ **and** isConsistent = **true then**
11: $\quad\quad \omega_{\text{up},i}(t_k) \leftarrow \tilde{\omega}_{D,i}(t_k) \oplus \omega_{D,i}(t_{k-1})$
12: $\quad$ **else**
13: $\quad\quad$ isConsistent $\leftarrow$ **false**
14: $\quad$ **end if**
15: **end for**
16: **if** isConsistent = **false then**
17: $\quad \omega_{\text{up},i}(t_k) \leftarrow \tilde{\omega}_{D,i}(t_k)\ \forall i = 1, \dots, W$
18: **end if**
19: $\boldsymbol{P}(t_k, t_{k+1}) \leftarrow \left[\boldsymbol{b}_{\text{up},i}(t_k) + u_{\text{up},i}(t_k)\,\boldsymbol{a}_i\right]_{i=1,\dots,W}$
20: $\boldsymbol{\Omega}(t_k) \leftarrow \left[\omega_{\text{up},i}(t_k)\right]_{i=1,\dots,W}$
21: **return** $\boldsymbol{P}(t_k, t_{k+1}), \boldsymbol{\Omega}(t_k)$

Preprocessing (lines 3–8)
Consistency Test (lines 9–15)
Update (lines 19–21)

In the next step, the consistency between the trust discounted opinions $\tilde{\omega}_{D,i}(t_k)$ and $\omega_{D,i}(t_{k-1})$ is checked using the DC. If the opinions are consistent, i.e., if the DC is below the consistency threshold θ_{DC}, the current opinion is merged with the last opinion using the CBF operator. Otherwise, the $\omega_{\text{up},i}(t_k)$ are overwritten with $\tilde{\omega}_{D,i}(t_k)$ obtained from $\langle o_m \rangle_{m=k-l_w,\dots,k}$.

Finally, the individual opinions on the respective transitions are reassembled in $\boldsymbol{\Omega}^{t_{k+1}}$ during the update step. Furthermore, the transition matrix $\boldsymbol{P}(t_k, t_{k+1})$ is calculated element-wise using the projected probabilities (3.3) of the corresponding entries in $\boldsymbol{\Omega}^{t_{k+1}}$.

The model for the communication channel usually is an application driven simplification of the underlying physical processes. Hence, finding the ground truth channel model corresponding to a real-world experiment proofs impossible in practice. Thus, it can only be approximated by using established identification schemes in combination with acausal postprocessing of the recorded data. But even then, the ground truth models generated this way will never be

absolutely accurate. Therefore, Algorithm 5.1 is first evaluated on simulations, where the exact ground truth is known. Based on this, in a consecutive step presented in Section 5.1.4, it is demonstrated by experiment, that the algorithm works in a real-world setting. The algorithm parameters used for the simulations and the experiments are listed in Appendix B.

5.1.3 Simulative Evaluation

For the simulation scenario, a Long Term Evolution (LTE) communication link is assumed that is evaluated for packet error rate before the first correction layer in LTE, the Hybrid Automated Response Request (HARQ) mechanism. The channel itself is modeled with the classical Gilbert-Elliot model [Gil60], i.e., as a two-state Markov chain with the states *Good* (G) and *Bad* (B). Hence, the general problem formulation (5.1) turns into

$$\boldsymbol{\pi}(t_{k+1}) = \begin{bmatrix} p_{GG}(t_k, t_{k+1}) & 1 - p_{GG}(t_k, t_{k+1}) \\ 1 - p_{BB}(t_k, t_{k+1}) & p_{BB}(t_k, t_{k+1}) \end{bmatrix} \cdot \boldsymbol{\pi}(t_k), \tag{5.2}$$

where Algorithm 5.1 creates an opinion on the state transition probabilities $p_{GG}(t_k, t_{k+1})$, $p_{BB}(t_k, t_{k+1})$, and the respective complementary events from samples.

In communication networks, bit errors are detected and corrected by error detection and correction codes that add up redundancy to the transmitted messages and therefore reduce the effective throughput of a communication channel. Thus, a trade-off arises: for channels with little distortions, transmitting the messages with detection codes only and retransmitting whenever a message gets corrupted might be more efficient than sending correction codes for all messages. This, in essence, is called Automated Response Request (ARQ) [Sau14, p. 263-266]. For strongly distorted channels, more complex correction mechanisms will outmatch ARQ. A further developement in error coding is the HARQ [Sau14, p. 263-266] mechanism that calculates correction codes for each message, but only transmits the messages with error detection codes. In case a message gets corrupted, the correction codes are transmitted rather than retransmitting the whole message. Thus, in average, the effective channel throughput is increased. To maximize the effective channel capacity using HARQ, the probability of correctly decoding a received package before HARQ has to be significantly below 100 %, while the packet error rate after HARQ usually is practically negligible. In LTE, the communication channel is typically controlled such that the packet error rate before HARQ is about 10 %. Thus, the simulation scenario starts with $p_{GG} = 90\,\%$ (see Fig. 5.1(a)). The simulation scenario is then evaluated over 100 000 packets, while at packet 19 081 p_{GG} instantaneously drops to 70 %. This can be interpreted, for example, as a sudden drop in Signal-to-Noise ratio (SNR) due to the sudden transition from line of sight (LOS) to non-line of sight (NLOS) conditions. For the following packets, p_{GG} recovers in a linear way, e.g. because the underlying controls in the base station take measures to reestablish the 90 % set value. At packet 30 851, there is another sudden change, as p_{BB} suddenly steps to 60 %. This

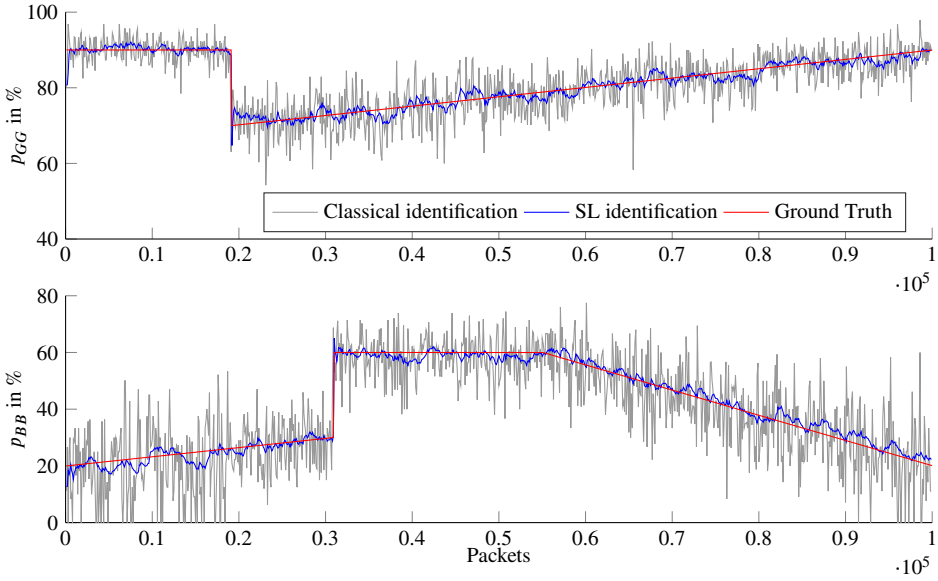
can be interpreted as the occurrence of sudden burst errors due to atmospheric interference. Before the sudden change, p_{BB} is increased linearly, starting with about $p_{BB} \approx 20\ \%$. After the step, p_{BB} is constant first, and decays linearly afterwards. The packet numbers at which changes occur are chosen as prime numbers to prevent an overly quick response by overfitting the algorithm to the simulation example by choosing a window size that is matched to the parameter changes. From a control perspective, the scenario includes challenging situations, as the steps require the identification algorithm to support high dynamics. The linear changes in between, in turn, require the algorithm to be sensitive with respect to parameter drifts.

Figure 5.1 shows the results of a classical identification method (cf. [Bis06, p.617]) and the results of Algorithm 5.1 in comparison, both with window size $l_w = 100$. It can be seen that Algorithm 5.1 identifies the parameters with much less variance. Furthermore, the proposed algorithm features an estimate on the statistical uncertainty. At the parameter steps, it can be seen that this uncertainty estimate jumps, since the algorithm has to discard the old measurements and resets its opinion. Then, the uncertainty decreases again to the level before, since the following mini-batches lead to opinions that are consistent with the previous ones.

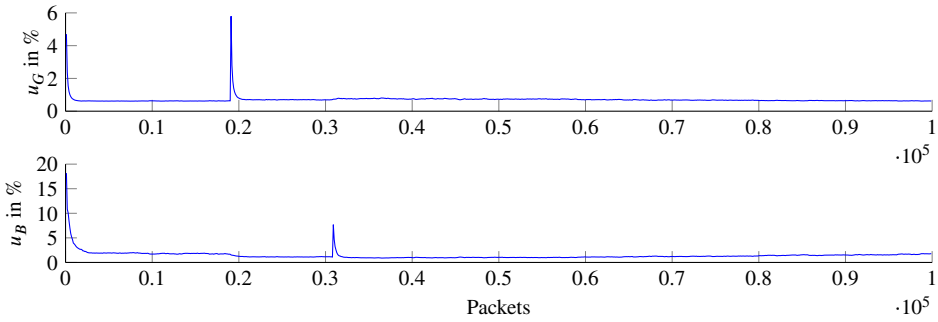
5.1.4 Real-World Experiment

The basic idea of the real-world experiment described in this section is to infer the reliability of the communication channel in the CAV by modeling the channel as a Markov chain and identifying its parameters by Algorithm 5.1 based on the measured delays of the incoming packets. The delay Δt of an incoming packet is determined as the difference between the time stamp when the CAV received the package and the timestamp when the MEC server transmitted the package. For that, the clocks of both, the CAV and the MEC server are synchronized to GPS time. The channel is modeled as a three-state Markov chain where the first state indicates that a package has been correctly received, while the second state indicates that the packet was corrected by the HARQ mechanism of the mobile network, and the third state represents the situation when further correction mechanisms are applied to the received packet.

However, the state transitions for the respective model cannot be observed directly, but have to be inferred from the delays by an additional classifier. It is known from the specification that if a package cannot be decoded immediately, the HARQ mechanism adds 7 ms delay to the package. If further correction is necessary, the package will arrive with further delay above the base line. However, the baseline, i.e., the delay involved in a packet successfully decoded on the first attempt, is not constant. Rather, it is dependent on various influences, such as atmospheric conditions or work load at the base station, and is therefore slowly time-varying. Thus, before the state transitions that a package took through the communication channel can be inferred, the base line has to be estimated. This is done using a moving average



(a) Identification Results.



(b) Uncertainty ([MGGB19], © 2019 IEEE).

Figure 5.1: Results of the classical identification and the SL-based algorithm for the simulated LTE channel in comparison (cf. [MGGB19]).

filter with outlier detection. The filter identifies the average delay t_{MA} of a package, which is successfully decoded at the first try. To account for statistical variation in the delay of packages that have been decoded correctly at the first try, the decision thresholds to classify a package as directly decoded is set to $t_{\mathrm{MA}} + 3.5\,\mathrm{ms}$. Following that logic, packages arriving with a delay $t_{\mathrm{MA}} + 3.5\,\mathrm{ms} \leq \Delta t < t_{\mathrm{MA}} + 10.5\,\mathrm{ms}$ are classified as corrected by the HARQ mechanism, while packages arriving with a delay $\Delta t > t_{\mathrm{MA}} + 10.5\,\mathrm{ms}$ are classified as corrected by further correction layers. In summary, the state transition classifier is described by

$$o_{ij} = \begin{cases} 1 \to 1, & \text{if } \Delta t < \theta_{\Delta t,1} = t_{\mathrm{MA}} + 3.5\,\mathrm{ms} \\ 1 \to 2 \to 1, & \text{if } \theta_{\Delta t,1} \leq \Delta t < \theta_{\Delta t,2} = t_{\mathrm{MA}} + 10.5\,\mathrm{ms} \\ 1 \to 2 \to 3 \to 1, & \text{otherwise} \end{cases} \tag{5.3}$$

Figure 5.2 visualizes the measured delay of a sequence of about 127 400 packages measured over about three hours. The decision threshold for the classifier (5.3) are marked in blue and red, respectively.

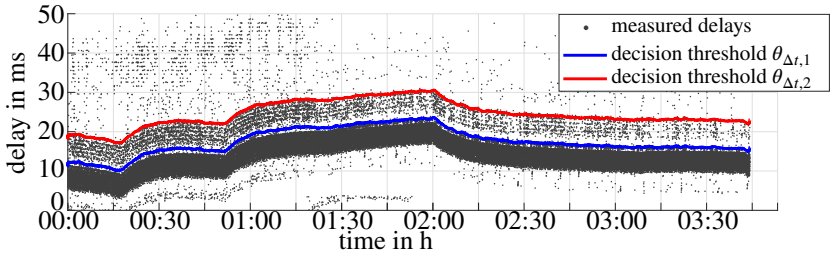
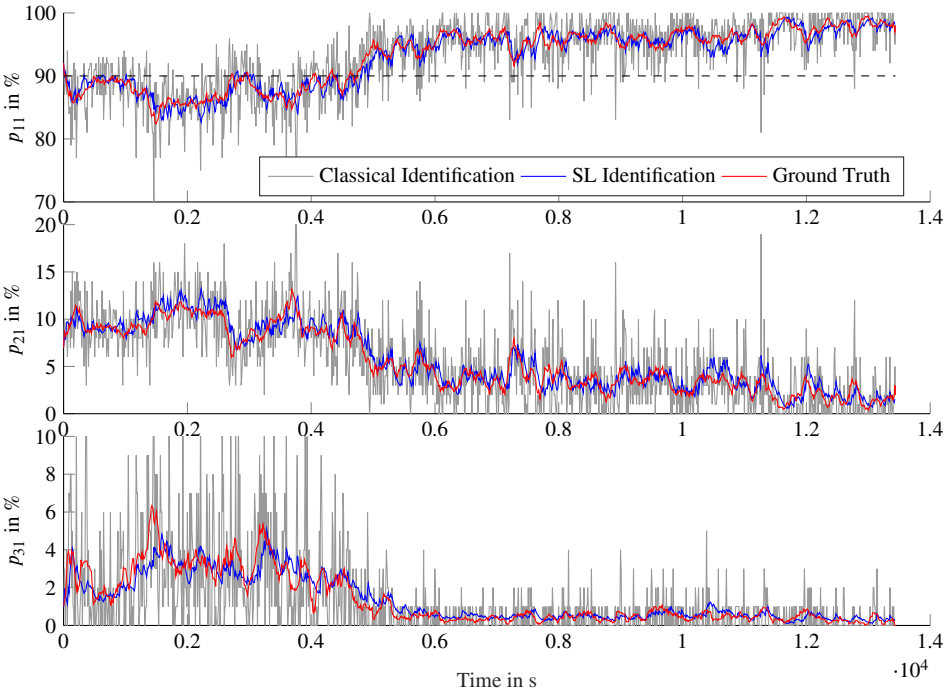
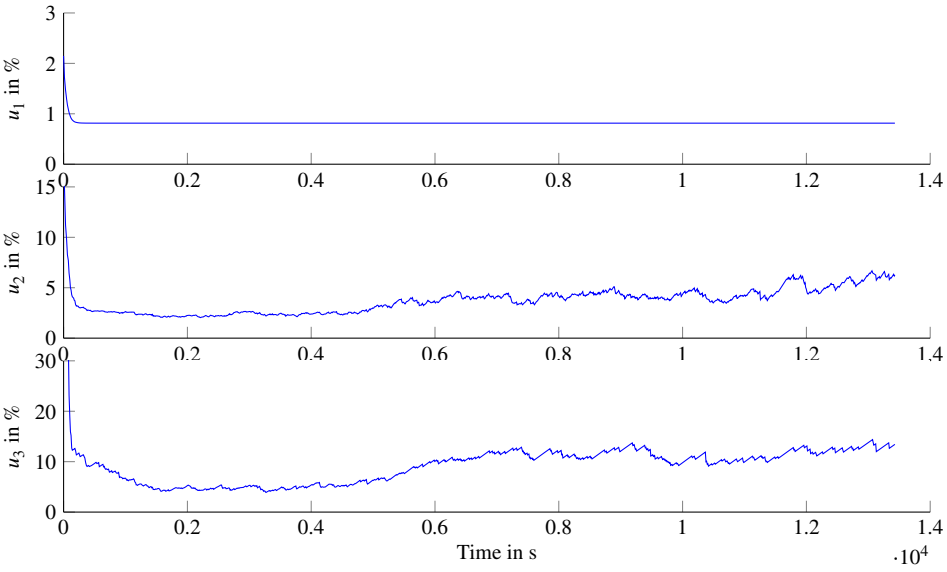


Figure 5.2: Recorded delays of a sequence of 127 400 packets. The decision thresholds are marked in blue and red, respectively (cf. [MGGB19]).

To obtain a ground truth channel model, first, a classical identification scheme with a window size $l_w = 100$ is used. Since the result is still very noisy, it then is improved by an acausal postprocessing with a moving average filter over the identification result of the whole sequence. This way, the best possible estimation of the state transition is calculated. The results of the classical identification (without filtering), the ground truth data (with filtering), and the results from Algorithm 5.1, all evaluated for $l_w = 100$, are compared in Fig. 5.3. As can be seen in Fig. 5.3, the estimation result of Algorithm 5.1 matches well with the ground truth data and comes with significantly reduced variance as compared to the classical identification. In addition, Algorithm 5.1 features an uncertainty measure that calculates the statistical uncertainty within the estimation.



(a) Experimental result on the communication channel identification.



(b) Statistical Uncertainty from SL identification ([MGGB19], © 2019 IEEE).

Figure 5.3: Comparison of the results from Algorithm 5.1 with a classical identification method for the experimental LTE channel identification (cf. [MGGB19]).

5.2 On-board Reliability Estimation of External Information

In this section, based on the author's pre-published paper [MGB19], a new method is presented that estimates the reliability of an external perception source using SL. The continuous opinion on the reliability of the data then can be further used in decision making, as described in Section 6.2.2. The key idea behind the reliability estimation scheme is to check the incoming cooperative data first for consistency using the incoming data only, and second for plausibility using the incoming data in combination with the ego perception of the CAV. Figure 5.4 gives an overview on the reliability estimation scheme.

The reliability estimation scheme comprises two consistency tests and two plausibility tests: the prediction test, in which former predictions are compared to current measurements; the map test, in which the objects are compared to the map; the ego localization test, in which the estimated ego position is compared to the position of the CAV from the external environment model; and finally the ego perception test, in which the objects perceived by the CAV are compared to the objects from the external environment model. The proposed reliability estimation scheme is configured for and experimentally evaluated at the test junction in Ulm-Lehr (see Chapter 2.1).

From application perspective, overestimating the reliability of the ITSS can have severe safety implications, while underestimating the reliability only reduces the performance in most cases. For example, if a road user is overlooked in the application scenario, the CAV might attempt to merge into a traffic gap that actually is not there. This results in a critical situation. In contrast, if a clutter object appears in the merged environment model, the vehicle will unnecessarily stop at the junction which results in nothing but a slightly reduced performance. This asymmetry is reflected in the reliability estimation in general and the plausibility tests in particular. For this, similar to a reputation system, trust is built slowly based on lots of positive evidence, while similar to a fault detection mechanism, trust is strongly decreased in case the incoming data appear implausible. More specifically, the ego perception test result is intended to minimize the decision risk rather than maximizing the likelihood of the actual traffic situation. This is something to keep in mind when using the reliability estimation result for trust building in a V-MAS (see Chapter 5.3.4).

Since the environment model of the MEC server and the perception of the ego vehicle are processed asynchronously, time-synchronicity has to be considered. On the one hand, the errors occurring due to asynchronous clocks affect the planning and decision making. Therefore, it makes sense to refrain from correcting such errors and instead simply use the most recently obtained data for the comparison. On the other hand, it is also reasonable to correct these known issues as good as possible by, e.g., buffering and interpolating the data over one time step. At the cost of a delay, this makes the plausibility test more specific.

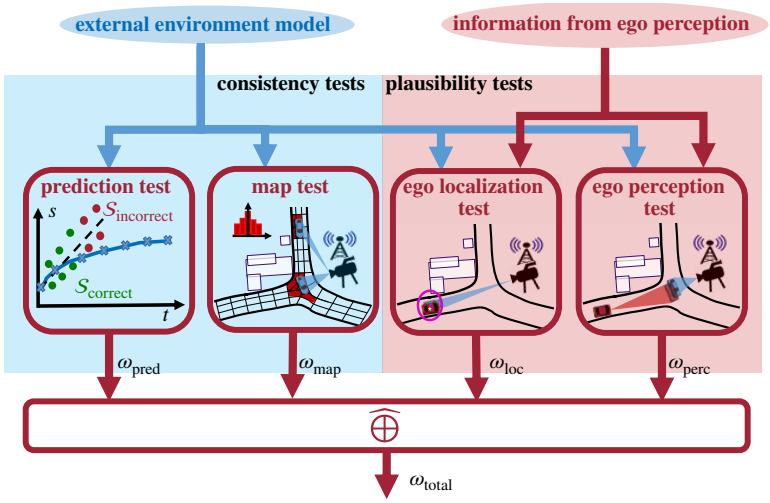


Figure 5.4: Overview on the reliability estimation scheme ([MGB19], © 2019 IEEE).

5.2.1 Prediction Test

The basic idea behind the prediction test is to collect the predicted states from former predictions as well as their uncertainties and compare the subsequently received measurements with these predictions. For comparison, the Mahalanobis distance d_M [Mah36] is used. If the distance between a measured and a predicted state is below a threshold $\theta_{M,\text{pred}}$, e.g. $\theta_{M,\text{pred}} = 3$ as used in this section, the predicted state is considered consistent. Otherwise, it is considered inconsistent, which leads to a low estimated reliability. Speaking in terms of the urn model from Definition 4.2, a red ball is drawn from the first urn whenever a measurement is within the uncertainty interval of a corresponding predicted state, i.e., if $d_M \leq \theta_{M,\text{pred}}$. Otherwise, if $d_M > \theta_{M,\text{pred}}$, a black ball is drawn from the first urn. It is assumed that the received external environment model contains state predictions $\boldsymbol{x}_{\text{pred}}$ given over a fixed time grid. For example, each object in the external environment model may contain state predictions for 0.5 s, 1.0 s, … 3.0 s. Furthermore, it is assumed that the currently measured state $\boldsymbol{x}_{\text{meas}}$ as well as the predictions contain covariance matrices indicating the uncertainties. Hence, each predicted state can be uniquely associated with a subsequently measured state and it can be decided whether a predicted state is consistent or not. If the external environment model contains multi-modal predictions, the predictions have to be treated depending on the use in the risk model of the motion planning (see Section 6.2.2). For example, if the risk model only considers the most likely prediction, the prediction test discards all other predictions. Otherwise, if the CAV plans based on the worst-case prediction, it is sufficient that one of the predicted trajectories contains a predicted state which fulfills $d_M \leq \theta_{M,\text{pred}}$.

Thus, the predictions can be split into the subsets

$$\mathcal{S}_{\text{correct}} = \{\boldsymbol{x}_{\text{pred}}(t_k) \mid d_M(\boldsymbol{x}_{\text{pred}}(t_k), \boldsymbol{x}_{\text{meas}}(t_k)) \leq \theta_M\}, \tag{5.4a}$$

$$\mathcal{S}_{\text{incorrect}} = \{\boldsymbol{x}_{\text{pred}}(t_k) \mid d_M(\boldsymbol{x}_{\text{pred}}(t_k), \boldsymbol{x}_{\text{meas}}(t_k)) > \theta_M\}, \tag{5.4b}$$

where t_k is the time step to which the respective states refer to.

In the urn model, Definition 4.2, the evidence vector $\tilde{\boldsymbol{r}}_{\text{pred}} = [\text{card}\{\mathcal{S}_{\text{incorrect}}\}, \text{card}\{\mathcal{S}_{\text{correct}}\}]^T$ reflects the number of red and black balls in $\tilde{\mathcal{S}}_A$, with $W = \text{card}\{\{\bullet, \bullet\}\} = 2$. Since the predicted states are correlated, some of the balls in $\tilde{\mathcal{S}}_A$ do not originate from $\mathcal{U}_1$, i.e., not all data carry information. Hence, the statistical uncertainty in the opinion $\tilde{\omega}_{\text{pred}}$, which is obtained from $\tilde{\boldsymbol{r}}_{\text{pred}}$ via (3.5), is underestimated. Since it is not known, which balls originate from $\mathcal{U}_1$, i.e., which part of the measurements is uncorrelated, the problem has to be addressed in a probabilistic way. For that, some balls are randomly removed from $\tilde{\mathcal{S}}_A$ using Algorithm 4.3 until the uncertainty fits its expectation. To realize this, $\tilde{\omega}_{\text{pred}}$ is trust discounted with the probability p_{indep} expressing the degree of correlation. This yields

$$\omega_{\text{pred}} = \mathcal{D}(\tilde{\omega}_{\text{pred}}, p_{\text{indep}}). \tag{5.5}$$

5.2.2 Map Test

The basic idea behind the map test is to compare the objects from the external environment model with the CAV's map. If the received objects match the road, the environment model is consistent and thus appears reliable. Otherwise, if reported objects drive over static obstacles according to the CAV's map, the incoming data is inconsistent with the map so that the estimated reliability drops. The map information is represented by a set of reference opinions $\mathcal{S}_{\omega,\text{ref}}$, which express the probability of observing an object in a respective position. For this purpose, the map is gridded and each grid cell is assigned a probability of observing an object in this position. Since the observed objects are supposed to follow the lanes, only the lateral positions of the objects carry significant information. Hence, the grid cells with the same longitudinal positions are bundled to lane segments. For each lane segment, a reference opinion $\omega_{\text{ref},i} \in \mathcal{S}_{\omega,\text{ref}}$ on the expected lateral position of an observed object is formulated. To explicitly account for the extents of the objects, a reference opinion can be constructed for each corner point of the bounding box around an object and the respective opinions can be added to $\mathcal{S}_{\omega,\text{ref}}$. However, for simplicity, in this thesis, only the object centers are considered. This suffices for most practical cases. The reference opinion can be obtained in many ways. For example, expert knowledge, fleet data, cooperative awareness message (CAM), or traffic observations can be used as data sources. Here, the reference opinion is obtained from traffic observations with the reference measurement system. For that, histograms over the recorded object positions are set up for each lane segment. Then, the occurrence probabilities assigned to the respective grid cells are calculated empirically from the histograms.

In the online mode, a set of multinomial opinions $\mathcal{S}_{\omega,\text{online}}$ is constructed by entering the reported object positions into the grid of the reference opinions. While for the reference opinions $\omega_{\text{ref},i}$ the number $N_{\text{ref},i}$ is big, $N_{\text{online},i}$ will be small, or even zero at times. To measure the consistency between the reported objects and the map, the DC is used as metric. For that, the averaged DC between the reference opinions in $\mathcal{S}_{\omega,\text{ref}}$ and the online gathered opinions in $\mathcal{S}_{\omega,\text{online}}$ is calculated by

$$DC_{\text{map}}^{\text{online}} = \frac{1}{\text{card}\{\mathcal{S}_{\omega,\text{ref}}\}} \sum_{i=1}^{\text{card}\{\mathcal{S}_{\omega,\text{ref}}\}} \sum_{j=1}^{W_i} \left| P_{o,\perp,ij} - P_{\text{ref},\perp,ij} \right| \frac{(1-u_{\text{ref},i})(1-u_{o,i})}{W_i}, \tag{5.6}$$

where $P_{o,\perp,ij}$ and $P_{\text{ref},\perp,ij}$ are the projected probabilities corresponding to $\omega_{o,i} \in \mathcal{S}_{\omega,\text{online}}$ and $\omega_{\text{ref},i} \in \mathcal{S}_{\omega,\text{ref}}$, respectively. Furthermore, W_i are the cardinalities of the event spaces. From $DC_{\text{map}}^{\text{online}}$, the binomial opinion ω_{map} is derived, which expresses the reliability of the external environment model according to the map test. Whenever the averaged threshold is below a threshold, i.e. $DC_{\text{map}}^{\text{online}} < \theta_{\text{DC,map}}$, the reported objects are considered consistent with the map. In this case, the estimated reliability is increased. Mathematically, this is expressed as

$$\omega_{\text{map}}(0) = \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, 1, \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \right), \tag{5.7a}$$

$$\omega_{\text{evid}}(k) = \left(\begin{bmatrix} \frac{p_d\, N_{\text{obj}}(k)}{W + p_d\, N_{\text{obj}}(k)} \\ 0 \end{bmatrix}, 1 - \frac{p_d\, N_{\text{obj}}(k)}{W + p_d\, N_{\text{obj}}(k)}, \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \right), \tag{5.7b}$$

$$\omega_{\text{map}}(k+1) = \omega_{\text{map}}(k) \oplus \omega_{\text{evid}}(k), \tag{5.7c}$$

where ω_{evid} corresponds to the statistical evidence obtained from observing N_{obj} objects in the scene and p_d is a discount probability accounting for the statistical dependence of the individual observations. Furthermore, k indicates the respective time steps.

In contrast, if the reported object positions are inconsistent with the map, i.e. $DC_{\text{map}}^{\text{online}} \geq \theta_{\text{DC}}$, the reliability estimated by the map test is revised through a modified trust revision operation (cf. [Jøs16, p. 266-268])

$$\omega_{\text{map}}(k+1) = \begin{cases} \tilde{b}_1(k+1) = \left(1 - DC_{\text{map}}^{\text{online,ref}}\right) b_1(k) \\ \tilde{b}_2(k+1) = b_2(k) + DC_{\text{map}}^{\text{online,ref}} (1 - b_2(k)) \\ \tilde{u}(k+1) = \left(1 - DC_{\text{map}}^{\text{online,ref}}\right) u(k) \\ \tilde{\boldsymbol{a}}(k+1) = \boldsymbol{a}(k) \end{cases}. \tag{5.8}$$

In analogy of the urn model representation of the DC, Theorem 4.5, each lane segment corresponds to three urns $\mathcal{U}_{\text{PD},i}$, $\mathcal{U}_{\text{2A},i}$, $\mathcal{U}_{\text{2B},i}$. The balls in $\mathcal{U}_{\text{2A},i}$ reflect the opinion $\omega_{\text{online},i}$

the balls in $\mathcal{U}_{2B,i}$ reflect $\omega_{ref,i}$, and the balls in $\mathcal{U}_{PD,i}$ reflect the mutual and disjoint parts of the corresponding projected probabilities. Furthermore, each lateral grid cell corresponds to a color in $\mathbb{X}$. Then, a ball is drawn from each urn and it is counted, how often none of the balls drawn from an urn triplet $\mathcal{U}_{PD,i}$, $\mathcal{U}_{2A,i}$, $\mathcal{U}_{2B,i}$ is a checkered ball. The expectation of this count referred to the number of lane segments then corresponds to DC_{map}^{online}.

To summarize, the map test is designed such that its estimated reliability, represented by the binomial opinion ω_{map}, increases slowly as long as the objects observed online are consistent with the reference, while ω_{map} decreases quickly when inconsistencies are detected.

5.2.3 Ego Localization Test

Optimizing the planned maneuver for safety strongly depends on the uncertainties included in the environment model of the external perception. Hence, it is an important aspect of the reliability estimation to check the uncertainties for plausibility. For this purpose, a highly accurate reference is required. The most precise state estimate available in the CAV is the ego state obtained from the localization. Therefore, in the ego localization test, the ego state from the localization is used as reference to check the uncertainties in the external environment model for plausibility. Once the CAV reaches the FOV of the ITSS, the external perception has to represent the CAV by an object in its environment model. If the ego vehicle does not appear in the external environment model, this is a strong indication for a missing detection. Thus, such incidence is reported to the ego perception test (see Section 5.2.4), which drastically reduces its estimated reliability in such a case. If the ego vehicle is detected, like any other object in the environment model, the object corresponding to the CAV contains a state estimate and a covariance matrix describing its uncertainties. With these information, the Mahalanobis distance d_M [Mah36] between the ego localization and the reported state of the corresponding object is calculated. The smaller the Mahalanobis distance, the more likely is the reported covariance an upper bound on the true uncertainty. Hence, if $d_M < \theta_{M,loc}$ is below a threshold $\theta_{M,loc}$, e.g. $\theta_{M,loc} = 3$ as used in this section, the reported uncertainties are plausible. Thus, the estimated reliability is increased. In contrast, if d_M exceeds the threshold $\theta_{M,loc}$, the uncertainties are likely to be underestimated and thus are implausible. Therefore, the estimated reliability is decreased. Note that the ego localization test can only detect if the uncertainties are underestimated. This, however, is acceptable, since under-confident measurements lead to overly conservative behavior, which might slightly affect performance, but does not at all pose a safety threat. Mathematically, the ego localization

test is formulated as iteration over the time steps k by

$$\omega_{\text{loc}}(0) = ([0,0]^T, 1, [0.5,\ 0.5]^T) \tag{5.9a}$$

$$\omega_{\text{conf}}(k) = \left(\begin{bmatrix} \frac{1}{W+1} \\ 0 \end{bmatrix}, \frac{W}{W+1}, \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \right), \tag{5.9b}$$

$$\omega_{\text{under}}(k) = \left(\begin{bmatrix} 0 \\ \frac{c_{\text{under}}}{W+c_{\text{under}}} \end{bmatrix}, \frac{W}{W+c_{\text{under}}}, \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \right), \tag{5.9c}$$

$$\omega_{\text{input}}(k) = \begin{cases} \omega_{\text{conf}}(k) & \text{if } d_M < \theta_{M,\text{loc}}, \\ \omega_{\text{under}}(k) & \text{otherwise,} \end{cases} \tag{5.9d}$$

$$\omega_{\text{loc}}(k+1) = \omega_{\text{loc}}(k) \oplus \omega_{\text{input}}(k). \tag{5.9e}$$

The mechanism can be illustrated with the urn model from Definition 4.1. If $d_M < \theta_{M,\text{loc}}$, evidence for plausible uncertainties is obtained, whereas $d_M \geq \theta_{M,\text{loc}}$ indicates strong evidence for implausible uncertainties. Hence, for $d_M < \theta_{M,\text{loc}}$, a red ball is drawn from $\mathcal{U}_1$, while otherwise c_{under} black balls are drawn, where c_{under} is a weight. The drawn balls are jointly put into the second urn $\mathcal{U}_2$, which corresponds to applying the CBF operator. Thus, the estimated reliability slowly increases with plausible and rapidly decreases with implausible uncertainties.

5.2.4 Ego Perception Test

Apart from correctly reported uncertainties, the completeness of the external environment model is a crucial aspect for the reliability of the external environment model. For the external environment model to be complete, all objects within the covered area of the external environment model must be reliably detected. The ego localization test can detect incompleteness with respect to the CAV, but in order to achieve the required high accuracy, it is limited to plausibility estimates with respect to the ego vehicle. Hence, in the ego perception test, the ego vehicle's perception is used to assess the plausibility that the external environment model is complete. In essence, the ego perception test searches for missing detections in the external environment model. If the ego perception test finds such missing detections, the estimated reliability of the external environment model is strongly decreased. Conversely, if the ego perception test can associate the objects from the ego perception with objects from the external environment model, the estimated reliability is slightly increased.

The CAV can directly use the sensor detections instead of the objects from the multi-object tracking. This way, the ego perception information can be used, even if the multi-object tracking cannot collect sufficient measurements of the mutually perceived objects to set up a new track. However, while using detections rather than tracked objects reduces the response

time and makes more information available to the ego perception test. These improvements come at the cost of reduced reliability of the ego perception information. So, the test gets more sensitive at the cost of an increase in the false positive rate. This, however, can usually be tolerated, since the consequences of a false alarm are far less severe than the consequences of a false negative.

The ego perception test is designed such that whenever the ego perception can confirm an object that is reported by the external environment model, trust in the external environment model is increased, while trust is significantly decreased whenever the ego perception detects an object that cannot be associated with any reported objects. All other cases, where no object is detected by the ego perception, are ignored. In terms of SL, this can be formulated as the following iteration over the time steps k:

$$\omega_{\text{perc}}(0) = \left([0,0]^T, 1, [0.5,\ 0.5]^T \right) , \tag{5.10a}$$

$$\omega_{\text{conf}}(k) = \left(\begin{bmatrix} \frac{1}{W+1} \\ 0 \end{bmatrix}, \frac{W}{W+1}, \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \right) , \tag{5.10b}$$

$$\omega_{\text{mis}}(k) = \left(\begin{bmatrix} 0 \\ \frac{c_{\text{mis}}}{W+w_{\text{mis}}} \end{bmatrix}, \frac{W}{W+c_{\text{mis}}}, \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \right) , \tag{5.10c}$$

$$d_{\text{min},i} = \min_{\boldsymbol{x}_{\text{ext},j}} \left\{ d_E \left(\boldsymbol{x}_{\text{ego},i}, \boldsymbol{x}_{\text{ext},j} \right) \right\} , \tag{5.10d}$$

$$\omega_{\text{perc}}(k+1) = \begin{cases} \omega_{\text{perc}}(k) \oplus \omega_{\text{conf}}(k) & \text{if } \max_i \{d_{\text{min},i}\} < \theta_{\text{EPT}} , \\ \omega_{\text{perc}}(k) \oplus \omega_{\text{mis}}(k) & \text{otherwise,} \end{cases} , \tag{5.10e}$$

where ω_{conf} describes the opinion resulting from an object within the ego perception that confirms an object from the external environment model. The opinion ω_{mis} is resulting from a missing detection, i.e., when the ego perception detects an object that cannot be associated with an object from the external environment model. The weight c_{mis} controls the asymmetry in reliability increase and decrease. For each object state estimate $\boldsymbol{x}_{\text{ego},i}$ from the ego perception in the mutual FOV, the minimal distance $d_{\text{min},i}$ to the nearest neighbor in the object list of the external perception is determined. Here, $\boldsymbol{x}_{\text{ext},j}$ denotes the state estimates corresponding to the objects in the external environment model. If all ego objects can be associated successfully, the maximum association distance is below a threshold, i.e. $\max_i \{d_{\text{min},i}\} < \theta_{\text{EPT}}$. In this case, the CBF operator is used to merge ω_{conf} with $\omega_{\text{perc}}(k)$. This increases the estimated reliability. Otherwise, ω_{perc} is merged with $\omega_{\text{mis}}(k)$ to decrease the estimated reliability.

Speaking in terms of the urn model from Definition 4.1, a red ball is drawn from $\mathcal{U}_1$ whenever all objects are associated successfully. Otherwise, c_{mis} black balls are drawn from $\mathcal{U}_1$. Thus, the uncertainty decreases with each time step, where the observations provide evidence for either reliable or unreliable behavior of the external perception.

5.2.5 Overall Fusion

So far, four tests have been proposed, each of which analyzes a certain aspect of reliability. The prediction test (see Section 5.2.1) analyzes whether the state predictions that come with the external environment model are accurate or not. In the map test (see Section 5.2.2), the objects from the external environment model are checked to be consistent with the vehicle's map. Besides these consistency tests, the perception test uses the ego perception to find missing detections in the external environment model (see Section 5.2.4), and the ego localization test (see Section 5.2.3) checks the reported uncertainties for plausibility.

In a final fusion step, the detailed analysis of the various aspects of reliability are condensed to a total opinion ω_{total}. This ω_{total} reflects the overall reliability of the external environment model, using a final overall fusion step. Figuratively speaking, the overall fusion interprets the information gathered so far through the four tests. The interpretation of the gathered data, however, depends on how important the respective aspects are considered to be. Therefore, the fusion step has to incorporate importance weights that balance the test results against each other. Mathematically, this fusion step is formulated in the as

$$\omega_{\text{cons}} = \omega_{\text{pred}} \hat{\oplus} \omega_{\text{map}}\,, \tag{5.11a}$$

$$\omega_{\text{plaus}} = \omega_{\text{perc}} \hat{\oplus} \omega_{\text{loc}}\,, \tag{5.11b}$$

$$\omega_{\text{total}} = \omega_{\text{cons}} \underset{c_{\text{cons}}, c_{\text{plaus}}}{\tilde{\oplus}} \omega_{\text{plaus}} \tag{5.11c}$$

In (5.11), ω_{cons} describes the merged opinion on the consistency tests, i.e., the prediction test and the map test, while ω_{plaus} describes the fusion of the ego-perception-based plausibility tests, i.e., the ego perception test and the ego localization test. Both opinions ω_{cons} and ω_{plaus} are obtained using the uncertainty-weighted ABF operator. Finally, the opinions ω_{cons} and ω_{plaus} are merged to the overall result ω_{total} using the general weighted ABF operator and the weights c_{cons}, c_{plaus}. The weights are design parameters which balance the influence of the plausibility tests and the consistency tests.

To be used in the risk model of the motion planning (see Section 6.2.3), the total opinion ω_{total} has to be converted to a common probability

$$p_{\text{rel}}(\omega_{\text{total}}, \eta) = \int_{\eta}^{1} \beta(\boldsymbol{p}, \boldsymbol{r}_{\text{total}}, \boldsymbol{a}_{\text{total}}) \mathrm{d}\boldsymbol{p}\,, \tag{5.12}$$

which describes the likelihood that the external perception is reliable. In (5.12), η describes the minimum first order probability that the external perception is reliable. The integration over the β-distribution then yields the evidence-based confidence, i.e., a second order probability, that η is a lower bound for the reliability of the external perception.

5.2.6 Experimental Setup

The reliability estimation scheme is evaluated by the example of the application scenario (see Section 2.1), where the ITSS from Section 2.2 and the CAV as described in Section 2.3 are used. However, to provoke perception errors upon which the reliability estimation can be tested, only the LIDAR subsystem is used. This results in a reduced perceptual capability for the ITSS in the middle of the intersection, which is the mutual FOV between the CAV and the ITSS. Furthermore, using only the LIDAR subsystem reduces redundancy and thereby makes the ITSS less reliable.

The LIDAR data then is processed using the online variant of the box detection algorithm described in [MHS+19]. The key idea of the box detection comprises four steps: first, the ground points and further background points resulting from static obstacles in the scene are removed. Second, the remaining LIDAR points are clustered using the DBSCAN algorithm [EKSX+96] and the cluster centers are calculated. Third, the contour points of the clusters are extracted and partitioned into subsets, where each subset of points describe an edge of the bounding box around. In the fourth step, a least square regression is applied to the subsets, which yields length, width, and orientation of the object and thus the 2D bounding box.

Then, the detections from the LIDAR subsystem are tracked using a map-supported multi-instance Kalman filter approach with nearest neighbor association. First, the detection centers are associated with the lanes the respective road users are driving on. Next, the detection centers are projected onto the reference path that corresponds to the respective lane. In a further step, the existing tracks are predicted along their lanes using a 1D CV model and the new, projected detections are associated with the existing tracks using nearest neighbor association with a maximum association range. The existing tracks then are updated with the associated detections. Detections that cannot be associated are used to set up new tracks. If no projected detection can be associated with a track for multiple time steps, the track and its corresponding object are removed from the external environment model. Additionally, the Kalman filter approach is used to unimodally predict the trajectory of the respective objects. For that, the objects are predicted along their lanes using the CV model, since for long time horizons in urban areas, the lane information is very helpful and the CV model is a simple yet sufficient approximation for most cases.

The CAV then receives the external environment model and runs the reliability estimation based on the CAV's perception. Since for the application scenario the mutual FOV is small, the time window to confirm an object from the external environment model is usually narrow. Accordingly, to increase the statistical evidence for the perception test, the perception module was bypassed by a light-weight perception module that only uses the raw data of the CAV's IBEO LUX laserscanner. In the light-weight perception module, the raw data of the LIDAR is transformed to the world coordinate frame using the pose estimation from the CAV's ego localization and the calibration of the LIDAR sensor. In the next step, a region of interest

(ROI) mask is overlayed over the junction topology and all LIDAR points that are outside this ROI are removed. Furthermore, all LIDAR points with a z-component below a threshold are classified as ground points and thus excluded from the data. The remaining LIDAR points then are clustered using DBSCAN [EKSX+96]. From these clusters, the cluster centers are calculated and further on used as reference points for the ego perception objects.

Note that very basic algorithms are used to compute both the ego environment model and the external environment model. This way, it is shown that the proposed reliability estimation scheme is undemanding with respect to the ego environment model and the external environment model. Furthermore, using less sophisticated algorithms as compared to the tracking in the MEC server [HMSB19; HPS+20] or the tracking on-board the CAV [RVVD14] helps to provoke perception errors. This is important for the experiments to be meaningful in terms of the capabilities of the reliability estimation scheme.

5.2.7 Experimental Results

In this section, the proposed reliability estimation method is evaluated using the experimental setup described previously in Section 5.2.6. First, the behavior of the individual tests is studied through characteristic events such as missing detections for the perception test. Then, the overall mechanism is evaluated using 30 manually labeled sequences, which shows that the overall mechanism yields reasonable results and serves well as a discrimination method to decide whether to rely on the external environment model or not.

Figure 5.5 exemplarily visualizes the output of the prediction test (see Section 5.2.1). In the given example, the observed vehicle is decelerating, while the prediction assumes a constant velocity model. Hence, some predictions become inaccurate. This shows in an increased number of red dots as well as in the $\beta-$probability density function (PDF) peak at about 80 % which corresponds to a medium reliability. Such results are expected and indicate that the prediction test reacts as intended.

Figure 5.6 shows the object centers measured at the junction, which are used to set up the reference opinion for the map test. It can be seen that there is a high density of blue dots in the northern branch of the junction, confined to a very narrow band. Thus, it can be concluded that the probability gradient in east direction is big and therefore, a high sensitivity of the map test to object shifts in the east direction are expected. To evaluate the map test, the objects from $\mathcal{L}_{\text{ext}}$ are shifted by a constant offset in east direction simulating a calibration error or a static mapping error. Hence, it is expected that the estimated reliability of the map test will significantly drop if the observed objects are shifted. This effect can clearly be seen in Fig. 5.7, where an offset of only 50 cm already leads to a significant decrease in the estimated reliability. The estimated reliability drops further with increasing offset, until it

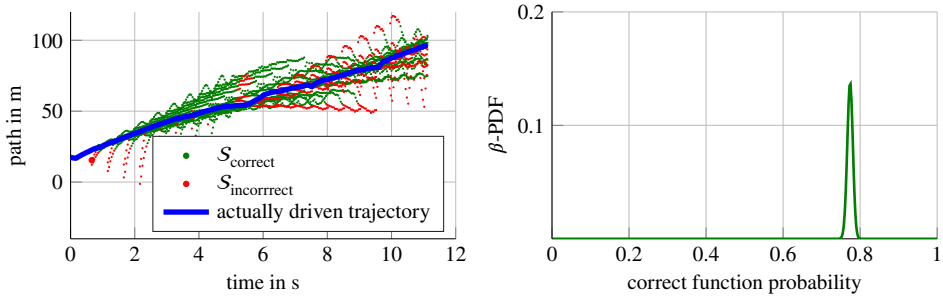


Figure 5.5: Prediction test: The left picture shows the predicted states tuples being classified into correct and incorrect. In the right picture, the resulting β-PDF is depicted (cf. [MGB19]).

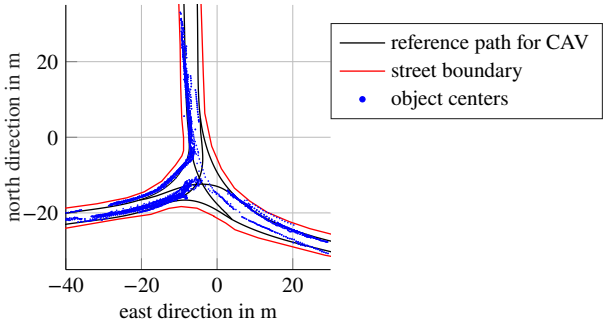


Figure 5.6: A map of the measured object centers. The centers are marked in blue ([MGB19], © 2019 IEEE).

reaches complete distrust at an east shift of 2.25 m, where the objects are reported to be on physically infeasible places.

Figure 5.8 shows an event of interest for the ego perception test as well as its reaction to this situation. As can be seen from Fig. 5.8(b) and (c), the CAV's perception found a cyclist the ITSS missed to detect, which leads to a massive drop of the estimated reliability. This can be seen from Fig. 5.8(b), where the estimated reliability drops from about 100 % to only about 50 %. The effect is even more strongly visible in the $\beta-$PDF, as can be seen when comparing the $\beta-$PDF before and after the event, as illustrated in Fig. 5.8(a). Moreover, Fig. 5.8(b) shows that the reliability slowly recovers after the event as the following measurements of the ITSS again are consistent with the perception of the CAV. The sudden drop of reliability as well as the slow recovery thereof after the missed detection show exactly the expected behavior and thus confirm that the test works.

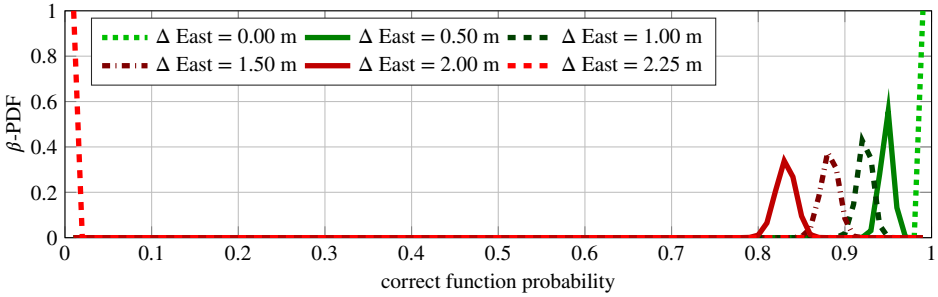


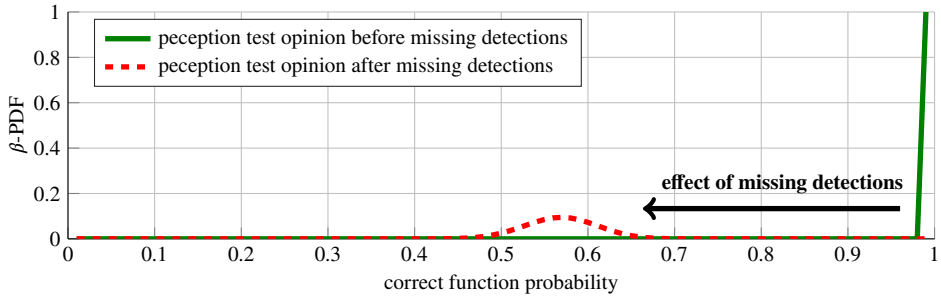
Figure 5.7: Map test opinion after shifting the RSU objects in east direction ([MGB19], © 2019 IEEE).

To illustrate the function of the ego localization test, a situation is chosen where the tracking accuracy is overestimated by the ITSS. For this situation, the CAV enters a poorly covered area at about $t = 21.75$ s, which leads to overconfident estimations and hence a drop in the estimated reliability, while the CAV leaves the FOV completely at about $t = 24$ s, thus the estimated reliability remains unchanged afterwards. It can be seen from Fig. 5.9 that the estimated reliability according to the test significantly drops as the test detects that the distance between the reported position of the CAV and the position of the CAV according to the ego localization differ by more than $3\sigma_{\text{total}}$. Note, however, that the estimated reliability drops less quickly than with the ego perception test. This is due to the fact that the weight w_{under} is chosen smaller than w_{mis}.

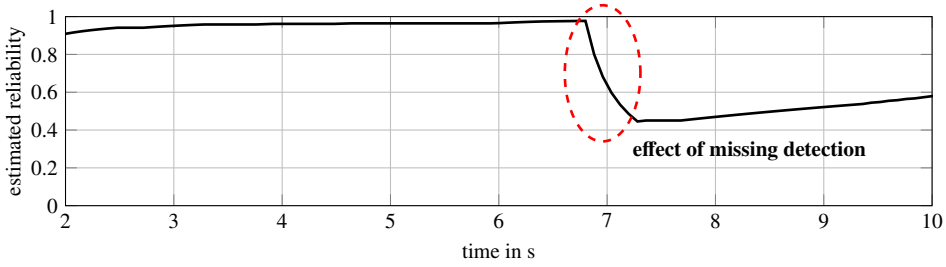
Finally, the overall result is evaluated using 30 manually labeled sequences. Figure 5.10 shows the result of the reliability estimation for the *good* sequences, marked in solid green, as well as the *bad* sequences, marked in dashed red. For this labeling, additional context information, such as the video stream of the CAV's front camera, was used. The curves show the probabilities $p_{\text{rel}}(\omega_{\text{total}}, \eta)$ corresponding to the SL opinions of the overall reliability estimation at the end of the sequences. It can be seen that the *bad* sequences can easily be distinguished from the *good* ones using the output of the reliability estimation as a criterion.

It shows that for the *good* sequences, the integral over the $\beta-$PDF from 90 % to 100 % is always above 90 %. This means that there is a 90 % confidence level supporting the hypothesis that the correct function probability is at least 90 %. In contrast, for the *bad* sequences, the confidence supporting the hypothesis that the actual correct function probability is more then 75 % is always below 1.5 %.

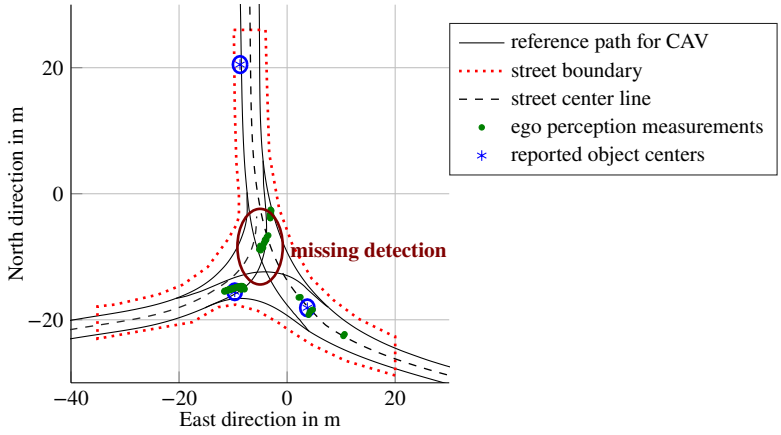
While Fig. 5.10 shows the reliability for a single moment of evaluation, i.e., at the end of the sequence, in Fig. 5.11, the temporal evolution of $p_{\text{rel}}(\omega_{\text{total}}, 0.9)$ is shown by the example of eight typical representatives. It has to be noted that $p_{\text{rel}}(\omega_{\text{total}}, 0.9) = 1$ has been observed



(a) Perception test opinion mapped to a $\beta-$PDF before and after missing detection ([MGB19], © 2019 IEEE).



(b) Estimated reliability ([MGB19], © 2019 IEEE).



(c) The bicycle marked in red is not detected by the RSU (cf. [MGB19]).

Figure 5.8: Ego perception test. At 6.8 s the perception of the CAV detects a bicycle that is not reported by the RSU. Thus, the reliability estimated by the perception test is significantly decreased and recovers slowly afterwards (cf. [MGB19]).

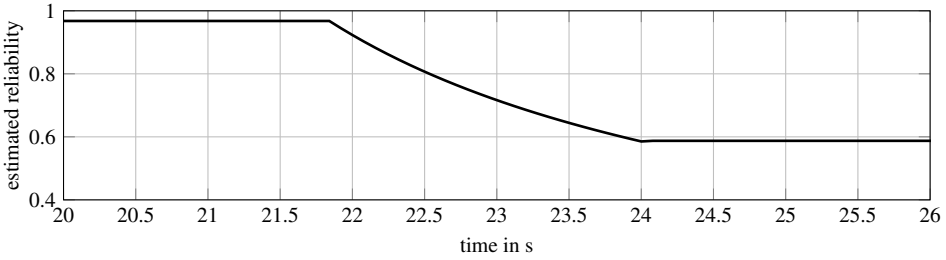


Figure 5.9: Estimated reliability according to the ego localization test (cf. [MGB19]).

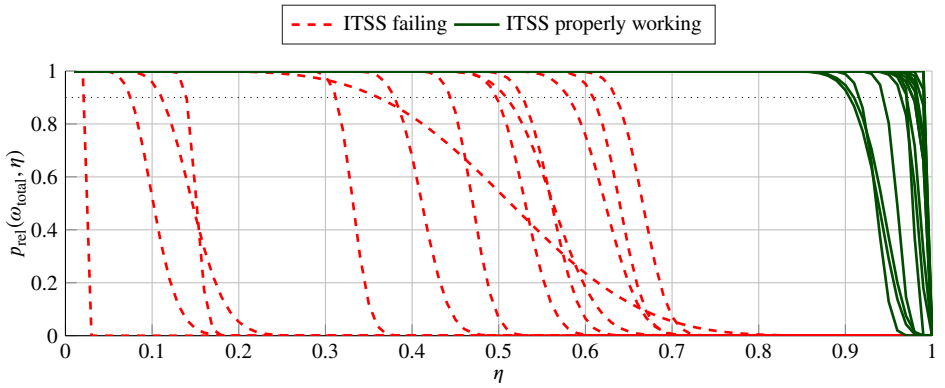


Figure 5.10: p_{rel} of 30 overall reliability estimates ω^{total} on real world sequences. η describes the minimum required reliability of the ITSS (cf. [MB21]).

most frequently for the *good* sequences. The sequences have been normalized to start with $t = 0$ when the CAV enters the FOV of the external perception. As can be seen from Fig. 5.11, if the external perception works well, the estimated reliability quickly converges to one, i.e. $p_{\text{rel}} \to 1$. Hence, in these cases, the influence on the reliability estimation on the risk model of the motion planning can be neglected about 0.5 s after the CAV has entered the FOV of the external perception. Therefore, the experiments provide good indication that the proposed reliability estimation interferes negligibly with the motion planning if the external perception works as expected. Accordingly, in such cases, the proposed motion planning scheme (see Chapter 6) with activated reliability estimation is expected to operate as if $p_{\text{rel}} = 1$ was chosen. In contrast, if the external perception is not reliable, the reliability may increase and reach high levels at first, but massively drops as soon as an inconsistency or implausible reporting is detected.

To evaluate the reliability of the reliability estimation scheme, following the author's argumentation from [MB21], the results from Fig. 5.10 are used in combination with the

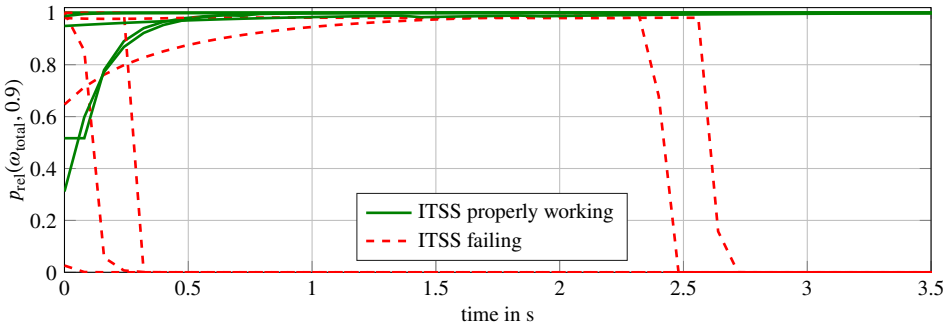


Figure 5.11: Typical time courses of the second order reliability $p_{\mathrm{rel}}(\omega_{\mathrm{total}}, 0.9)$.

proposed urn model from Definition 4.1 (see Chapter 4). For that, each *good* sequence with $p_{\mathrm{rel}}(\omega_{\mathrm{total}}, 0.9) \geq 0.9$ and each *bad* sequence with $p_{\mathrm{rel}}(\omega_{\mathrm{total}}, 0.8) \leq 0.01$ is represented as red ball, while all other sequences are represented as black balls. Hence, $W = \mathrm{card}\big\{\{\bullet, \bullet\}\big\} = 2$. Furthermore, the prior $\boldsymbol{a} = [0.5,\ 0.5]^T$ is assumed, which indicates no prior knowledge. As can be seen from Fig. 5.10, all 30 sequences fulfill the requirements, so $\mathcal{U}_2$ contains 30 red balls and $W = 2$ checkered balls. With Theorem 4.1, this corresponds to the Dirichlet distribution $\mathrm{Dir}(\boldsymbol{p}_A, [30, 0]^T, [0.5, 0.5]^T)$. For this distribution, the p-value $p(\eta) \geq 90\ \%$ is calculated with reference to a $\eta = 95\ \%$ confidence level. In total, these findings demonstrate by experiment that the overall reliability estimation algorithm works as expected and is useful to decide whether to rely on the cooperative information received from the ITSS.

5.3 Reliability and Trust in a Vehicular Multi-Agent System

So far, it has been shown experimentally by the example of the application scenario that the estimation scheme from Chapter 5.2 can online estimate the reliability of an external environment model. In the next step, based on the author's pre-publication [MMSB19], the reliability estimation scheme is generalized such that the reliability of multiple external environment models shared among different agents in a V-MAS can be estimated. Furthermore, the reliability estimates of an environment model gathered by multiple agents from multiple traffic scenarios are aggregated to build trust between the agents. This way, the estimation quality can be further improved. Moreover, the agents can use the feedback of other trusted agents to detect their own malfunctions and correct them. For example, if the sensor calibration of an ITSS deteriorates due to heavy weather, the ITSS can use the feedback of trusted CAVs to detect the error and recalibrate itself automatically. However,

when extending the reliability estimation to multiple environment models, there is no longer a global reference and the question arises of how the vMAS can be protected against misbehaving agents. Here, misbehaving agents are agents that intentionally or unintentionally share incorrect environment models. This issue is addressed by an SL-based misbehavior detection mechanism that identifies misbehaving agents and thus allows to revise their trust level or, in the extreme case, ban them from the system. Since this issue is security-related, an attacker model is formulated and the system design as well as the trade-offs between protection level and security overhead are discussed. The generalized reliability estimation scheme is first evaluated simulatively for the application scenario. Then, a large-scale traffic simulation in the urban area of Cologne is performed to study the scaling properties of the generalized reliability estimation. Thus, it is shown that the proposed approach, in contrast to many state-of-the-art trust management systems [HDLK18], scales well even to large V-MASs.

5.3.1 System Description and Trade-Off Considerations

In general, there are many ways to establish reliability and trust in a V-MAS (see e.g. the survey [HDLK18]). In this thesis, one particular system is considered, for which the underlying assumptions and trade-off considerations are discussed. Basically, the V-MAS comprises four types of agents: providers that share their perception, users that receive shared perception information and operate upon them, a broker that routes the information through, and hidden observers used to unmask collaborative attacks. To share perception information, e.g, CAM, cooperative perception messages (CPMs), or decentralized environmental notification messages (DENMs) can be used [ETS14; ETS19]. For the network topology, push-based cellular vehicle to anything (C-V2X) communication is assumed, since C-V2X is commonly used for connected intelligent transportation systems in urban areas [HDLK18] and since push-based C-V2X communication is successfully applied to the application scenario. To reduce bandwidth consumption, the V-MAS is divided into local subsystems to which the agents must subscribe. The broker then forwards the information from the providers to the users, as needed. This has two beneficial side effects. First, the broker serves as an abstraction layer between the agents so that a high level of privacy is obtained without spending much efforts. Second, it disguises the number of attackers required for a successful collaborative attack. Since pseudonymization usually makes a V-MAS vulnerable to so-called Sybil attacks and produces communication overhead [HDLK18], full pseudonymity is not considered. Rather, the agents subscribe to the broker with their clear names, e.g., in the form of a electronic license plate, while the broker assigns a temporary identifier for each road user. Thus, the agents need to trust the broker, but none of the other agents. This appears as a good trade-off, since protecting against a malicious broker, i.e. network operator, would require uneconomic protection efforts, anyway.

A Public Key Infrastructure (PKI) is used to guarantee authenticity of the information. Hence,

the security is significantly increased at the cost of some message overhead. Such PKI can be implemented, e.g., according to the ETSI TS 103 097 standard [17].

To harden the system with respect to misbehaving agents, the users watch out for inconsistencies within the incoming data. If the data is consistent, a positive feedback is sent to the broker to build trust between the involved agents. Otherwise, the incident is reported to the broker. The broker then decides in a court-case-like misbehavior detection procedure based on the available evidence whether the blamed agent actually was misbehaving or not. If so, the trust in the misbehaving agent is revised and, in severe cases, the broker even revokes the certificates of the misbehaving agent, which leads to a permanent exclusion of the agent from the system. Otherwise, the broker checks whether there is sufficient evidence indicating that the reporting agent intentionally stated a wrong accusation. In such a case, the broker treats the wrongly accusing agent as a misbehaving agent and revises its trust or revokes its certificate. If the evidence does not clearly indicate either misbehavior or false blame, the opinion of a highly trusted agent such as an ITSS may break the tie. However, in cases where no highly trusted agent is involved, the case is dropped and none of the agents' trust is revised. Since no ground truth is available, the misbehavior detection mechanism generally requires a local honest majority, i.e. that a majority vote would unmask a misbehaving agent. Thus, without further measures, the V-MAS would be vulnerable to collaborative attacks, where multiple malicious agents coordinate their attack through a side-channel to overrule a local honest majority. For this reason, the broker operates hidden observers, which secretly report highly trusted observations to the broker. These observations allow the broker to uncover collaborative attacks. However, hidden observers are generally expensive to operate and require to be highly trustworthy, which poses an additional security threat. Conversely, even a small number of hidden observers can greatly increase the chance of detecting a joint attack. Hence, operating a few hidden observers appears reasonable.

The communication procedure consists of the following steps: Both, providers and users, register at the broker and specify which type of information they offer or request for which region and for which period of time. Then, the broker verifies the identities of the involved agents and checks, whether the offered data matches the requested data. If so, the broker routes the data through. Then, the users check the incoming data for plausibility. If the data is plausible, the users act upon it and trigger the trust building (see Section 5.3.3) at the broker. In contrast, if the data is implausible, the users report the incidence to the broker. Such incidence report comprises an opinion on the incidence, measured evidence that support the opinion, and the respective timestamps. The opinion on the incidence can be either binomial or multinomial. For example, if the ego perception test from Section 5.2.4 is used for plausibility checking, an incidence report would comprise the missing detections measured by the reporting agent, the timestamps, the object lists received from the blamed provider, and a binomial opinion counting the number of missing detections referred to the total number of measurements in this time period. A multinomial opinion can be used, e.g., to approximate a noise distribution. This case is discussed in detail in Section 5.3.5. If an incidence is reported, the broker determines all involved agents, temporarily revokes their

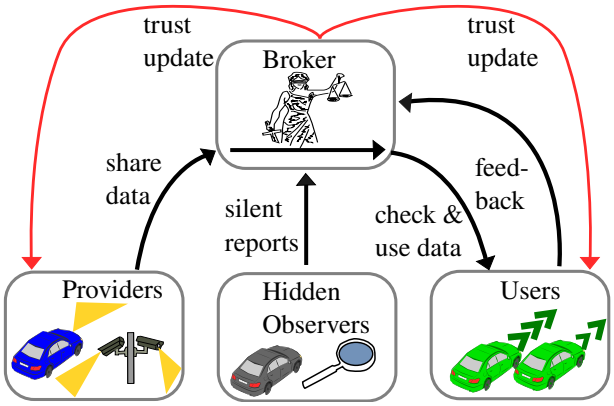


Figure 5.12: Illustration of the described V-MAS.

certificates, and requests an opinion on the incidence. The agents provide this opinion based on their perception and the broker performs the misbehavior detection from Section 5.3.4. A system overview is presented in Fig. 5.12.

5.3.2 Attacker Model

Since in the field of security it is never possible to guarantee absolute security, an attacker model has to be specified, describing what kind of attacks have been anticipated during the design of the security system. Until now, in literature, there is no universally accepted attacker model that is used consistently for cooperative intelligent transportation systemss (C-ITSs) [HDLK18]. Furthermore, [HDLK18] also states that the classical Dolev-Yao attacker model, which is widely used in cryptography to prove the security of communication protocols, is too strong to be reasonably applied to C-ITS. Therefore, and due to the fact that the focus of this thesis lies on using cooperative information in a CAV safely rather than protecting a CAV against sophisticated attacks, the following simple attacker model is assumed:

- Attackers can send syntactically correct, yet semantically wrong data (sensor fault).
- Attackers might use several pseudonyms in parallel (known as Sybil attack) [HDLK18].
- Attackers must be agents. Hence, their physical presence is limited, while they might have extended computational power.
- The broker and the hidden observers are fully trustworthy and honest. Attackers neither infer the number of users (i.e. listening agents) nor spot hidden observers.

Note that despite its simplicity, the stated attacker model covers most of the state-of-the-art assumptions [HDLK18] and is sufficient for most practical situations.

5.3.3 Trust Building

Whenever a communication is successfully completed, the involved agents are rewarded by increasing the trust into them. This corresponds with the following intuition: An agent that has completed communications successfully multiple times will provide reliable information the next time as well. Furthermore, intuitively, it is unlikely that an attacker will actively help to identify other attackers. Thus, agents that correctly report misbehaving agents are strongly rewarded, i.e., the normal reward is weighted with the weight c_{TR}.

To account for the fact that successful conversations with the same agents over and over again are not statistically independent, the idea behind the consensus operator for partially dependent β-PDFs from SL [JMP06] is adopted for the regular trust building. The idea of the operator is to split the available evidence into a statistically independent and a statistically dependent part. Then, the dependent part is merged with the ABF operator and the result is merged with the independent part using the CBF operator:

$$\omega_A \leftarrow (\omega_{A,\mathrm{dep}} \underline{\oplus} \omega_{B,\mathrm{dep}}) \oplus \omega_{A,\mathrm{ind}} \oplus \omega_{B,\mathrm{ind}}\,. \tag{5.13}$$

If the agent is found honest in a misbehavior detection process, the CBF operator is used to increase the trust in the respective agent:

$$\omega_A \leftarrow \omega_A \oplus \omega_{\mathrm{TR}}, \quad \omega_{\mathrm{TR}} = \left(\left[\frac{c_{\mathrm{TR}}}{c_{\mathrm{TR}} + W}, 0 \right]^T, \frac{2}{c_{\mathrm{TR}} + W}, [0.5, 0.5]^T \right) \Bigg|_{W=2}. \tag{5.14}$$

Finally, to model trust fading over time, the trust of the agent is discounted with p_{aging}, i.e.

$$\omega_A(t_{k+1}) \leftarrow \mathcal{D}(\omega_A(t_k), p_{\mathrm{aging}})\,. \tag{5.15}$$

5.3.4 Misbehavior Detection and Trust Revision

While receiving data from other agents, the users continuously check the incoming data for consistency. This is mainly for two reasons: first, to protect themselves from being misled into a dangerous situation and, second, to warn others by reporting inconsistencies to the broker who then takes the following steps:

1. **Trust Discounting:** The opinion on the data is discounted by the age of the data and the spatial distance of the data source to the observation, since most perception systems not only suffer from an increased measurement uncertainty, but also degrade in reliability when measuring distant objects. This is reflected in the discount probability $p_{\text{age,loc}}$, which can be calculated as described in [MLN+19]. Furthermore, the reliability of the data is discounted by the reliability trust of the data source $p_{A_i,\text{trust}}$. Mathematically, this is expressed with the trust discounting operator from SL (see Definition 3.4).

$$\omega_{A_i} \leftarrow \mathcal{D}(\omega_{A_i},\, p_{\text{age,loc}}\, p_{A_i,\text{trust}})\,. \tag{5.16}$$

2. **Clustering:** The trust discounted opinions are clustered using the pair-wise projected distance from SL [Jøs16, p. 80]:

$$PD_{A_i}^{A_j} = \frac{1}{2}\sum_{l=1}^{W} |P_{\perp,A_j,l} - P_{\perp,A_i,l}|\,. \tag{5.17}$$

Each cluster corresponds to a hypothesis about the reported incidence, where each opinion in it represents an agent supporting it. The set of opinions $\mathcal{S}_{\omega,A}$ and the set of pair-wise projected distances $\mathcal{S}_{\text{PD}} = \{PD_{A_i}^{A_j} | PD_{A_i}^{A_j} < \theta_{\text{PD}}\}$, which do not exceed a threshold, then form a graph $\mathcal{G}(\mathcal{S}_{\omega,A}, \mathcal{S}_{\text{PD}})$. This graph $\mathcal{G}(\mathcal{S}_{\omega,A}, \mathcal{S}_{\text{PD}})$ can be subdivided into a set of connected subgraphs $\mathcal{S}_{\mathcal{G},\text{sub}}$, where each subgraph is not connected to any other subgraph. To ensure a certain density of the subgraphs, a minimum number of edges can be required to add a node to a subgraph. This reflects the idea behind DBSCAN [EKSX+96]. For each of the subgraphs, a reference opinion $\omega_{\text{ref},j}$ is calculated using the ABF operator:

$$\omega_{\text{ref},j} = \bigoplus_{\forall \omega_{A_i,\text{sub}} \in \mathcal{S}_{\omega,A,\text{sub}}} \omega_{A_i,\text{sub}}\,, \quad \mathcal{G}(\mathcal{S}_{\omega,A,\text{sub}}, \mathcal{S}_{\text{DC,pair,sub}}) \in \mathcal{S}_{\mathcal{G},\text{sub}}. \tag{5.18}$$

Then, the clusters are calculated as the set of opinions which have a maximum DC of $\theta_{\text{DC}}^{\text{ref}}$ with the reference opinion, i.e. $\mathcal{S}_{\omega}^{\text{ref},i} = \{\omega_{A_j} | DC(\omega_{A_j}, \omega_{\text{ref},i}) \leq \theta_{DC}^{\text{ref}}\}$. If two clusters are identical, i.e. $\mathcal{S}_{\omega}^{\text{ref},i} = \mathcal{S}_{\omega}^{\text{ref},j}$, the second cluster is pruned and the reference opinion is recalculated as $\omega_{\text{ref},i} \leftarrow \omega_{\text{ref},i} \,\underline{\oplus}\, \omega_{\text{ref},j}$.

3. **Selection of the reference:** From the clusters, the reference opinion ω_{ref}^{*} reflecting the assumed truth is determined as follows. If at least one hidden observer has been present in the scenario, i.e. $\mathcal{S}_{\text{ho}} \neq \emptyset$, the ABF result of the hidden observer opinions is chosen as the reference. Otherwise, the reference opinion of the cluster with the highest support is chosen, where highly trusted agents, such as an ITSS, have 1.5 votes

in case of a voting tie. Mathematically, this is expressed as

$$\omega^*_{\text{ref}} = \begin{cases} \bigoplus\limits_{\forall \omega_{\text{ho}} \in \mathcal{S}_{\text{ho}}} \omega_{\text{ho}}, & \text{if } \mathcal{S}_{\text{ho}} \neq \emptyset, \\ \omega_{\text{ref},j} | j = \arg\max_j \left\{ \text{card}\{ \mathcal{S}_{\omega,A,\text{sub},j} \} \right\} & \text{otherwise} \end{cases}. \tag{5.19}$$

If the reference opinion is not unique, because more than one subgraph has the most support, i.e. the biggest $\text{card}\{ \mathcal{S}_{\omega,A,\text{sub},j} \}$, the broker cannot decide which group of agents is right. In this case the misbehavior detection is dropped.

4. **Behaviour classification:** The DC between the reference opinion and the opinion of each agent is used to classify the agents into honest and misbehaving. If the DC is below a classification threshold $\theta^{\text{clas}}_{DC}$, the agent is considered honest. Otherwise, it is considered misbehaving. This leads to the partitioning of involved agents

$$\mathcal{S}_{\text{honest}} = \left\{ \omega_{A_j} \in \mathcal{S}_{\omega,A} : DC(\omega_{A_j}, \omega^*_{\text{ref}}) < \theta^{\text{clas}}_{DC} \right\}, \tag{5.20a}$$

$$\mathcal{S}_{\text{misb}} = \left\{ \omega_{A_j} \in \mathcal{S}_{\omega,A} : DC(\omega_{A_j}, \omega^*_{\text{ref}}) \geq \theta^{\text{clas}}_{DC} \right\}. \tag{5.20b}$$

5. **Trust revision:** The final step then corresponds to the sentence spoken by the judge in a court case. A simple way of punishing the agents found guilty of misbehaving would be to directly ban them from the system. This policy, however, would be too harsh assuming that even honest agents would occasionally provide wrong data, e.g. owing to temporary sensor faults or other unfortunate circumstances. Therefore, the trust revision mechanism from [JZW17] is used, i.e.,

$$AC\left(\mathcal{S}_{\omega,A}\right) = \frac{1}{\text{card}\left\{\mathcal{S}_{\omega,A}\right\}} \sum_{j=1}^{\text{card}\left\{\mathcal{S}_{\omega,A}\right\}} DC(\omega_{A_j}, \omega^*_{\text{ref}}), \tag{5.21a}$$

$$RF(A_i) = \frac{\max\limits_{\forall \omega_{A_j} \in \mathcal{S}_{\omega,A}} \left\{ DC(\omega_{A_j}, \omega^*_{\text{ref}}) \right\} \left(DC(\omega_{A_i}, \omega^*_{\text{ref}}) - AC\left(\mathcal{S}_{\omega,A}\right) \right)}{\max\limits_{\forall \omega_{A_j} \in \mathcal{S}_{\omega,A}} \left\{ DC(\omega_{A_j}, \omega^*_{\text{ref}}) \right\} - AC\left(\mathcal{S}_{\omega,A}\right)}, \tag{5.21b}$$

$$\tilde{\omega}_{A_i} = \begin{cases} \tilde{b}_{A_i,1} = (1 - RF(A_i))\, b_{A_i,1} \\ \tilde{b}_{A_i,2} = b_{A_i,2} + RF(A_i)(1 - b_{A_i,2}) \\ \tilde{u}_{A_i} = \left(1 - RF(A_i)\right) u_{A_i} \\ \tilde{\boldsymbol{a}}_{A_i} = \boldsymbol{a}_{A_i} \end{cases}, \tag{5.21c}$$

is used. Here, $\omega_{A_i} \in \mathcal{S}_{\text{misb}}$ are the agents classified as misbehaving, $AC\left(\mathcal{S}_{\omega,A}\right)$ is the

average conflict, $RF(A_i)$ are the revision weights, and $\tilde{\omega}_{A_i}$ is the respective revised trust in agent A_i.

The trust revision is designed such that it rapidly decreases trust once a misbehavior is detected, while it takes comparatively long to build up trust in the trust building step (see Section 5.3.3). The intuition behind this asymmetry is that trusting in a misbehaving agent is much worse than not trusting in a honest agent. Furthermore, an agent that has always been trustworthy can change to a misbehaving agent instantaneously, e.g., due to a sudden sensor fault or because the agent has been compromised by an attacker, while it needs a sufficient number of samples to be able to guarantee a certain statistical confidence.

5.3.5 Simulations of the Generalized Reliability Estimation Scheme for the Application Scenario

The proposed trust management system is not restricted to use binomial opinions like the opinion obtained from the ego perception test (see Section 5.2.4), but also operates on multinomial opinions. Hence, the proposed method can also be applied for discretized continuous problems like estimating a distribution of measurement noise. To illustrate this, consider again the application scenario as depicted in Fig. 2.1 (see Section 2.1), where the green vehicle is approaching the occluded yield junction. At the same time, the blue vehicle follows the red vehicle on the main road coming from left, having the right of way. The ITSS has the whole junction under surveillance. All four agents measure the position of the red vehicle at the same time. For simplicity and to focus on the main effects, only the 1D position s of the red vehicle on its lane is considered. Furthermore, it is assumed that the motion of the red vehicle is already compensated for by a feed-forward control. Hence, $s = 0$ means that the red vehicle is measured at its predicted position. For the measurement of the red vehicle's position, Gaussian noise $\mathcal{N}(0.25\text{ m}, (0.75\text{ m})^2)$ with a mean of $\mu = 0.25$ m and a standard deviation of $\sigma = 0.75$ m is added. The agents build their SL opinion on the position of the red vehicle processing the following steps.

1. The measurements s_i are normalized using the transform $z_i = \frac{s_i - \hat{\mu}}{\hat{\sigma}}$. Here, $\hat{\mu}$ and $\hat{\sigma}$ are estimates of the mean and standard deviation, respectively. Thus, for perfectly known parameters $\hat{\mu} = \mu^*$ and $\hat{\sigma} = \sigma^*$, the measurements would be transformed to a standard normal distribution $\{z_i\} \sim \mathcal{N}(Z|0, 1)$.

2. The distribution of the measurements $\{z_i\}$ is estimated using a histogram. The histogram is mapped to an SL opinion using the bijective mapping (3.5). The respective opinions are published.

If one of the agents reports inconsistency, the broker performs the misbehavior detection. For the simulations, a measurement frequency of 10 Hz is assumed. Moreover, in the simulations,

the blue vehicle, the red vehicle itself, and the ITSS observe the position of the red vehicle for 5 s. Hence, their opinion is based on 50 samples. Due to the occlusion, the green vehicle spots the red one only for 1 s. For this simulation setup, the following scenarios are evaluated:

1. All agents behave correctly or, in terms of security, are honest. This scenario acts as the baseline to subsequently choose a reasonable detection sensitivity.

2. The red and the blue vehicle conspire against the ITSS and launch a cooperative attack trying to discredit the ITSS. Therefore, both of them report measurements according to $z_i \sim \mathcal{N}(1.00\text{ m}, (0.75\text{ m})^2)$ rather than reporting the samples they measure, which are $\mathcal{N}(0.25\text{ m}, (0.75\text{ m})^2)$ distributed. However, the attackers are unaware that the ITSS is a highly trusted agent and that a further user, the green vehicle, confirms the opinion of the ITSS . Hence, whenever the data of the green vehicle and the ITSS are consistent, the opinions form a cluster with the same support as the cluster of collaborative attackers. The tie-braking property of the ITSS unmasks the attack.

3. The ITSS is faulty and thus sends semantically wrong measurements of the red vehicle. Hence, $\hat{\sigma}_{ITSS} \neq \sigma^*$, $\hat{\mu}_{ITSS} \neq \mu^*$ holds. For the scenario, the ITSS gets caught by the misbehavior detection mechanism of the broker. The broker then revises the trust of the ITSS and sends the respective reference opinions from the other road users. The ITSS uses the reference to correct its fault by reidentifying $\hat{\mu}_{ITSS}$ and $\hat{\sigma}_{ITSS}$.

In Fig. 5.13, scenarios 1 and 2 are evaluated in dependence of the threshold parameter θ_{DC}. An attacker is detected, if its opinion is classified as misbehaving, i.e. $\omega_{A_{\text{misb}}} \in \mathcal{S}_{\text{misb}}$. It can be seen from Fig. 5.13 that $\theta_{DC} \in [0.15, 0.20]$ is an adequate choice: for $\theta_{DC} < 0.15$ the false alarm rate is too high as statistically insignificant inconsistencies would be considered a threat. For $\theta_{DC} < 0.1$, evaluating the detection rate does not even make sense, since the results becomes just random due to the oversensitivity. Conversely, for $\theta_{DC} > 0.20$, the misbehavior detection gets insensitive. For $\theta_{DC} \in [0.15, 0.20]$, it can be seen that the detection probability, even for a collaborative attack, is at about 70 %, while the probability of identifying at least one of the attackers is even higher, at about 80 %. The probability for the attackers to successfully launch their cooperative attack is between 5.7 % and 11.7 %. This illustrates that the trust management and misbehavior detection mechanism can cope even with collaborative attacks.

Scenario 3 is evaluated by determining the receiver operator characteristic (ROC) curves of the mechanism in dependence of $\hat{\mu}$ and $\hat{\sigma}$. The closer an ROC curve gets to the ideal point $(0, 1)$, the better the system works. The ROC curves are depicted in Fig. 5.14. It can be seen that even estimation errors below the standard deviation can be detected efficiently at a low false positive rate. Moreover, it can be seen that the misbehavior detection performs moderately for slightly overconfident measurements, i.e. $\hat{\sigma} = 0.5$ m, but can detect more overconfident measurements, i.e. $\hat{\sigma} = 0.25$ m, with high efficiency.

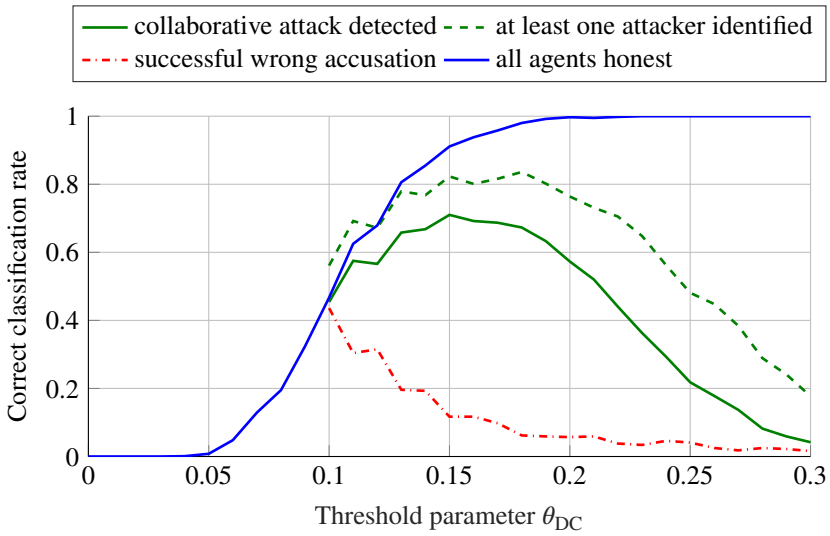


Figure 5.13: Classification rate as a function of the threshold θ_{DC} in the collaborative attack scenario ([MMSB19], © 2019 IEEE).

If the ITSS is classified as misbehaving, this triggers an automatic recalibration. For the simulation example, the ITSS recalibrates itself according to

$$\hat{\mu} = \overline{\mathbf{z}}^{\top}\mathbf{P}_{\text{ITSS}} - \overline{\mathbf{z}}^{\top}\mathbf{P}_{\text{ref}}\,, \qquad \hat{\sigma} = \sqrt{\sum_{\forall i}((\overline{z}_i - \hat{\mu})P_{\text{ITSS},i})^2}\,, \tag{5.22}$$

using its own faulty opinion as well as the reference opinion provided by the broker. In (5.22), $\overline{\mathbf{z}}$ describes the centers of the histogram bins. The recalibration was evaluated over 1000 runs testing with $\theta_{DC} = 0.15$. In average, $\hat{\mu} = 0.250$ m and $\hat{\sigma} = 0.765$ m resulted. In practice, the automatic self-calibration scheme [MHS+19] can be used. Thus, it is shown that the proposed approach can not only detect faults, but can also correct them if the misbehaving agent is faulty but not malicious.

5.3.6 Simulation Study on Scaling the Approach by the Example of Traffic Jam Detection

In this section, the scaling properties of the approach are examined. Therefore, the scenario of a single junction with only a few agents is scaled up to a huge V-MAS covering the whole traffic of Cologne. To avoid research bias, the experimental design and the simulation

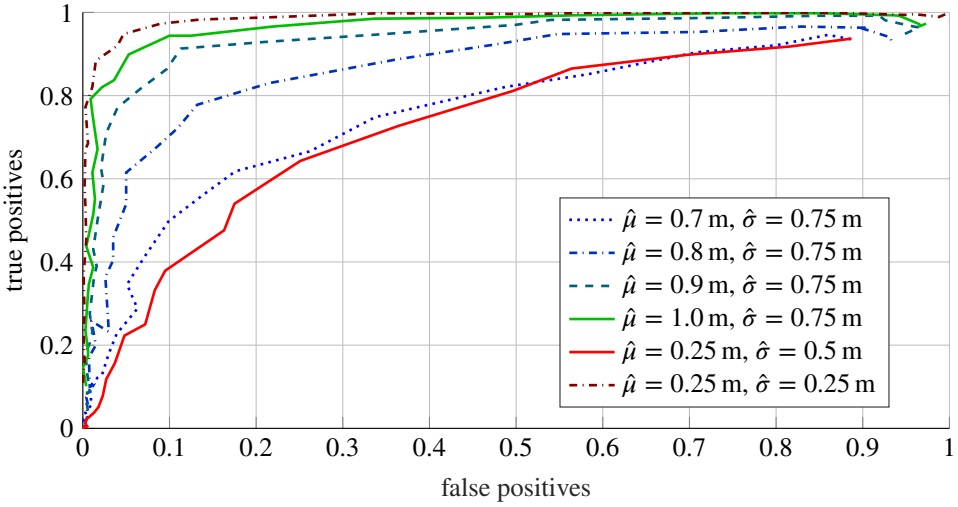
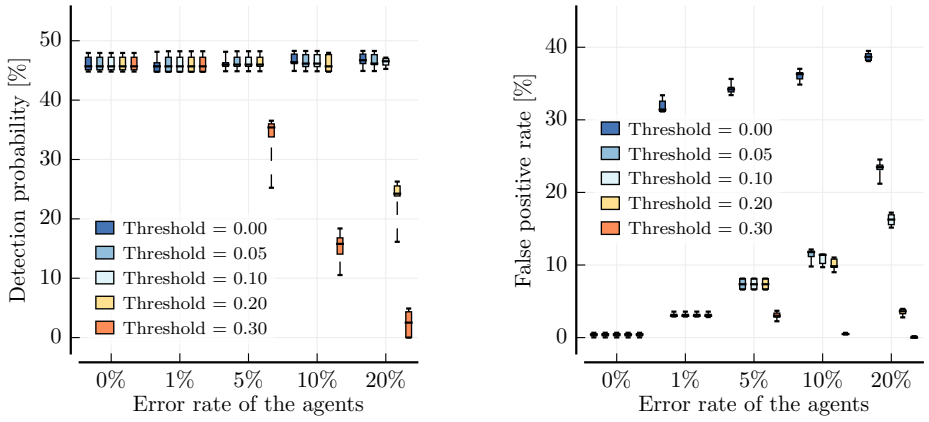


Figure 5.14: ROC curve for the misbehaving RSU scenario (scenario 3) ([MMSB19], © 2019 IEEE).

itself was split: while the author of this thesis designed the simulations and provided an implementation of the mechanism described in this chapter, Tobias Meuser integrated the mechanism into the Simonstrator [MBSR19] simulation framework that is based on SUMO [KEBB12], carried out the simulations and provided an uninterpreted summary of the results. The results have been published in [MMSB19].

In the scenario, 10 % of the agents are assumed to be attackers, i.e. misbehaving. To show that the approach is not restricted to CPM, for the large scale scenario, DENM-based traffic jam detection is evaluated. The algorithm is run to detect the misbehaving agents. The binary conclusion of the algorithm, i.e., misbehaving or honest as it results after the behavior classification step, is compared to the ground truth, i.e., whether the classified agent actually is honest or misbehaving according to the simulator. Therefrom, a statistic is created covering over all communications that occur during the simulation and the results are referred to the overall number of communications. This yields an empirical detection probability p_{tp} as well as an empirical false positive rate p_{fp}. To abstract from the noise characteristic of the vehicles' perception system, an error rate for the honest agents is assumed. This error rate represents the fact that even honest agents occasionally report wrong information due to measurement noise and imperfection of their perception systems. The simulation is run several times for different assumed error rates of the honest agents and different choices of the threshold parameter θ_{DC}. The results are depicted in Fig. 5.15.

It can be seen in Fig. 5.15 that the false alarm rate strongly depends on the error rate



(a) Detection probability as a function of the error rate and the threshold parameter θ_{DC}.

(b) False positive rate as a function of the error rate and the threshold parameter θ_{DC}.

Figure 5.15: Detection probability and false positive rate in dependence of the threshold θ_{DC} and the error rate of the agents resulting from the large-scale traffic simulation in Cologne ([MMSB19], © 2019 IEEE).

of the agents, while the detection probability of an actual attacker stays almost constant. Furthermore, it shows that the classification results are almost the same for a wide range of $\theta_{DC} \in [0.05, 0.20]$, while choosing the parameter oddly deteriorates the results. This holds as long as the error rate is in a reasonable range, i.e. below 20 %, while at 20 %, an adequate choice of θ_{DC} reflects the trade-off between false positives and detection probability. Therefrom, it can be concluded that the proposed algorithm is quite robust with respect to the choice of the design parameter θ_{DC} as long as it is chosen within a reasonable interval and the error rate of the honest agents stays moderate.

To further interpret the results, a simplified illustrative example is chosen. It can be seen from Fig. 5.15 that the detection probability for an error rate of 10 % and $\theta_{DC} = 0.1$ is about $p_{\text{tp}} \approx 45$ %, while the false positive rate is $p_{\text{fp}} \approx 10$ %. For the simplified example, it is assumed that the trust in an agent is depleted if four misbehaviors are detected within a batch of four consecutive reports, while the trust is restored for each batch. Then, the probability of correctly detecting an actual misbehaving agent p_{dm} and the probability of wrongly blaming an actually honest agent p_{wb} are given by

$$p_{\text{dm}} = 1 - (1 - p_{\text{tp}}^4)^n \quad \text{and} \quad p_{wb} = 1 - (1 - p_{\text{fp}}^4)^n \,, \tag{5.23}$$

respectively, where n is the number of batches. Hence, after $n = 30$ batches or a total of 120 reports, $p_{wb} = 71.5$ % of the misbehaving agents are banned from the system, while

only $p_{wb} = 0.30\,\%$ of honest agents are excluded. Thus, even if several measurements are bundled to a single reported opinion and even if there are time gaps between the reports, at a typical measurement update rate of 10 Hz, it is a matter of minutes until the vast majority of misbehaving agents are excluded from the system. This demonstrates that the misbehavior detection mechanism scales well even to big scale scenarios.

5.4 Summary

While external information can be used to increase the performance of a motion planning scheme, relying on a corrupted external environment model or a breaking communication link may lead to significant harm. Hence, knowing the reliability of the communication link and the external environment model is vital to safe connected automated driving. To estimate the reliability of the V2X link to the CAV, the communication channel was modeled as a Markov chain, and a novel SL-based identification algorithm was introduced to identify its parameters. Evaluations of simulations and real-world experiments showed that, besides featuring an additional measure for the statistical uncertainty, the proposed algorithm yields results with far less variance than the classical method. In the next step, an SL-based reliability estimation scheme was presented which used two consistency tests and two plausibility tests to estimate the reliability of an external environment model. The proposed estimation scheme reliably operates online in real time even if only very few measurements are available. In the fusion, the four individual test results are weighted according to the statistical evidence supporting them. Hence, the overall result is adapted to the individual situation and reflects the available knowledge. The reliability estimation scheme was evaluated on experiments showing that the reliability of the reliability estimation scheme for a confidence level of 95 % is above 90 %. Finally, the approach was applied to a V-MAS showing that the estimation scheme can be used to establish trust among the agents. It was shown with simulation studies that the respective trust management system can effectively identify and remove misbehaving agents and even collaborating attackers from the V-MAS. Large scale simulations on the mixed traffic in the city of Cologne, conducted by a different research group, have verified that the approach scales well even to large-scale systems.

Chapter 6

Motion Planning for Connected Automated Vehicles at Occluded Intersections

In this chapter, based on the author's pre-published article [MB19; MHSB21], a motion planning scheme is proposed that can use external information to resolve occlusion. Furthermore, the proposed planning scheme solves behavior and trajectory planning holistically and it accounts for the uncertainties and the reliability of its input data. Thus, it can handle the application scenario from Chapter 2.1.

An overview of the motion planning scheme is given in Fig. 6.1. The motion planning scheme consists of two main modules, the context awareness module and the actual motion planning module, which are then subdivided into further submodules. The context awareness module preprocesses the information from the reliability estimation (see Chapter 5), the external and the ego environment model, and it derives a formalized representation of the situation context $\mathcal{C}$. The situation context $\mathcal{C}$, the object lists from the external environment model $\mathcal{L}_{\text{ext}}$, and the object list $\mathcal{L}_{\text{ego}}$ from the ego environment model then are used in the motion planning module to determine the trajectory of the CAV. The additional information from $\mathcal{L}_{\text{ext}}$ herein serves to resolve occlusions in the ego perception. Instead of merging the external and ego object lists through T2T fusion, the object lists are used in parallel. Thus, the origin of any piece of information is maintained and self-fulfilling prophecies by circling the same information between communication partners can be avoided.

The CAV has three main goals while driving through the structured environment of urban areas: safety, passenger comfort, and robustness. These goals are addressed by an optimization-based approach, which is subdivided into two parts: First, the sampling-based longitudinal planning examines the available behavior options from the situation context and, with that, calculates a longitudinal trajectory which fulfills certain optimality criteria. Thus, decision making and trajectory planning is solved holistically. Second, the MPC-based

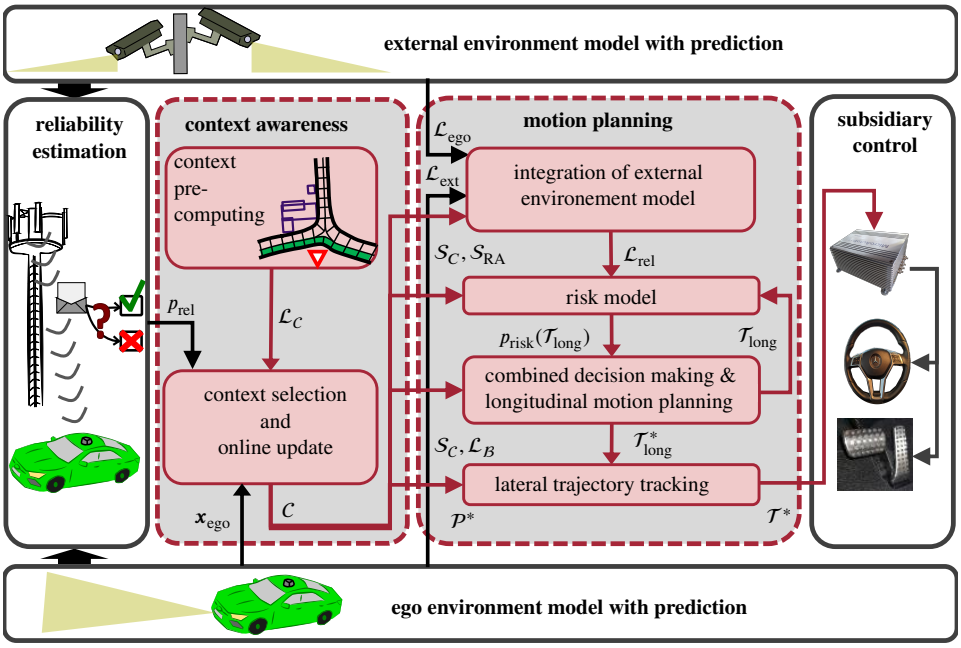


Figure 6.1: Overview on the motion planning scheme (cf. [MHSB21]).

lateral trajectory tracking takes care of lane keeping while following a longitudinal reference trajectory.

The longitudinal planning is formulated as an OCP that accounts for the maximum acceptable residual risk p_{risk} of a planned maneuver as a constraint, while optimizing for safety and passenger comfort. This p_{risk} is provided by a risk model that calculates the probability of leaving a set of safe states using the predicted trajectories of the objects in the environment models, their uncertainties, and the estimated reliability of the external environment model. By incorporating set-based methods into the motion planning scheme, it inherits their safety guarantees for the special case of a vanishing residual risk, while the guarantees are softened to guarantees in a probabilistic sense otherwise. In contrast to most POMDP-based methods, such as [HSX+18], the interactions between all known objects and the CAV are considered as constraint, while the risk of endangering other road users is minimized. Hence, the prediction of other road users is only loosely coupled with the motion of the CAV, and it can thus be expected that there is only an insignificant reduction in performance when performing the prediction and the motion planning separately. On the other hand, the computational burden can be reduced significantly, and the robustness can be increased, since errors and uncertainties in the object lists cannot be amplified in a respective feedback loop.

6.1 Context Awareness

A key challenge for motion planning is balancing optimality for a specific situation and generality. A motion planning scheme needs to be sufficiently specific so that it can efficiently solve the corresponding task. At the same time, it needs to be general to be applicable to a wide range of scenarios. In the proposed motion planning scheme, this challenge is addressed by the concept of context awareness, which is first presented in general (see Section 6.1.1) and then illustrated by the example of the application scenario (see Section 6.1.3).

6.1.1 General Concept

The key idea behind the context awareness described in this thesis is to preprocess the inputs of the motion planning scheme and aggregate them into a formalized representation of the current traffic situation, the situation context

$$\mathcal{C}_{[s_j,s_{j+1}]} = (\mathcal{S}_C, \mathcal{S}_{\mathrm{RA}}, \mathcal{L}_{\mathcal{B}})\,. \tag{6.1}$$

The situation context (6.1) comprises four basic parts. First, an interval $[s_j, s_{j+1}]$ indicates which part of the route from the waypoint s_j to the waypoint s_{j+1} is described by the respective context. Second, a set of constraints $\mathcal{S}_C$ that the CAV has to obey, is included. Third, the context contains a set of regularizing assumptions $\mathcal{S}_{\mathrm{RA}}$ used to ensure that the planning problem is well-posed an can be solved efficiently. Fourth, $\mathcal{C}_{[s_j,s_{j+1}]}$ comprises a list of behavior options $\mathcal{L}_{\mathcal{B}}$, which spans the search space of longitudinal planning.

While intuitively, the constraints and regularizing assumptions should stay the same within a situation, $\mathcal{S}_C$ and $\mathcal{S}_{\mathrm{RA}}$ generally depend on the longitudinal position s. More precisely, $\mathcal{S}_C$ and $\mathcal{S}_{\mathrm{RA}}$ are segment-wise constant but change in between the segments. Hence, the route is split and the situation context is defined on these route segments specified by $[s_j, s_{j+1}]$. Thus, $\mathcal{S}_C$ and $\mathcal{S}_{\mathrm{RA}}$ become position invariant within the respective situation contexts.

The set of constraints $\mathcal{S}_C$ consists of conditions like the traffic rules that the CAV must fulfill. These conditions are formulated as mathematical statements that either require a variable to be an element of a given set or to exclude a variable from a given set. One option to specify such a set is to describe it as a function, which is required to be zero. This leads to equality constraints. Alternatively, such a set can be defined by describing its bounds. This yields inequality constraints. For example, $\mathcal{S}_C$ includes yield lines, mandatory stops, the legal speed limit $v_{\mathrm{sl}}(s)$, and the safety distances $s^+(t)$ and $s^-(t)$ to the road user ahead and behind the CAV. Moreover, priority lanes can be described by polygons. The objects within these polygons correspond to road users on the priority lanes to whom the CAV must yield the right of way. Hence, these objects are sorted into the list of relevant objects $\mathcal{L}_{\mathrm{rel}}$ so that

they are considered in th risk model (see Section 6.2.2). Furthermore, $\mathcal{S}_C$ contains physical limitations like the longitudinal minimum and maximum accelerations $a_{\min}$ and $a_{\max}$, and the speed limit profile

$$\tilde{v}_{\max}(s) = \min\left\{ v_{\mathrm{sl}}(s),\ \sqrt{\frac{||a_{\perp,\max}||}{\kappa_{\mathrm{ref}}(s)}} \right\}. \tag{6.2}$$

In (6.2), the curvature of the road $\kappa_{\mathrm{ref}}(s)$ is calculated with the method from Section 3.2.

The set of regularizing assumptions $\mathcal{S}_{\mathrm{RA}}$ comprises the same type of mathematical statements as $\mathcal{S}_C$. These statements are used to ensure that the planning problem is well-posed. Moreover, $\mathcal{S}_{\mathrm{RA}}$ supports the planning algorithm to determine a good solution quickly. For example, $\mathcal{S}_{\mathrm{RA}}$ comprises the rational driver assumption, which states that other road user also follow the traffic rules. However, if a road user is observed in clear violation of the traffic rules, this assumption must be dropped. Another example is the assumption of reliable perception used to construct the ISS in which the CAV is supposed to move. However, this assumption is rarely completely accurate. Hence, the reliability estimation from Section 5.2 is used to determine the likelihood p_{rel} that the assumption is satisfied sufficiently well. To reduce computational effort, $\mathcal{S}_{\mathrm{RA}}$ also includes heuristics such as the assumption that passengers like to travel at constant speed through curves. Under this assumption, $\tilde{v}_{\max}(s)$ can be approximated by a section-wise constant function $v_{\max}(s)$, where $\tilde{v}_{\max}(s)$ is either $v_{\mathrm{sl}}(s)$ or the maximum velocity at the local curvature maximum $\kappa_{\mathrm{ref,max}}$. Furthermore, $\mathcal{S}_{\mathrm{RA}}$ contains rules with which the available behavior options are prioritized. This allows the planning algorithm to find favored solutions quickly and return as soon as such a solution is found.

Finally, the situation context comprises the list of behavior options $\mathcal{L}_{\mathcal{B}}$, where each behavior option correspond to a trajectory candidate. Such a behavior option is formalized as

$$\mathcal{B} : (\mathcal{P},\ \langle(\boldsymbol{x}_f, t_f)\rangle_i,\ I). \tag{6.3}$$

This is, $\mathcal{B}$ consists of three parts: First, $\mathcal{B}$ includes a reference path $\mathcal{P}$, which guides the CAV along its route. Second, $\mathcal{B}$ comprises a target sequence $\langle(\boldsymbol{x}_f, t_f)\rangle_i$, $i = 1, \dots, N$ of N target states $\boldsymbol{x}_{f,i}$ with their respective target times $t_{f,i}$ that specify the maneuver timing. Third, the importance I in $\mathcal{B}$ is used to prioritize the behavior options and thus to sort $\mathcal{L}_{\mathcal{B}}$. The reference path $\mathcal{P}$ is represented as the curvature along the 1D positon $\kappa_{\mathrm{ref}}(s)$, since this representation is required by the nonlinear vehicle model (3.16) from Section 3.2. In most situations, $\mathcal{P}$ is just the curvature of the road at the specified position and, thus, the same $\mathcal{P}$ is used for all behavior options. However, for situations like obstacle avoidance or overtaking, e.g., the method from [TK05] can be used to generate behavior options differing paths $\mathcal{P}$. Since the motion planning scheme calculates the trajectory segment-wise from one target state to the next one along the target sequence $\langle(\boldsymbol{x}_f, t_f)\rangle_i$, the placement of the individual

target states adds additional degrees of freedom to the specification of a certain behavior. This, however, comes at the price of an increased sample space and thus higher computational burden. Moreover, each additional target state further restricts the optimization and thus reduces the solution quality. Therefore, the target sequences are kept as short as possible and rarely exceed $N \leq 3$ in practice. Finally, the importance I, besides prioritizing the behavior options, serves as break condition. As soon as a valid trajectory is found and there are no further unexplored behavior options of the same importance I, the search for trajectories is terminated. Hence, the average calculation time is reduced. Furthermore, the importance I increases the robustness of the planning scheme, since optimization weights only have to be balanced within their importance class. For example, an uncomfortable fail-safe trajectory has always the lowest I and, thus, is only considered, if there is no other choice.

6.1.2 Processing of the Situation Context

The situation context is determined in two stages: In the first stage, static information from the map $\mathcal{M}_{\text{local}}$, like the road geometry and the traffic rules along the route $\mathcal{R}$, are precomputed. This yields $\mathcal{S}_C$, $\mathcal{S}_{\text{RA}}$, and the segments $[s_j, s_{j+1}]$ for which $\mathcal{S}_C$ and $\mathcal{S}_{\text{RA}}$ are position invariant. Moreover, some behavior options, particularly those of the fail-safe strategy, are determined. This yields the precomputed situation contexts, which are put into $\mathcal{L}_C$.

In the second stage, the currently active situation context is identified and updated with dynamic information such as the perception data. To select the active situation context, the CAV's curent state is projected to the road, which yields the longitudinal state $\boldsymbol{x}(0)$. Then, the $C_{[s_j,s_{j+1}]} \in \mathcal{L}_C$ which fulfills $s_j \leq s < s_{j+1}$ is selected. However, dynamically adapting s_j, s_{j+1} is not considered, since the expected gain does not justify the additional computational effort. In the next step, the selected situation context is updated with dynamic information from the perception. First, $\mathcal{S}_C$ and $\mathcal{S}_{\text{RA}}$ are updated. For example, the speed limit profile $v_{\max}(s)$ is reduced to the dynamic speed limit profile

$$v_{\max}(s,t) = \begin{cases} v_{\max}(s)\,, & s < s_{\text{pred}}(t) - s^{+}(t)\,, \\ \min\left\{v_{\text{pred}}(t), v_{\max}(s)\right\}\,, & \text{otherwise}\,, \end{cases} \tag{6.4}$$

if a road user is detected ahead of the CAV. In (6.4), $s_{\text{pred}}(t)$ and $v_{\text{pred}}(t)$ denote the predicted position and velocity of the road user ahead. Hence, all valid trajectory candidates remain outside the reachable set (see Section 3.3) of the road user ahead, which prevents a collision.

The next step is to check if the available behavior options still comply with the updated $\mathcal{S}_C$ and $\mathcal{S}_{\text{RA}}$. At this point, set-based methods get involved to provide the motion planning with safety guarantees. For this, the reachable set is calculated for all other road users and the under-approximations $\hat{\mathcal{S}}_{\text{ISS}}(t)$ of the CAV's ISS (see Section 3.3) are computed, e.g., using

the method from [PA18b]. The key property of the ISS is that the CAV can stay in the ISS for arbitrary time and that the safety is formally guaranteed as long as the CAV stays within it. Thus, by requiring that every valid trajectory ends in a set $\hat{S}_{\text{ISS}}(t)$, the time over which the residual risk must be accumulated is restricted to the time interval $[0, t_{f,N}]$.

Invalidated behavior options are dropped. For the remaining behavior options, it is checked whether they stay within one $\hat{S}_{\text{ISS}}(t)$. If so, the safety of the corresponding trajectory is formally guaranteed and the residual risk becomes zero. Otherwise, the PGA and the PNR are calculated for the behavior option and the target sequence is modified such that the final target state becomes the PGA and that another target state becomes the PNR. This is important for the risk model (see Section 6.2.2). Furthermore, the target sequences are updated by dropping target states, which the CAV has already passed and by adjusting the target times according to the time the CAV already is in the current situation context. Moreover, new behavior options are generated. For example, if an obstacle occurs in front of the CAV, alternative paths are generated to avoid the obstacle. For these new behavior options, the PGAs and the PNRs are also computed and the target sequence is determined accordingly.

6.1.3 Situation Context for the Application Scenario

To illustrate the general concept of context awareness, the situation context for the application scenario from Section 2.1 is determined as an example. First, the reference path $\mathcal{P}$, which in this case is the same for all behavior options, is determined from the local map $\mathcal{M}_{\text{local}}$ and the route $\mathcal{R}$. For this, the $\kappa_{\text{ref}}(s)$ representation is computed using the curvature MPC (see Section 3.2). Furthermore, the traffic rules, $a_{\min}$, and $a_{\max}$ are loaded and $v_{\max}(s)$ is computed as the approximation of (6.2). This is, the speed limit profile get $v_{\max}(s) \approx 8.33\ \text{ms}^{-1}$ around the junction due to the speed limit, and $v_{\max}(s) \approx 4\ \text{ms}^{-1}$ while cornering.

In the next step, the PNRs and PGAs are determined. For the application scenario, the PNR is the state on the trajectory, where the CAV can ultimately stop at the yield line. Accordingly, the PGA is a state on the trajectory where the CAV has successfully merged. This is specified by a regularizing assumption, which is determined from the road geometry. Thus, while the 1D position of the PNR s_{PNR} is a function of the velocity in the PNR v_{PNR}, all PGAs share a common position s_{PGA} for varying velocities v_{PGA}. Since the PGA represents the end of the merging scenario, it also concludes the situation context, i.e. $s_{j+1} = s_{\text{PGA}}$. In turn, the beginning of the situation context s_j is determined as the first position from which s_{PGA} is in the reachable set of the CAV with respect to the prediction horizon T_{pred}.

The following heuristics are used to prioritize the behavior options: Dynamic merging gets the highest priority $I = 2$. For gently stopping at the yield line, $I = 1$ is assigned. Finally, the lowest importance $I = 0$ is given to the fail-safe strategy of stopping at the yield line with constant deceleration.

The target sequence $\langle(\boldsymbol{x}_f, t_f)\rangle_i$ of behavior options $\mathcal{B}_k$ that correspond to dynamically merging comprises $N \leq 3$ target states: a PNR, the point at which the CAV has turned and can accelerate again, and a PGA. Multiple behavior options $\mathcal{B}_k$ are generated for this behavior by sampling over the PNR velocities v_{PNR} and the corresponding target times $t_{f,1}$ on the one hand, and over the PGA velocities v_{PGA} and their target times $t_{f,3}$ on the other hand. From these quantities and the maximum curve velocity, $\langle(\boldsymbol{x}_f, t_f)\rangle_2$ is calculated. In contrast, the target sequences of behavior options from the other two behavior classes consist of only one target state $\langle(\boldsymbol{x}_f, t_f)\rangle_1 = \langle([s_{\text{yield}}, 0, 0]^T, t_{f,1})\rangle$. To generate multiple behavior options $\mathcal{B}_k$ for these behavior classes, the final time $t_{f,1}$ is sampled.

If a road user is detected ahead of the CAV, the behavior options are adapted dynamically. In this case, the CAV can either follow the road user ahead and stop at the yield line, or follow the road user to the PGA. Both behaviors comprise behavior options that contain target sequences with $N = 2$. Furthermore, for both behavior classes, based on the predicted trajectory of the road user ahead, $v_{\max}(s)$ is updated according to (6.4). Moreover, in both cases, the first state of the target sequence

$$\langle([s_{\text{pred}}(t_{f1}) - s^{+}(t_{f1}), v_{\max}(s(t_{f1}), t_{f1}), 0], t_{f1}),\ ([s_{f2}, v_{f2}, 0], t_{f2})\rangle \tag{6.5}$$

of each behavior option is determined by sampling over $t_{f,1}$ and calculating the respective positions and velocities. Furthermore, the final times $t_{f,2}$ are sampled for both behavior classes. However, the calculation of the second target state differs: for the prioritized behavior option of driving to the PGA $s_{f2} = s_{\text{PGA}}$,$v_{f2} = v_{\max}(s_{\text{PGA}}, t_{f2})$, and $I = 2$ is set, whereas yielding is parametrized through $s_{f2} = s_{\text{yield}}$, $v_{f2} = 0$, and $I = 1$.

All this information is aggregated in the situation context $C_{[s_j, s_{j+1}]}$, where $[s_j, s_{j+1}]$ is determined from the PGA position as s_{j+1} and the first position that includes the PGA position in its reachable set as s_j. The dynamic constraints $a_{\min}$, $a_{\max}$, and $v_{\max}(s,t)$, the position of the yield line, and a polygon describing the priority lane are included in the set of constraints $\mathcal{S}_{\text{C}}$. Furthermore, the set of regularizing assumptions $\mathcal{S}_{\text{RA}}$ comprises the probability of reliable perception p_{rel} determined by the reliability estimation scheme (see Section 6.2.2), v_{sl} and the respective minimum and maximum acceleration for the other road users, and the position of the PGA s_{PGA}. Finally, all behavior options are listed in $\mathcal{L}_{\mathcal{B}}$. They all share the same $\mathcal{P}$.

6.2 Motion Planning

The motion planning module consists of four submodules that determine the optimal trajectory $\mathcal{T}^*$ from the situation context and the perception information: the integration of the external information, the risk model, the longitudinal planning, and the lateral trajectory tracking. Both the longitudinal planning and the lateral trajectory tracking are formulated as OCPs.

However, the longitudinal planning is addressed by a sampling-based approach, whereas an MPC method is used for the lateral trajectory tracking. Since the longitudinal planning solves behavior planning and (longitudinal) trajectory planning holistically, particularly for complex maneuvers in urban areas, long planning horizons are required. Moreover, the solutions might be distributed over several homotopy classes, i.e. disjoint convex areas in the state space that correspond to distinct behaviors, like dynamically merging into a traffic gap versus gently stopping at a yield line. This makes a sampling-based approach preferable for the longitudinal planning. In contrast, the prediction horizon of the lateral trajectory tracking can be chosen shorter, since it is guided by the longitudinal trajectory. Furthermore, the lateral trajectory model requires the nonlinear vehicle model (3.16), which is difficult to grasp analytically, while it can be solved easily with numerical methods. Therefore, the MPC approach appears suitable for the lateral trajectory tracking.

6.2.1 Integration of an External Environment Model

While the ego perception is designed to feature sufficient accuracy and range so that motion planning schemes can safely and efficiently navigate automated vehicles, particularly in urban areas, occlusions due to static obstacles like buildings or due to other road users might significantly compromise efficiency and, at times, even safety. Combining external information with the ego perception can resolve such occlusions and thus mitigates their negative impact. In CAVs, external information is typically combined with the ego perception through T2T fusion. Thus, in the mutually observed areas, the precision can be increased and clutter objects can be pruned, since they usually occur only in one information source and their existence probability drops if they are not confirmed by other information sources. Furthermore, T2T fusion may benefit the birth process of objects in the ego perception, if adjoining areas of the ego perception are covered by external perception. Such an environment model resulting from T2T fusion could be used as input for the motion planning.

However, in the context of connected automated driving, T2T fusion comes with some dangers. First, since the ego perception is designed to feature sufficient accuracy and since bandwidth in low latency communication is limited and expensive, it can be expected that the shared external perception typically covers an FOV that is mostly disjoint with the ego perception. This is illustrated in Fig. 6.2 using the application scenario as an example. In such a scenario, the T2T fusion practically results in the concatenation of the object lists $\mathcal{L}_{\text{ego}}$ and $\mathcal{L}_{\text{ext}}$ and therefore cannot play out its advantages. Furthermore, since classical T2T fusion does not keep track of the origin of the information making its way into the fusion result, self-fulfilling prophecies may occur if the fusion result is transmitted to further agents or fed back to a central authority, like a MEC server. Moreover, in a connected automated driving setup, it can be expected that the data from different sources contain some degree of inconsistency, since the sensors are typically widely distributed and the tracking has to deal with extended objects. However, if the data is not fully consistent, the data association might

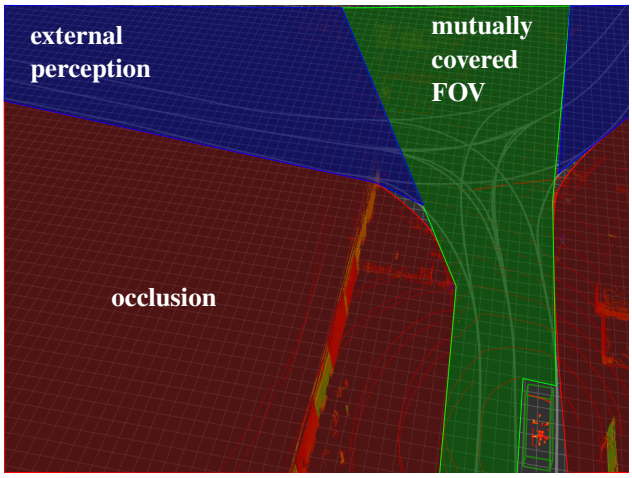


Figure 6.2: Mutual FOV and FOV of the external perception. Due to the occlusion, only the external perception covers the regions relevant for planning the merging maneuver.

fail in some situations, which could lead to either spawning duplicate objects or wrongly pruning objects that are not sufficiently confirmed by multiple sensors.

Therefore, an alternative approach is proposed that circumvents the use of T2T fusion and the dangers that come with it by using the external and the ego environment model in parallel. First, a simple nearest neighbor association with a maximum association range is used to identify clearly redundant objects. Then, based on the situation context, it is decided which of the objects are relevant for planning and are therefore sorted into the list of relevant objects $\mathcal{L}_{\text{rel}}$, and which are not. As disregarding objects in the environment of the CAV is safety-critical, when in doubt, a clutter object is considered rather than a relevant object is overlooked. Since the environment models are kept completely separate and $\mathcal{L}_{\text{rel}}$ is only used in the proposed planner and not communicated to other agents, with the parallel use approach, the self-fulfilling prophecy issue can be avoided. Furthermore, selecting objects according to their relevance rather than according to their association probability makes the approach more conservative, but as well more robust with respect to decision making. The list of relevant objects $\mathcal{L}_{\text{rel}}$ then is fed into the risk model, which is used in combination with the situation context $C_{[s_j, s_{j+1}]}$ to choose the behavior option that yields the optimal trajectory.

6.2.2 General Risk Model

One of the key goals of the motion planning scheme is guaranteeing the safety of the CAV while navigating it through urban traffic. Set-based methods allow to determine ISSs, where trajectories that are fully included within such an ISS are formally guaranteed to be safe, if the set of constraints $\mathcal{S}_C$ and the set of regularizing assumptions $\mathcal{S}_{\mathrm{RA}}$ used to determine the respective ISS are fulfilled. Thus, the proposed motion planning scheme settles upon set-based methods to inherit the formal safety guarantee.

However, purely set-based methods, despite their appealing formal safety guarantees, come with severe issues that limit their practical usability. First, finding an ISS big enough to contain comfortable trajectories in complex urban traffic scenarios requires to concisely estimate the reachable sets of the other road users. However, concisely estimating the reachable sets rather than roughly over-estimating them requires complex set-based models and computationally expensive sampling over possibly large state spaces [AGH+20]. Although progress has been made and the application of purely set-based methods has been shown for some practical examples, e.g. [KA20], the principal problem remains. In any case, the resulting trajectories are maximally conservative with respect to decision making. Second, set-based methods have not been commonly used so far. Hence, most of the existing methods describe the objects of an environment model probabilistically. Probabilistic descriptions, however, are incompatible with deterministic set descriptions. Therefore, specific regularizing assumptions on modeling errors of the probabilistic description, e.g. a 3σ cutoff of distribution tails, is required to interface with probabilistic environment models. These regularizing assumptions, however, undermine the formal safety guarantees. Third, some of the regularizing assumptions typically used by set-based methods are at least questionable. While the rational driver assumption holds in general, there are always some drivers who do not obey the traffic rules. Furthermore, the assumption of reliable perception requires the ego perception and the external perception to detect all relevant objects under all circumstances at any time. However, although the detection capabilities of current perception systems are pretty high, they still can silently fail in rare corner cases.

Therefore, a probabilistic risk model is proposed that extends the set-based methods by a probabilistic interface and thus bridges the gap between the set-based and the probabilistic methods. ISSs are estimated based on rough over-approximations of the reachable sets of the other road users. Thus, the estimated ISSs are vastly under-approximated, yet with low computational burden. If the traffic situation is simple enough that an estimated ISS and a comfortable trajectory within this estimated ISS can be found and, furthermore, the assumptions are certainly fulfilled, the respective trajectory is chosen and the full formal guarantees are inherited from the set-based methods. Moreover, based on set reasoning, a fail-safe strategy is developed for the respective situation that allows to stay within an ISS as long as possible. However, if there is some risk that one of the regularizing assumptions could be violated, or no trajectory other than the fail-safe trajectory is found, which is completely

contained in the ISS around the CAV, the probabilistic extension comes into play. In this case, the risk model uses the estimated risk that a regularizing assumption could be violated, i.e. p_{RA}, to calculate the residual risk p_{risk}. Furthermore, it uses the PNR and PGA of the examined behavior options to calculate the residual risk of the respective trajectory candidate

$$p_{\mathrm{risk}}(\mathcal{T}_{\mathrm{long}}) = (1 - p_{\mathrm{RA}}) + p_{\mathrm{RA}} \int_{t_{\mathrm{PNR}}}^{t_{\mathrm{PGA}}} \int_{s_{\mathrm{ego}}(t)-s^{-}(t)}^{s_{\mathrm{ego}}(t)+s^{+}(t)} f_{\mathcal{L}_{\mathrm{rel}} \cup \mathcal{L}_{\mathrm{virt}}}(s,t)\, \mathrm{d}s\, \mathrm{d}t\,. \tag{6.6}$$

The residual risk $p_{\mathrm{risk}}(\mathcal{T}_{\mathrm{long}})$ then describes the probability that the trajectory candidate $\mathcal{T}_{\mathrm{long}}$ is not included in the estimated set of safe states $\hat{\mathcal{S}}^{t}_{\mathrm{safe}}$. In (6.6), $s^{-}(t)$ and $s^{+}(t)$ describe the safety distances to other road users behind and in front of the CAV, s is the 1D position of the CAV, p_{RA} represents the probability that the regularizing assumptions hold, and $f_{\mathcal{L}_{\mathrm{rel}} \cup \mathcal{L}_{\mathrm{virt}}}(s,t)$ refers to the joint distribution of the objects from the list of relevant objects $\mathcal{L}_{\mathrm{rel}}$ and from the list of virtual objects $\mathcal{L}_{\mathrm{virt}}$. The list of virtual objects $\mathcal{L}_{\mathrm{virt}}$ is used to model epistemic uncertainty, e.g., due to remaining occlusions or the end of sight (EOS). The influence of the respective objects from $\mathcal{L}_{\mathrm{rel}} \cup \mathcal{L}_{\mathrm{virt}}$ to $f_{\mathcal{L}_{\mathrm{rel}} \cup \mathcal{L}_{\mathrm{virt}}}(s,t)$ can be weighted with the probability p_{im} that the respective object actually impacts the motion planning. For example, with respect to virtual objects, p_{im} can be interpreted as the probability that the virtual object actually exists. With respect to an object from $\mathcal{L}_{\mathrm{rel}}$, p_{im} can be interpreted as the probability that the object stays on a lane relevant to the intended maneuver. Thus, figuratively speaking, the CAV can speculate on a specific evolution of the current traffic situation and adapt its behavior respectively. Since the whole set $\hat{\mathcal{S}}_{\mathrm{safe}}(t)$ gets invalid if one of the assumptions it bases on breaks down, $(1 - p_{\mathrm{RA}})$ directly adds to p_{risk}, while the integral term in (6.6) has to be weighted with p_{RA} for normalization.

6.2.3 Risk Model for the Application Scenario

For the application scenario from Section 2.1, it is assumed as certain that the rational driver assumption applies with regard to lane keeping, as long as other road users are not observed to change their lane illegally. Therefore, all objects are projected onto the reference path that corresponds to their lane. Then, the motion of other road users is predicted with a probabilistic forecasting model that accounts for the possibility that drivers may violate the speed limit. Hence, the rational driver assumption is not used to strictly constrain the other road users to the speed limit. Moreover, the probability that the assumption of reliable perception holds is calculated with the reliability estimation scheme, as described in Section 5.2.5. More specifically, $p_{\mathrm{RA}} = p_{\mathrm{rel}}$ is obtained from (5.12). As stated in the situation context, the relevant objects are the road users on the ego lane and, for $t \in [t_{\mathrm{PNR}}, t_{\mathrm{PGA}}]$, the road users on the lane the CAV intends to merge. Furthermore, a virtual object is placed at the EOS which approaches the junction with the velocity of the speed limit $v = v_{\mathrm{sl}}$. This reflects

the epistemic uncertainty of the external perception at the EOS and makes the CAV prefer to merge quickly into the junction as long as the absence of other road users is guaranteed. The individual objects are modeled by Gaussian distributions. Therefore, the probabilities

$$\tilde{\tilde{p}}_{\text{risk},j}(\mathcal{T}_{\text{long}}, t) = \operatorname{erf}\left(\frac{s(t) - s^{-} - \mu_{o_j}(t)}{\sigma_{o_j}(t)} \right) - \operatorname{erf}\left(\frac{s(t) + s^{+} - \mu_{o_j}(t)}{\sigma_{o_j}(t)} \right) \tag{6.7}$$

for each object $o_j \in \mathcal{L}_{\text{rel}} \cup \mathcal{L}_{\text{virt}}$ can be used to express the space integral from (6.6). Here, $\tilde{\tilde{p}}_{\text{risk},j}(t)$ from (6.7) describes the probability that the object o_j violates the safety distances at time t such that at least one state on the trajectory of the CAV gets into the reachable set of o_j and thus is not included in $\hat{\mathcal{S}}^t_{\text{safe}}$. Moreover, erf$(\cdot)$ refers to the error function of the standard normal distribution, whereas μ_{o_j} and σ_{o_j} are the mean and standard deviation of the predicted object position. Aggregating the risk $\tilde{\tilde{p}}_{\text{risk},j}(\mathcal{T}_{\text{long}}, t)$ for all instances of time and all objects then yields the result of the temporal and spacial integral term from (6.6). Note, however, that simply summing up $\tilde{\tilde{p}}_{\text{risk},j}(\mathcal{T}_{\text{long}}, t)$ does not yield correct results, since the interdependence between the objects, so far all modeled as independent Gaussians, is not reflected. To account for the interdependence, (6.6) is modeled by the iteration

$$\tilde{p}_{\text{risk},j}(\mathcal{T}_{\text{long}}) = \max_t \{ \tilde{\tilde{p}}_{\text{risk},i}(\mathcal{T}_{\text{long}}, t) \},\ t \in [t_{\text{PNR}}, t_{\text{PGA}}] , \tag{6.8a}$$

$$p_{\text{risk},0}(\mathcal{T}_{\text{long}}) = \tilde{p}_{\text{risk},0} , \tag{6.8b}$$

$$p_{\text{risk},j+1}(\mathcal{T}_{\text{long}}) = p_{\text{risk},j} + (1 - p_{\text{risk},j}) \cdot \tilde{p}_{\text{risk},j+1} , \tag{6.8c}$$

$$p_{\text{risk}}(\mathcal{T}_{\text{long}}) = (1 - p_{\text{rel}}) + p_{\text{rel}} \cdot p_{\text{risk},N}(\mathcal{T}_{\text{long}}) . \tag{6.8d}$$

Put simply, the iteration (6.8) provides the probability that, either because of unreliable perception or because one of the relevant objects comes dangerously close, the trajectory of the CAV is not included in a validly estimated safe set.

6.2.4 Combinde Decision Making and Longitudinal Trajectory Planning

In essence, the combined decision making and longitudinal trajectory planning determines the optimal trajectories along the reference path $\mathcal{P}$ for each behavior option, calculates their respective total costs, and selects the behavior option whose corresponding trajectory satisfies the constraints and has minimum total cost. Mathematically, the cost function J_{long} for the longitudinal planning is formulated as

$$J_{\text{long}}(\mathcal{T}_{\text{long}}) = \sum_{i=1}^{N} \left(\int_0^{\Delta t_{f,i}} \frac{1}{2} \left(\frac{w_t - 1}{1 + t} + 1 \right) u^2(t) \mathrm{d}t + w_{t_f}\, \Delta t_{f,i}^2 \right) + w_{\text{risk}}\, p_{\text{risk}}(\mathcal{T}_{\text{long}}) , \tag{6.9}$$

where $\Delta t_{f,i} = t_{f,i} - t_{f,i-1}$ describes the time the CAV needs to travel from the target state $\boldsymbol{x}_{f,i-1}$ to the target state $\boldsymbol{x}_{f,i}$, w_t is a time weight for the longitudinal jerk $u(t)$, w_{t_f} is a weight that balances priority of time optimality versus time-weighted jerk optimality, and w_{risk} is the weight of the residual risk. Since low jerk is a common criterion for passenger comfort [WZKT10], a function of the longitudinal jerk $u(t)$ is chosen for the integral cost. However, due to the steadily changing environment, the planning might frequently switch between strongly different behavior options. To mitigate jerky behavior due to this replanning effect, the longitudinal jerk $u(t)$ is time-weighted such that jerk that has to be applied soon is weighted more than jerk that has to be applied far into the future. This time-weigthing furthermore is an additional parameter to tune the behavior of the CAV and render it more human-like. As short travel times are convenient, the final time is added to the cost function as terminal cost. Moreover, to support safety, the residual risk $p_{\text{risk}}(\mathcal{T}_{\text{long}})$ as determined by the risk model for the trajectory $\mathcal{T}_{\text{long}}$ is added to the terminal cost. The cost function is defined segment-wise along the target sequence, which furthermore acts as equality constraint. This allows to calculate the trajectory candidates segment-wise as will be shown in Theorem 6.1.

Since the motion planning scheme is intended to guide the CAV through the structured environment of urban areas, path-velocity decomposition can be applied. This allows the combined planning to focus on decision making and longitudinal planning based on the linear point-mass model (3.14), while the MPC-based lateral trajectory tracking takes care of laterally stabilizing the CAV on its lane. Finally, to increase safety, the residual risk of the planned trajectory is constrained by a maximum acceptable residual risk $p_{\text{risk,max}}$. Thus, in total, the longitudinal motion planning can be described by the OCP

$$u^*(t) = \arg\min_{u(t)} \left\{ J_{\text{long}} \right\} \quad \text{subject to} \tag{6.10a}$$

$$\dot{\boldsymbol{x}}(t) = \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} s \\ v \\ a \end{bmatrix} (t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} s \\ v \\ a \end{bmatrix} (t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t)\,, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{6.10b}$$

$$\mathbf{0} = \boldsymbol{x}(t_{f,i}) - \boldsymbol{x}_{f,i}\,, \tag{6.10c}$$

$$\mathbf{0} \geq \left[v(t) - v_{\max}(s,t), \quad -v(t), \quad a(t) - a_{\max}, \quad a_{\min} - a(t) \right]^T\,, \tag{6.10d}$$

$$0 \leq p_{\text{risk}}(\mathcal{T}_{\text{long}}) \leq p_{\text{risk,max}}\,. \tag{6.10e}$$

The combined planning is addressed by a sampling-based approach. This means, the OCP (6.10) is first simplified and an analytical solution is determined for the reduced problem. Then, a solution for the original problem is recovered by sampling over the parameters of the analytical solution. To simplify the OCP (6.10), the inequality constraints (6.10d) and the risk constraint (6.10e) are disregarded. Furthermore, the risk term $w_{\text{risk}} \cdot p_{\text{risk}}(\mathcal{T}_{\text{long}})$ is neglected. While these constraints and the respective risk term would be difficult to handle analytically, they can be checked easily for a trajectory candidate. Furthermore, by using a sampling strategy to deal with these constraints and the risk term, the requirements that these parts of the OCP (6.10) have to meet are reduced to the requirement that they must be

evaluable point-wise and sufficiently smooth with regard to the sampling grid. This gives the planning scheme additional flexibility, since the risk model can be easily replaced and the dynamic constraints (6.10d), e.g., the speed limit profile $v_{\max}(s,t)$, can be arbitrarily complex. To check the constraints and to evaluate the risk term, the trajectory candidate has to be discretized only over time, i.e., over one dimension, and the trajectory candidates are directly calculated from the behavior options. Hence, no further sampling is necessary. The simplified OCP then can be solved analytically, as stated in Theorem 6.1.

Theorem 6.1 (Time-Weighted Jerk Optimal Trajectory (similarly pre-published in [MB19])) *The segment-wise defined trajectory*

$$\mathcal{T}^*_{long} : \begin{bmatrix} \boldsymbol{x}(t) \\ u(t) \end{bmatrix} = \begin{bmatrix} t^5 & t^4 & t^3 & t^2 & t & 1 \\ 5t^4 & 4t^3 & 3t^2 & 2t & 1 & 0 \\ 20t^3 & 12t^2 & 6t & 2 & 0 & 0 \\ 60t^2 & 24t & 6 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{bmatrix} + \gamma \begin{bmatrix} \frac{-3t^2-2w_t t+2(w_t+t)^2 \ln(w_t+t)}{4} \\ -t+(w_t+t)\ln(w_t+t) \\ \ln(w_t+t) \\ \frac{1}{w_t+t} \end{bmatrix} \tag{6.11}$$

with $t \in [0, \Delta t_{f,i}]$, $\Delta t_{f,i} = t_{f,i} - t_{f,i-1}$ *for the respective segments, and* γ *being a function of the coefficients* α_1, α_2, *and* α_3, *solves the optimal control problem*

$$u^*(t) = \arg\min_{u(t)} \left\{ \sum_{i=1}^{N} \int_0^{\Delta t_{f,i}} \frac{1}{2} \left(\frac{w_t - 1}{1+t} + 1 \right) u^2(t) dt + w_{t_f} \cdot \Delta t_{f,i}^2 \right\} \quad \textit{subject to} \tag{6.12a}$$

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} s \\ v \\ a \end{bmatrix}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot u(t)\,, \tag{6.12b}$$

$$\boldsymbol{0} = \left[\boldsymbol{x}(t_{f,i}) - \boldsymbol{x}_{f,i}, \quad \boldsymbol{x}(0) - \boldsymbol{x}_{f,i-1} \right], \quad \boldsymbol{x}_{f,0} = \boldsymbol{x}_0\,, \tag{6.12c}$$

where the coefficients α_1, α_2, α_3, α_4, α_5, α_6 *are uniquely determined through its boundary conditions* (6.12c) *and* $\Delta t_{f,i}$.

Proof. In the first step, it is shown that if there is a solution for each segment, which is optimal with respect to the argument of the sum term from (6.12a), the concatenation of these solutions yield the solution to the overall OCP (6.12). Due to the equality constraints (6.12c), the optimal trajectory must pass the target states $\boldsymbol{x}_{f,i}$ at time $t_{f,i}$. Hence, because of the Bellman principle [KS12, p. 259], the trajectory segment $\tilde{\mathcal{T}}^{*,N-1,N}_{\text{long}}, t \in [0, \Delta t_{f,N}]$ must be part of the optimal trajectory $\mathcal{T}^*_{\text{long}}$, and, thus, the optimal trajectory $\mathcal{T}^*_{\text{long}}$ contains the trajectory segment $\tilde{\mathcal{T}}^{*,N-2,N-1}_{\text{long}}, t \in [0, \Delta t_{f,N-1}]$ as well. This argument can be inductively

continued to $\tilde{\mathcal{T}}^{*,0,1}_{\text{long}}, t \in [0, \Delta t_{f,1}]$, where $\boldsymbol{x}_{f,0} = \boldsymbol{x}_0$ is the initial state. Therefore, the OCP (6.12) can be solved section-wise, where (6.12a) reduces to

$$u^*(t) = \arg\min_{u(t)} \left\{ \int_0^{\Delta t_{f,i}} \frac{1}{2} \left(\frac{w_t - 1}{1+t} + 1 \right) u(t)^2 \mathrm{d}t + w_{t_f} \cdot \Delta t_{f,i}^2 \right\} \tag{6.13}$$

for the respective segments. For (6.13) the Hamiltonian (see Section 3.2) is given by

$$H(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = \frac{1}{2} \left(\frac{w_t - 1}{1+t} + 1 \right) u^2 + \lambda_1 x_2 + \lambda_2 x_3 + \lambda_3 u \,. \tag{6.14}$$

From that, the co-states $\boldsymbol{\lambda}$ can be determined by the optimality condition (3.21b)

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial x} = \begin{bmatrix} 0 \\ -\lambda_2 \\ -\lambda_3 \end{bmatrix} \overset{\int dt}{\Longrightarrow} \begin{bmatrix} \lambda_1 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ -c_1 t + c_2 \\ \frac{1}{2} c_1 t^2 - c_2 t + c_3 \end{bmatrix}, \tag{6.15}$$

where $c_1, c_2, c_3 \in \mathbb{R}$ are integration constants. Since $u(t)$ is unconstrained, Pontryagin's Maximum Principle (3.21c) yields the input equation

$$0 = \frac{\partial H}{\partial u} \iff u = -\frac{1+t}{w_t + t} \lambda_3 \,. \tag{6.16}$$

Then, a polynomial division is carried out on (6.16), which leads to

$$u = \underbrace{-\frac{1}{2} c_1}_{=:\tilde{\alpha}_1} t^2 + \underbrace{(c_2 - \frac{1}{2} c_1 + \frac{1}{2} c_1 w_t)}_{=:\tilde{\alpha}_2} t + \underbrace{(w_t - 1)(c_2 + \frac{1}{2} c_1 w_t) - c_3}_{=:\tilde{\alpha}_3} + \frac{\gamma}{w_t + t} \,, \tag{6.17}$$

$$\gamma = w_t (c_3 - c_2 + w_t c_2 - \frac{1}{2} c_1 w_t + \frac{1}{2} c_1 w_t^2) - c_3 \,, \tag{6.18}$$

with γ being a function of the coefficients c_1, c_2, and c_3. These results then are inserted in (6.12b) and the system equation (6.12b) is integrated over time, which yields

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) = \begin{bmatrix} x_2, & x_3, & u \end{bmatrix}^T (t) \underset{\text{subs. } \tilde{\alpha}_{1\ldots6}}{\overset{\int dt}{\Longrightarrow}} \text{eq. (6.11)}, \tag{6.19}$$

where substituting $\alpha_1 = \frac{\tilde{\alpha}_1}{60}$, $\alpha_2 = \frac{\tilde{\alpha}_2}{12}$, $\alpha_3 = \frac{\tilde{\alpha}_3}{6}$, $\alpha_4 = \frac{\tilde{\alpha}_4}{2}$, $\alpha_5 = \tilde{\alpha}_5$, $\alpha_6 = \tilde{\alpha}_6$ results in (6.11).

Finally, the parameters $\alpha_1, \ldots, \alpha_6$ are determined for the respective sections using the equality constraints, which serve as boundary conditions. For that, γ is treated as a parameter first, while (6.11) is solved for $\alpha_1, \ldots, \alpha_6$. By treating γ as a parameter, the second term of (6.11) turns into a constant. Hence, (6.11) becomes a linear set of equations with respect to $\alpha_1, \ldots, \alpha_6$. Furthermore, the boundary conditions yield three equations for $\boldsymbol{x}(0)$ and three equations for $\boldsymbol{x}(\Delta t_{f,i})$, i.e., a linear set of six equations is available to solve six variables.

Therefore, the parameters can be uniquely determined in dependence of $\boldsymbol{x}(0)$, $\boldsymbol{x}(\delta t_{f,i})$, $\delta t_{f,i}$, w_t and γ. Moreover, it can be seen from (6.17) that the integration constants c_1, c_2, c_3 can be uniquely expressed as functions of α_1, α_2, and α_3. Thus, c_1, c_2, and c_3 can be substituted with functions of α_1, α_2, and α_3, where α_1, α_2, and α_3 only depend on w_t, γ, $\boldsymbol{x}(0)$, $\boldsymbol{x}(\Delta t_{f,i})$, and $\Delta t_{f,i}$. Accordingly, (6.18) allows to express γ as a function of w_t, $\boldsymbol{x}(0)$ and $\boldsymbol{x}(\Delta t_{f,i})$. Resubstituting γ into $\alpha_1, \dots, \alpha_6$ then allows to uniquely express them in terms of w_t, $\boldsymbol{x}(0)$, $\boldsymbol{x}(\Delta t_{f,i})$, and $\Delta t_{f,i}$. Since the equality constraints (6.12c) are accounted for as boundary conditions to solve the simplified OCP, the trajectory need to pass these states at time $t_{f,i}$. For this reason and due to the Bellman Principle, it is guaranteed that concatenating the segments calculated with (6.11) yields the optimal trajectory $\mathcal{T}^*_{\text{long}}$. □

Note that expressing the parameters $\alpha_1, \dots, \alpha_6$ in dependence of w_t, $\Delta t_{f,i}$, $\boldsymbol{x}(0)$ and $\boldsymbol{x}(\Delta t_{f,i})$ yields lengthy terms, which are omitted here for the sake of brevity. However, these expressions can be calculated easily using a computer algebra system such as MATLAB Symbolic Toolbox [The17]. A C++ function to calculate time-weighted jerk optimal trajectories is given in Appendix B. Furthermore, note that the proposed solution is a generalization of the sampling scheme presented by Werling et al. [WZKT10]. As can be seen from (6.18) and (6.11), for the special choice $w_t = 1$, i.e. no time-weighting, γ vanishes and thus $\mathcal{T}^*_{\text{long}}$ becomes a quintic polynomial. Hence, whenever either the behavior options are strongly restricted and thus the negative replanning effect is not expected to occur, or high dynamics are needed, $w_t = 1$ can be set. For example, between the PNR and the PGA of a trajectory, $w_t = 1$ is chosen, because in this case only one behavior option is left and all available resources can be concentrated on reacting to the surrounding road users.

6.2.5 Lateral Trajectory Tracking

Its basic idea is to abstract away the nonlinear vehicle dynamics so that the longitudinal trajectory planning can operate on a simple point mass model. Therefore, the lateral trajectory tracking takes the longitudinal trajectory $\mathcal{T}^*_{\text{long}}$ projected on the reference path $\mathcal{P}$ as input and determines the overall optimal trajectory $\mathcal{T}^*$ based on a nonlinear MPC approach and the nonlinear vehicle model (3.16) (see Section 3.2). The main goal of lateral trajectory tracking is to keep the CAV on its lane, while following the longitudinal trajectory as closely as possible. Furthermore, the lateral trajectory tracking has to account for the nonlinear vehicle model (3.16) and its constraints (3.15). Mathematically, this is expressed by the OCP

$$\boldsymbol{u}(t) = \arg\min_{\boldsymbol{u}(t)} \left\{ \int_0^{t_{f,\perp}} w_v(x_5 - v)^2 + w_a(u_2 - a)^2 + w_{u_1} u_1^2 + w_\perp(x_6 - d_{\perp,\mathrm{ref}})^2 + w_\varphi(x_3 - x_7)^2 \mathrm{d}t \right\} \tag{6.20a}$$

subject to

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}_{\mathrm{NLVM}}\big(\boldsymbol{x}(t), \boldsymbol{u}(t), t\big), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0\,, \tag{6.20b}$$

$$\boldsymbol{0} \geq \left[x_5 - v_{\max}(t), \quad -x_5, \quad a_\perp^2(t) - a_{\perp,\max}^2, \quad x_4^2 - \delta_{\max}^2 \right]^T, \tag{6.20c}$$

$$\boldsymbol{u}(t) \in \big[\, [u_{1,\min}, u_{1,\max}], \quad [a_{\min}, a_{\max}] \big]^T\,. \tag{6.20d}$$

In (6.20), the weights w_v, w_a, w_{u_1}, $w_\perp$, and w_φ are used to balance the optimization criteria velocity tracking, acceleration tracking, small steering wheel actuation, lateral distance tracking, and the tracking of the course angle, respectively. Furthermore, an offset $d_{\perp,\mathrm{ref}}$ can be added to the lateral distance to adjust how much the CAV tends to the middle of the road. The OCP (6.20) then is solved through the augmented Lagrangian method implemented in the numeric tool GRAMPC [EVM+19]. This yields the overall optimal trajectory $\mathcal{T}^*$.

6.2.6 Planning Algorithm

After the individual steps of the motion planning scheme have been presented in detail, the overall planning algorithm is described, which combines the individual parts to realize the motion planning of the CAV. Algorithm 6.1 outlines this planning algorithm. It takes the ego state $\boldsymbol{x}_{\mathrm{ego}}$, the route $\mathcal{R}$, and the local map $\mathcal{M}_{\mathrm{local}}$ as well as the ego and the external environment model as input. Furthermore, during initialization, the situation context is precomputed from the local map $\mathcal{M}_{\mathrm{local}}$ and the route $\mathcal{R}$ and is stacked into the list $\mathcal{L}_C$ (see Section 6.1). This list $\mathcal{L}_C$ is also passed to the algorithm as an input. The environment models are represented by the object lists $\mathcal{L}_{\mathrm{ext}}$ and $\mathcal{L}_{\mathrm{ego}}$, respectively, which also include the predicted trajectories of the respective objects. From this information, the algorithm determines the optimal trajectory $\mathcal{T}^*$ for the CAV in each replanning cycle to navigate through the current traffic situation.

The algorithm is repeated in each replanning cycle. First, the minimal cost $c_{\min}$ is set to infinity and the optimal longitudinal trajectory $\mathcal{T}^*_{\mathrm{long}}$ is initialized with the empty set to indicate that no valid solution is available. Thus, any valid trajectory improves the planning

Algorithm 6.1 Motion Planning Scheme (cf. [MHSB21])

Input: Local digital map $\mathcal{M}_{\text{local}}$, current ego state $\boldsymbol{x}_{\text{ego}}$, route $\mathcal{R}$,
ego and external object list with prediction $\mathcal{L}_{\text{ego}}$, $\mathcal{L}_{\text{ext}}$, list of precomputed situation contexts $\mathcal{L}_C$
Output: Optimal Trajectory $\mathcal{T}^*$

$c_{\text{min}} \leftarrow \infty$; $\mathcal{T}^*_{\text{long}} \leftarrow \emptyset$
$C \leftarrow$ SelectCurrentContext($\boldsymbol{x}_{\text{ego}}, \mathcal{L}_C$)
$C \leftarrow$ UpdateSituationContext($\mathcal{L}_{\text{ego}}, \mathcal{L}_{\text{ext}}, \boldsymbol{x}_{\text{ego}}, C$)
$\mathcal{L}_{\text{rel}} \leftarrow$ ExtractObjectsForRiskModel($C, \mathcal{L}_{\text{ego}}, \mathcal{L}_{\text{ext}}$)
for all $\mathcal{B}_k \in \mathcal{L}_{\mathcal{B}}$ **do** *% remember:* $C = (S_C, S_{\text{RA}}, \mathcal{L}_{\mathcal{B}})$
  $c_{\text{tmp}} \leftarrow \infty$; $\mathcal{T}_{\text{long}} \leftarrow \emptyset$
  $\boldsymbol{x}_0 \leftarrow$ ProjectToPath($\boldsymbol{x}_{\text{ego}}, \mathcal{B}_k$)
  for all $(\boldsymbol{x}_{f,i}, t_{f,i}) \in \mathcal{B}_k$ **do**
    $\tilde{\mathcal{T}}_{\text{long}} \leftarrow$ CalculateAnalyticTrajectoryCandidate($\boldsymbol{x}_0, \boldsymbol{x}_{f,i}, t_{f,i}$)
    if constraintCheck($\tilde{\mathcal{T}}_{\text{long}}$) **then**
      $c_{\text{tmp}} \leftarrow c_{\text{tmp}} + \text{cost}(\tilde{\mathcal{T}}_{\text{long}})$; $\mathcal{T}_{\text{long}} \leftarrow [\mathcal{T}_{\text{long}}, \tilde{\mathcal{T}}_{\text{long}}]$; $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}_f$
    else
      continue with next behavior option $\mathcal{B}_{k+1}$
    end if
  end for
  $p_{\text{risk}} \leftarrow$ EvaluateRisk($\mathcal{T}_{\text{long}}, p_{\text{rel}}$)
  if $p_{\text{risk}} < p_{\text{risk,max}}$ **and** $c_{\text{tmp}} + w_{\text{risk}} p_{\text{risk}} < c_{\text{min}}$ **then**
    $c_{\text{min}} \leftarrow c_{\text{tmp}} + w_{\text{risk}} p_{\text{risk}}$; $\mathcal{T}^*_{\text{long}} \leftarrow \mathcal{T}_{\text{long}}$
  end if
  if $\mathcal{T}^*_{\text{long}} \neq \emptyset$ **and** $I_{k+1} < I_k$ **then**
    break for loop
  end if
end for
$\mathcal{T}^* \leftarrow$ LateralTrajectoryTracking($\mathcal{T}^*_{\text{long}}, \mathcal{P}, \boldsymbol{x}_{\text{ego}}$)
return $\mathcal{T}^*$

result and is therefore considered. Furthermore, $\boldsymbol{x}_{\text{ego}}$ is used to select the current situation context C from $\mathcal{L}_C$, and C is updated through the perception information, as described in Section 6.1. Then, the situation context is used to extract the list of relevant objects $\mathcal{L}_{\text{rel}}$ from the environment model. Afterwards, the planning scheme explores the behavior options provided by C for the current situation. For this purpose, it iterates over the list of behavior options $\mathcal{L}_{\mathcal{B}}$ and, for each behavior option, constructs $\mathcal{B}_k$ the longitudinal trajectory $\mathcal{T}_{\text{long}}$ segment by segment along the respective target sequence $\langle(\boldsymbol{x}_f, t_f)\rangle_i$. The segments of the longitudinal trajectory are determined from the analytical solution (6.11) from Section 6.2.4 and checked for the dynamic constraints (6.10d). If the trajectory segment passes the constraint check, the cost of the segment is aggregated in c_{tmp}, the segment is added to the trajectory $\mathcal{T}_{\text{long}}$, and the end point of $\mathcal{T}_{\text{long}}$ is set as new starting state $\boldsymbol{x}_0$. Otherwise, the behavior option $\mathcal{B}_k$ is invalid and the algorithm moves on to $\mathcal{B}_{k+1}$.

As soon as the whole longitudinal trajectory is assembled, its residual risk $p_{\text{risk}}(\mathcal{T}_{\text{long}})$ is evaluated using the risk model. If the residual risk is below the maximum acceptable residual risk $p_{\text{risk, max}}$, the overall cost of $\mathcal{T}_{\text{long}}$ is calculated. The cost then is compared to the cost $c_{\min}$ of the best solution found so far. In case the cost is less, $\mathcal{T}^*_{\text{long}}$ and $c_{\min}$ are updated with the current longitudinal trajectory and its cost, respectively. This way, the solution quality is iteratively increased until a valid solution is found and all behavior options of the importance class, i.e., all behavior options with equal importance I, are explored. Finally, the optimal longitudinal trajectory $\mathcal{T}^*_{\text{long}}$ and the reference path $\mathcal{P}$ of the behavior option corresponding to $\mathcal{T}^*_{\text{long}}$ is fed into the lateral trajectory tracking (see Section 6.2.5) to obtain the overall optimal trajectory $\mathcal{T}^*$. The planning algorithm returns $\mathcal{T}^*$ after each replanning cycle, so that $\mathcal{T}^*$ can be applied to the actuators.

6.3 Simulation Studies for the Application Scenario

In this section, the proposed motion planning scheme is evaluated using simulations. First, the time-weighted jerk optimal trajectories from Section 6.2.4 are compared with ordinary jerk optimal trajectories to point out the difference in their behavior. Furthermore, it is shown by a simulative example how time-weighted jerk optimal trajectories mitigate jerky behavior induced by steadily changing behavior decisions. In the second step, the decision making of the motion planning scheme is evaluated using Monte-Carlo simulations. Monte-Carlo simulations allow to collect large amounts of statistical data under controlled conditions to study the effect of different parameter settings, which was not possible to this extent in real traffic.

6.3.1 Comparison Between Jerk Optimal and Time-Weighted Jerk Optimal Solution

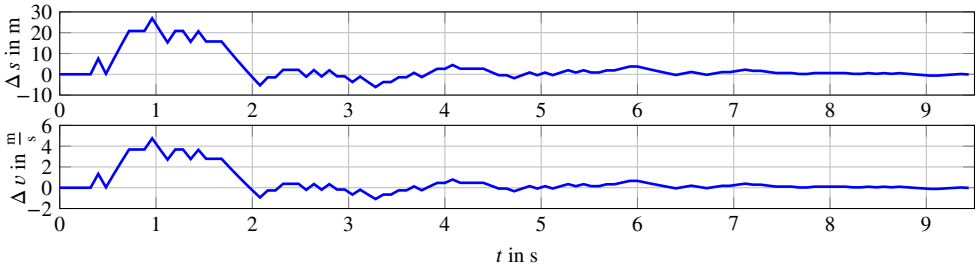
To illustrate the effect of the time weight, as an example, for $w_t = 1$ and $w_t = 100$, trajectories are calculated from $\boldsymbol{x}_0 = [0\,\text{m},\ 6.96\,\text{m}\,\text{s}^{-1},\ 0\,\text{m}\,\text{s}^{-2}]^T$ to $\boldsymbol{x}_f = [62.2\,\text{m},\ 7.0\,\text{m}\,\text{s}^{-1},\ 0\,\text{m}\,\text{s}^{-2}]^T$ at $t_f = 9.5\,\text{s}$. Note that despite similar start and target states, the timing is chosen such that the CAV has to adapt its motion to reach its target at the designated time. First, a single-shot planning is compared. Then, replanning is taken into account for strongly moving target states, as illustrated in Fig. 6.3(a). The simulation results are summarized in Fig. 6.3. While the effect of w_t is hardly visible in the position $s(t)$, the effect becomes apparent in the velocity $v(t)$, the acceleration $a(t)$, and most clearly in the jerk $u(t)$. It can be seen from Fig. 6.3(b) that, at the beginning of the time-weighted jerk optimal trajectory, the jerk is significantly closer to zero than that of the (unweighted) jerk optimal trajectory ($w_t = 1$).

Hence, the squared jerk is significantly lower. Therefore, the acceleration changes less at the beginning of the trajectory and the velocity profile appears delayed and less steep compared to the jerk optimal trajectory. This, however, changes towards the end of the trajectory, where the CAV following the time-weighted jerk optimal trajectory has to catch up. In this second phase, it shows that the jerk of the time-weighted jerk optimal trajectory stronger deviates from zero and, thus, the acceleration profile and the velocity profile get steeper compared to the jerk optimal trajectory. Hence, as expected, jerk that has to be applied soon is kept small at the cost of having to apply bigger jerks towards the end of the trajectory. This behavior appears more human-like, so w_t can be used to make the behavior of the CAV more comfortable to the passengers.

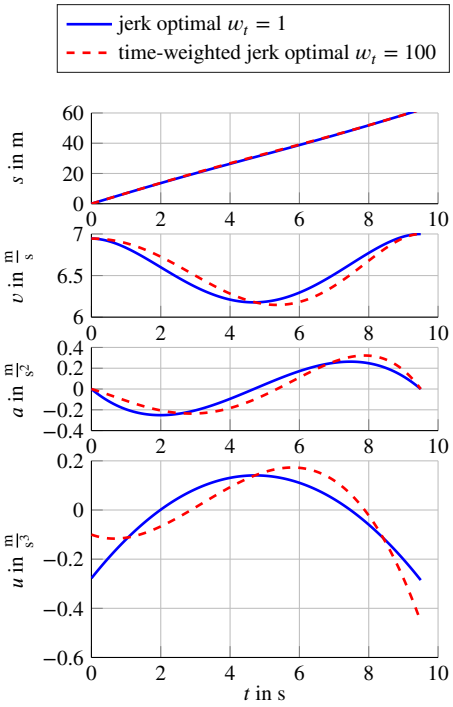
Moreover, as shown in Fig. 6.3(c), the time-weighting can also help to mitigate jerky behavior due to replanning. Therefore, as illustrated in Fig. 6.3(a), strongly changing target states are assumed, which decrease over time. For this setting, the CAV replans its trajectory cyclically, where the first action is applied to the actuators. However, when the CAV passes the PNR, which happens at about 7.5 s, the target state is locked and $w_t = 1$ is set for both trajectories. Thus, for the ending of the trajectories, the jerk is practically the same, where the small differences are due to slightly different initial states at the PNR. Besides, the big jerk at the ending of the planned trajectory can be explained by the strongly changing target states, which requires the CAV to adjust its ego state to a strongly differing target state over a short, receding horizon. It shows that, for the simulation example, the applied jerk is lower for the time-weighted jerk optimal trajectory. Therefore, the acceleration profile and the velocity profile appear smoother for the time-weighted jerk optimal trajectory than for the (unweighted) jerk optimal one. Particularly, the bumps in the jerk optimal trajectory are much smoother for the time-weighted jerk optimal trajectory. The static choice of w_t is a trade-off. Whenever high dynamics are required, e.g., if the CAV has to slow down for a curve, small values of w_t are reasonable. In contrast, e.g., for following another vehicle, where the speed is expected to remain mostly constant, high values of w_t can mitigate jerky behavior. Hence, an adaption strategy for w_t would be a conceivable next step.

6.3.2 Monte-Carlo Simulations on Decision Making

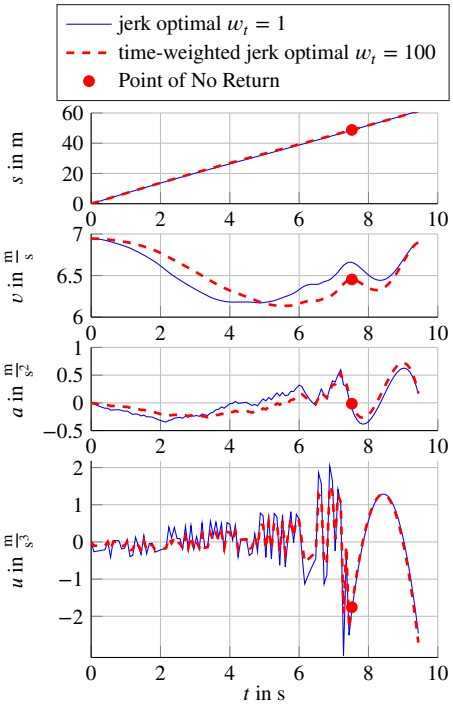
To evaluate the decision making of the proposed motion planning scheme, Monte-Carlo simulations are used to create a multitude of traffic scenarios and temporal evolutions of the respective situations. Three Monte-Carlo studies are conducted for the example of the application scenario (see Section 2.1). For each of theses studies, the behavior decisions are categorized into categories merging before the first car (*before*), merging into the gap between the two cars (*gap*), dynamically merging behind the second car (*behind*), merging behind the second car with gentle stop (*gentle stop*), and applying the fail-safe strategy (*fail-safe*). The individual counts of the respective categories related to the total number of simulation runs then yields the empirical probability that the motion planning scheme



(a) Noisy target state. Due to the prediction uncertainty, the initial target state varies by the position Δs offset and the velocity offset Δv, while the target acceleration is always $a = 0\frac{\text{m}}{\text{s}^2}$.



(b) Jerk optimal and time-weighted jerk optimal trajectory in comparison for a single shot.



(c) Comparison of actual jerk optimal and time-weighted jerk optimal trajectory after replanning.

Figure 6.3: Simulation results comparing time-weighted jerk optimal and jerk optimal trajectories (b) for a single-shot planning and (c) with replanning for heavily changing targets, as illustrated in (a). The trajectory consists of the states position s, velocity v, acceleration a and jerk u over time t. ([MB19], © 2019 IEEE)

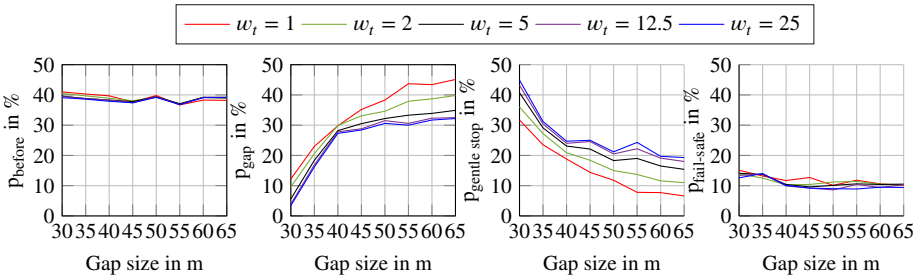
will choose the corresponding behavior option. Furthermore, a sweep over the gap size of the other vehicles at the start of the scenario is performed. Thus, the influence of the gap size to the behavior on the planning is studied. For all simulation runs, in total 47 000, an independent collision check was performed, where no collision was detected at all.

The first study is a basic simulation, which neglects the lateral dynamics of the CAV and uses an extended IDM [THH00] to represent the motion of the other two vehicles. Due to the simplifications, the simulation runs required moderate computation time and hence allowed for many simulation runs. Thus, the gap size was swept between 30 and 65 m in steps of 5 m, where 1000 runs per gap size and setting of the time weight $w_t \in \{1, 2, 5, 12.5, 25\}$, i.e. 40 000 simulation runs, are performed. To model variations in the vehicle's driving behavior, a Gaussian noise with standard deviation $\sigma_a = 0.25$ m s^{-2} is added to the IDM acceleration. These acceleration profiles then are applied to the Point-Mass-Model (3.14) to yield ground truth trajectories for the respective vehicles. From these ground truth trajectories, simulated measurements are derived by adding a further Gaussian noise with $\sigma_s = 0.25$ m to the positions of the vehicles. Furthermore, the maneuver timing is adjusted by varying the initial states of the vehicles. The initial velocity of the other vehicles is chosen as a Gaussian distribution with mean $\mu_v = 8.33$ m s^{-1} and variance $\sigma_v = 0.3$ m s^{-1}, while their initial position is chosen such that they arrive at the intersection within a time interval of $[5, 13]$ s. In contrast, the ego vehicle always starts at the same initial position, but the initial velocity of the ego vehicle is varied randomly in the interval $[6.94,\ 8.33]$ m s^{-1}. With these settings, the simulation focuses on the behavior decisions *before*, *gap*, and *gentle stop*, while *behind* was hardly possible.

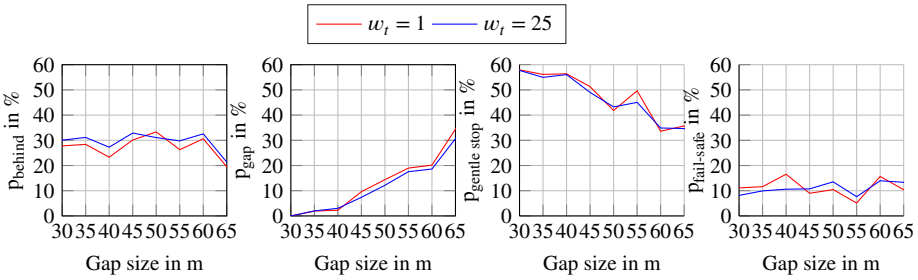
In contrast, the second and third studies are conducted using the deep-software-in-the-loop (DeepSIL) simulation framework proposed jointly by Jan Strohbeck and the author in [SMHB21]. In this approach, the motion of the other road users is modeled by the multi-trajectory-prediction network from [SBM$^+$20]. Thus, the behavior of the other road users is much more realistic compared to the extended IDM. To represent the more realistic behavior in numbers, the position error was evaluated in relation to ground truth data recorded in real traffic for the application scenario. It was found that vehicles simulated with the extended IDM had a position error that was a factor of 2.5 larger than the error of the vehicles simulated using the neural network [SMHB21]. Furthermore, the lateral dynamics are considered in the simulation. For the more sophisticated simulation, the computational burden increases, so the number of simulation runs had to be decreased. For the second study, about 2000 runs were carried out for $w_t = 1$ and $w_t = 25$, i.e., a total of about 4000 runs distributed over the different gap sizes. It focuses on the categories *gap*, *behind*. Hence, the probabilities of merging before the first vehicle was only about 3 % for gap size 30 m, 2 % for 35 m, and 1.5 % for 35 m, and 0 % for bigger gap sizes. The aim of the third study was to provide a direct comparison between the basic and the DeepSIL simulation. For that, the simulation scenario of the first study was reproduced with the DeepSIL for $w_t = 25$ and a total of 5000 runs. Like for the first study, *behind* was neither observed in the third one.

Figure 6.4 summarizes the simulation results. As can be seen, the first simulation study shows a strong influence of w_t to the decision making, where larger values of w_t yield a more defensive behavior of the CAV. However, for the second study, i.e., if the lateral dynamics are also considered, this effect reduces significantly and practically vanishes. For that reason and since the more realistic simulations require much more computation time per simulation step, in the third study, only $w_t = 25$ was considered. The reduced effect of w_t to the decision making can be explained by the lateral dynamics, which were considered in the second and third study. The lateral dynamics impose strict constraints on the longitudinal motion of the CAV. Thus, the solution space for both, $w_t = 1$ and $w_t > 1$, is strongly restricted and the solutions do not differ much. Furthermore, the lateral trajectory tracking counteracts the accumulation of postponed actuation beyond the actuation limits. Therefore, the motion planning has to resort less frequently to gentle stops. In summary, the less constrained the traffic scenario, the stronger is the influence of w_t on decision making.

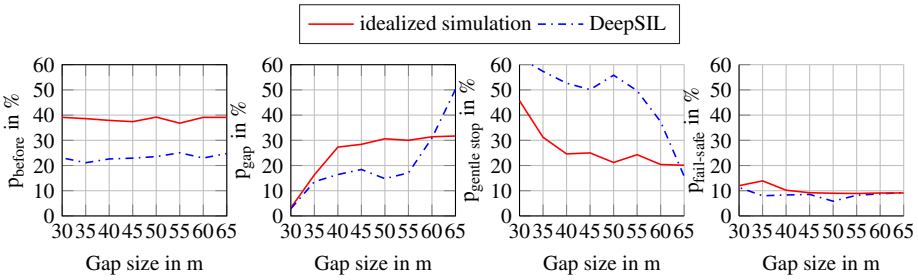
In all simulations and for all settings, and independent of the gap size, the fail-safe probability is always about 10 %. The fail-safe strategy in this case means that the CAV has wrongly predicted the scenario and thus has to brake with deceleration of 4 m s^{-2} at most. This is still quite endurable, and therefore fail-safe trajectories are factored in to some degree. The probability of *before* or *behind* is more or less constant in all simulations. This is due to the fact that these behavior options only require to adjust to one other road user and therefore do not depend on the gap size. However, as expected, the probability of *gap* increases with the gap size, while the probability of *gentle stop* decreases accordingly. Moreover, it can be observed that in the DeepSIL simulations, the merging probability is shifted to larger gap sizes. This complies with the expectation that due to the lateral constraints, the CAV has to slow down and thus needs larger traffic gaps to merge at an acceptable risk. Similarly, due to the enforced slow down of the CAV while turning right, the CAV usually has less velocity than the traffic on the main road and merging before the first car thus gets less likely.



(a) Idealized simulation using an extended IDM. 1000 runs are carried out for each setting of the initial gap size and w_t. The gap sizes are swept in 5 m steps. The initial states of the vehicles are chosen such that *behind* does not occur ([MB19], © 2019 IEEE).



(b) A total of about 4000 DeepSIL [SMHB21] runs are carried out for $w_t = 1$ and $w_t = 25$ each. The parameters are chosen such that p_{before} is negligible. The influence of w_t to the behavior is much smaller than in the idealized simulation.



(c) Idealized simulation and DeepSIL simulation in comparison. A total of 5000 DeepSIL [SMHB21] runs were performed for $w_t = 25$. The setup was chosen like in Fig. 6.4(a).

Figure 6.4: Statistics on merging decision in dependence of the traffic gap between the vehicles on the main road and the time weight w_t.

6.4 Experimental Validation

So far, the Monte-Carlo simulations from Section 6.3.2 provided statistical evidence that the decision making of the planning scheme yields reasonable decisions. However, since these evaluations were conducted based on simulations, the planning scheme has to be validated using real world experiments to confirm that the results gained from simulations are applicable to real-world scenarios. Impressions of the real-world experiments are summarized in the video [MHSB20]. The experiments had to be conducted under difficult conditions. In the project MEC-View, the validating experiments were planned from mid of March to mid of May 2020 [MEC20]. After the project, the pilot site was shut down and no further experiments were possible. Because the ITSS and the fusion algorithm on the MEC server had been revised until shortly before the experiments, the parameters of the overall system were largely unknown. Due to the COVID-19 pandemic and the respective lockdown, instead of two month of experimentation, only two days of experimentation with further two days of preparation remained. For that reason, not all aspects could be covered by the experiments. Instead, the experiments had to be prioritized.

For the experiments, safety had absolute priority, particularly, since the experiments were conducted in real traffic. Therefore, the CAV was parametrized conservatively, i.e., the maximum lateral acceleration was limited to $a_{\perp,\max} = 1.45\ \mathrm{m\ s^{-2}}$, while the longitudinal acceleration was restricted to $a \in [-4,\ 2]\ \mathrm{m\ s^{-2}}$. The CAV was supervised by a safety driver and a co-driver, where the safety driver focused on the driving and the co-driver monitored the systems. Furthermore, since neither the driver nor the co-driver could see into the occluded junction, the co-driver was in steady communication with additional track marshals. This way, a full-human fail-safe layer was implemented for the experiments. However, monitoring such a complex system is difficult and requires a period of familiarization even for experienced safety drivers. Hence, to keep the system manageable, all non-essential subsystems had to be shut down. Particularly, the reliability estimation from Section 5 was not activated, i.e., $p_{\mathrm{rel}} = 1$ was set. Furthermore, since according to the simulations shown in Fig. 6.4(c) the expected influence of w_t on the experiments is minor, $w_t = 1$ was chosen during the experiments. The experiments then focused on two main aspects: first, the lateral trajectory tracking is validated with respect to its lane-keeping property. Second, the longitudinal planning and its decision making are validated with respect to the maneuver timing of the CAV. As will be shown in Section 6.4.1 and Section 6.4.2, these two key properties are verified experimentally in multiple test drives. A positive side effect of having only the essential modules online is that the observed effect can be clearly attributed to the respective modules. For example, the sensor logs revealed that the RTK localization lost its precision in some of the recordings. Nonetheless, the motion planning was able to fulfill its task. Hence, the observed robustness can be attributed solely to the motion planning. It can be expected that the reliability estimation could make the system even more robust.

6.4.1 Lane Keeping Properties of the Motion Planning Scheme

To validate that the situation context is correctly extracted for the application scenario and that the lateral trajectory tracking robustly keeps the lane, the following metric is proposed: The position of the CAV is approximated by a bounding box with the corner points $\boldsymbol{x}_{FR}$, $\boldsymbol{x}_{FL}$, $\boldsymbol{x}_{RL}$, and $\boldsymbol{x}_{RR}$ for front right, front left, rear left, and rear right, respectively. The corner points $\boldsymbol{x}_{FR}$ and $\boldsymbol{x}_{RR}$ are represented in Frenet coordinates with respect to the right boundary $\mathcal{P}_{\mathrm{LB}}$ of the lane, while $\boldsymbol{x}_{FL}$ and $\boldsymbol{x}_{RL}$ are represented with respect to the left boundary $\mathcal{P}_{\mathrm{LB}}$:

$$\begin{aligned}(\boldsymbol{x}_{\mathrm{FL}},\, \boldsymbol{x}_{\mathrm{FR}},\, \boldsymbol{x}_{\mathrm{RR}},\, \boldsymbol{x}_{\mathrm{RL}}) &\xrightarrow{\text{Frenet}(\mathcal{P}_{\mathrm{LB}},\mathcal{P}_{\mathrm{RB}})} \\ &([s,d]^T_{\mathrm{FL}}, [s,d]^T_{\mathrm{FR}}, [s,d]^T_{\mathrm{RR}}, [s,d]^T_{\mathrm{RL}})\,. \end{aligned} \tag{6.21}$$

Then, the signs of the lateral coordinates are chosen such that positive values indicate that the CAV stays on the lane and the minimum lateral coordinate is selected as metric

$$d_{\mathrm{Lane}} = \min\left\{ -d_{\mathrm{FL}}, d_{\mathrm{FR}}, d_{\mathrm{RR}}, -d_{\mathrm{RL}} \right\}. \tag{6.22}$$

The metric d_{Lane} then is evaluated for 15 automated merging maneuvers performed at the junction from the application scenario. The results are presented in Fig. 6.5, where $s = 0$ is the stop line of the junction in the application scenario (see Section 2.1).

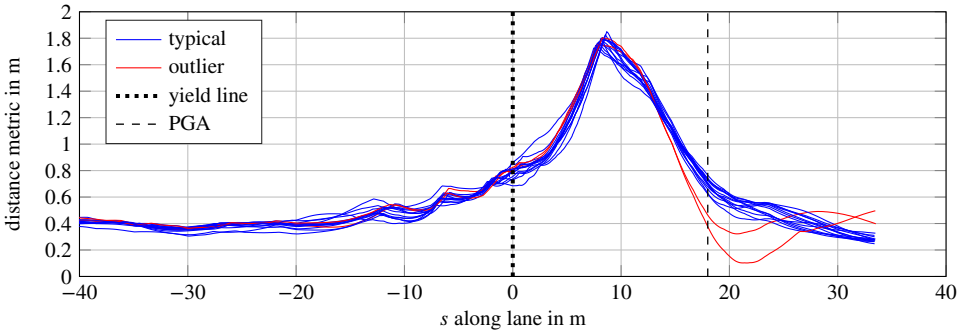


Figure 6.5: Evaluation of the lane keeping behavior of the CAV according to the metric (6.22). The metric describes the minimal distance of the CAV to the nearest lane boundary. The position s is referred to the yield line.

It shows that the metric is always positive. Yet, the value of d_{Lane} differs significantly along the road: While d_{Lane} is typically in the range of $20 - 50$ cm when driving along mostly straight roads, it grows significantly in the junction area, i.e., for $s \in [-10, 20]$ m, since

the road gets wider there. Hence, the typical behavior shows that the situation context is correctly extracted so that the CAV drives at a speed for which the lateral trajectory tracking can keep the CAV on the lane.

Besides, there are two outliers, one of them coming as close as about 10 cm to the nearest lane boundary. The reasons for such outliers are manifold: there might be timing issues when transferring the trajectory $\mathcal{T}^*$ to the subsidiary control, the accuracy of the state estimation could drop, or some distorting force, e.g., from the automatic transmission of the CAV using the clutch. For the present case, the log files of the CAV revealed that the RTK system lost its high-precision state which deteriorated the accuracy of the state estimation. Nevertheless, as can be seen from the still postive d_{Lane}, the lateral trajectory tracking is robust enough to keep the CAV on its lane. Hence, Fig. 6.5 shows that the lateral trajectory tracking and the context awareness module are successfully verified with respect to the lane keeping.

6.4.2 Maneuver Timing

In a next step, the maneuver timing is used as a criterion to validate three key aspects of the motion planning scheme: first, that it makes rational behavioral decisions; second, that it generates appropriate longitudinal trajectories that take into account the replanning effects; and third, that the risk model correctly processes the environment models and responds appropriately to changes in the traffic situation. For that, multiple fully automated test drives are recorded at the junction from the application scenario, from which the position-over-time trajectories $s(t)$ are reported. Within these test drives, five principle behavior types are expected: dynamically merging without other road users (*w/o*), merging before another road user (*before*), merging into the gap between two road users (*gap*), dynamically merging behind another road user (*behind*), and stopping at the yield line (*stop*). While stopping at the yield line according to the situation context comprises two importance classes, i.e. *gentle stop* and *fail-safe*, the other principle behavior types are subclasses of the importance class of dynamically merging.

Figure 6.6 shows three trajectories for each of the principle behavior types except *gap*. As can be seen, the trajectories for merging *w/o* and merging *before* are practically identical, with a variation in t_f of only about 100 ms over all six trajectories. The similarity is expected, since staying in the intersection area increases the heavily penalized residual risk. Moreover, the small variation in t_f can be anticipated, because merging without other road users is deterministic apart from the motion controller which regulates some distortions. The trajectories of merging *behind* show a significantly wider variation in shape and arrival time, where t_f varies by about 1.65 s. Furthermore, the trajectories corresponding to merging after a *stop*, show the expected horizontal segment at $s = s_{\text{yield}}$ and also vary in shape and arrival time. Here, t_f varies by 1.15 s. The trajectories corresponding to merging behind and those corresponding to merging after stopping both show a flatter shape as compared to

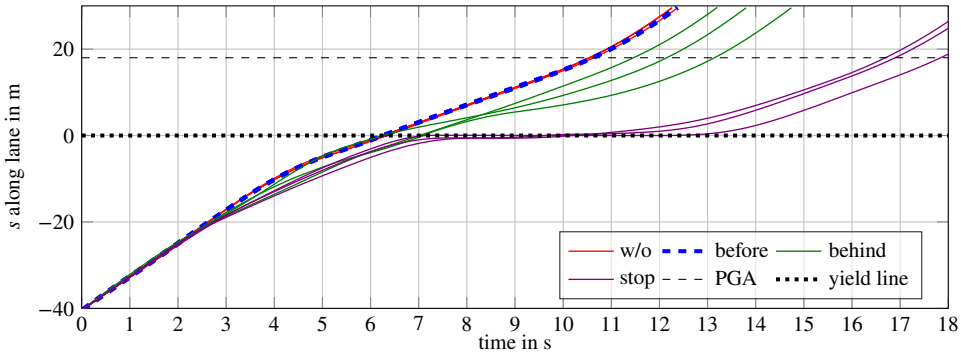


Figure 6.6: Trajectories for merging without other road users, before another road user, behind another road user, and merging after stopping at the yield line. The position s is referred to the yield line.

merging without other road users, since the CAV slows down early to synchronize its motion to the other road users.

In contrast to the aforementioned behavior types, for merging into a traffic gap, two behavior types are anticipated: class 1 that behaves like merging *behind* and class 2 that resembles the behavior of merging *before*. Class 1 is expected to occur if the CAV synchronizes its motion to the first road user to merge behind it into the gap. In contrast, class 2 is anticipated, if it is clear from the beginning that the CAV will merge after the first road user and mainly times its motion to merge before the road user at the end of the gap. Figure 6.7 shows five trajectories from class 1 plotted in green and four trajectories from class 2 plotted in red. Moreover, a trajectory corresponding to merging before another road user is plotted as reference in dashed blue. As expected, both classes occur, where class 1 shows a broad variation in shape and arrival time, while the trajectories from class 2 strongly resemble the trajectory of merging before another road user. Yet, the arrival times differ slightly more (by about 0.5 s), which can be explained by some remaining adjustments to the road user driving in front of the gap. The experimental results therefore agree with the expectations and, thus, further support the validity of the motion planning scheme.

Finally, the motion planning scheme is validated with respect to its capabilities to dynamically react to a changing environment. For that, traffic scenarios are selected in which two vehicles approach the junction, but the first one turns right. Thus, the first vehicle opens up space for the CAV to merge. Since the CAV strongly interacts with the road users when planning the corresponding maneuvers, it can be expected that there will be a broad variation in the shape and arrival time. Depending on when the first vehicle turns, trajectories very similar to merging before another road user are anticipated, whereas other trajectories might look similar to merging behind another road user if the CAV has already synchronized

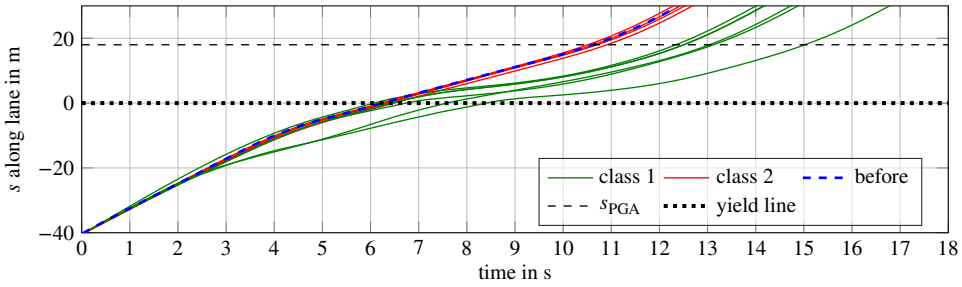


Figure 6.7: Timing for merging into a traffic gap. Class 1 resembles merging behind a road user, while class 2 behaves similarly to merging before another one. The position s is referred to the yield line.

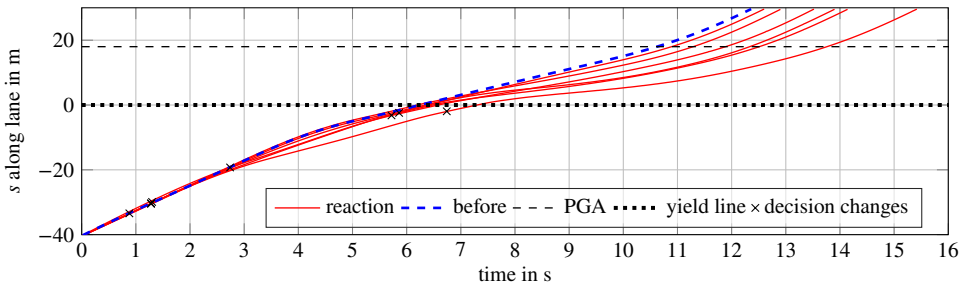


Figure 6.8: Trajectories corresponding to merging maneuvers with changing environment model. The position s is referred to the yield line.

its motion to the traffic gap. Figure 6.8 contains six trajectories showing how the CAV adapts its motion to a changed environment. Decision changes are marked as black crosses. Furthermore, an example of merging before another road user is plotted in dashed blue for comparison. As anticipated, the arrival time t_f and the shape strongly differ between the individual trajectories. For the given examples, t_f varies by about 2.75 s. This confirms the anticipations and thus validates that the CAV can quickly react to a changing environment.

In total, the evaluation of the maneuver timing validates the proposed motion planning scheme. Albeit the experiments have been conducted for the application scenario only, the application scenario is sufficiently complex and features a broad variation of challenges typical to multiple urban traffic scenarios. Hence, it can be expected that the results can be transferred to further traffic scenarios. The experiments have shown exemplarily that the CAV can quickly react to a changing environment, that it handles other road users reasonably, that the planning scheme yields appropriate trajectories, and that it makes reasonable behavior decisions.

6.5 Summary

Automated driving in urban areas presents significant challenges, such as safely interacting with other road users, accounting for uncertainty, adequately timing the maneuvers, and appropriately dealing with occlusions from other road users or static obstacles, to name a few. All these challenges have to be addressed for a multitude of possible traffic scenarios under strict real-time constraints, while providing passenger comfort at the same time. The proposed motion planning scheme addresses these challenges based on an optimization-based approach that holistically solves decision making and longitudinal planning to assure adequate maneuver timing. For the longitudinal planning, an OCP for time-weighted jerk optimal trajectories was formulated and solved analytically. Thus, the available behavior options formalized through the situation context of the proposed context awareness concept can directly be turned into longitudinal trajectories. With the context awareness concept, the general motion planning problem can be simplified and parametrized to a broad variety of traffic scenarios, which allows to efficiently calculate optimal trajectories in real-time. Sampling over the behavior options then leads to the optimal longitudinal trajectory, which is used by the lateral trajectory tracking as reference to calculate the overall optimal trajectory. The lateral trajectory tracking is formulated as nonlinear MPC and solved with the numerical tool GRAMPC. Occlusions are resolved by an external environment model communicated to the CAV via V2X communication. Uncertainties and the appropriate interactions with other road users are governed by a risk model that takes into account the reliability of the external information. Monte-Carlo simulations show that the proposed motion planning scheme yields reasonable behavior decisions. Furthermore, it was demonstrated with real-world experiments by the example of the application scenario that the CAV robustly keeps the lane, adequately reacts to other road users, and robustly realizes the intended maneuver with adequate timing.

Chapter 7

Conclusion and Future Work

In this thesis, a novel motion planning scheme has been proposed, which optimizes the trajectory of a CAV over multiple behavior options with respect to safety and passenger comfort. The motion planning was formulated in terms of two OCPs, one solving the decision making and longitudinal trajectory planning holistically, and one for the lateral trajectory tracking. The latter was solved by a nonlinear MPC approach and the numerical tool GRAMPC. The longitudinal problem was solved by a sampling-based approach that iterates over the available behavior options, while the trajectories corresponding to the respective options are calculated with a closed-form analytical solution derived within this thesis. The available behavior options as well as the parametrization of the OCPs according to the respective traffic situation is provided by the situation context, a formalized representation of the traffic scenario derived from an HD map and perception information. To handle measurement and prediction uncertainties within the perception data, a risk model was proposed that settles upon set-based methods. This risk model then determines the residual risk of an intended trajectory, which is accounted for as terminal cost. Furthermore, the residual risk is limited to a maximum residual risk. Thus, the motion planning comes with general safety guarantees in a probabilistic sense, while strict safety guarantees are inherited from the set-based methods for the special case that the planned trajectory is fully included within an ISS.

Occlusions, which frequently occur in urban automated driving are resolved with external perception. Thus, the CAV can efficiently realize typical maneuvers in urban traffic such as merging into an occluded yield junction. Since it is vital to the safety of the CAV that external perception information is only used for planning if it is sufficiently reliable, an online reliability estimation scheme has been proposed. This is used to assess the reliability of the V2X communication link as well as the external environment model. The reliability of the external environment model is assessed with two consistency tests and two plausibility tests: the prediction test, the map test, the ego localization test, and the ego perception test. Each of these tests analyzes a particular aspect of reliability. It is shown that the reliability estimation, in addition to the use for motion planning, can also be used in a V-MAS to establish trust

between the agents. Simulations show that this trust management mechanism can be used to identify misbehaving agents and even collaborative attackers and remove them from the V-MAS. Furthermore, it is shown on a large-scale simulation example of the whole traffic of Cologne that the approach scales well even to large-scale V-MAS.

Since typically only very few measurements are available, the reliability estimation scheme has to operate on insufficient statistics. Therefore, classical probability theory was not applicable. Instead, the reliability estimation scheme was based on the mathematical theory *Subjective Logic* specially designed to cope with insufficient statistics. To provide a better understanding of SL, which was up to now only described very theoretical in literature, an urn model intuition was proposed that links SL with Pólya's urn scheme. For this urn model, representations of the SL opinion as well as representations of the most important operators of SL, namely the CBF, the WABF, and the trust discounting operator, have been derived. Furthermore, the connection of the CBF operator to Bayesian inference and other properties of the urn model have been proven mathematically.

Both, the motion planning and the reliability estimation have been evaluated in real-world experiments by the example of a challenging application scenario, where the CAV dynamically merges into an occluded urban yield intersection using the external environment model calculated by a MEC server from the data of an ITSS. It was shown in real traffic that the CAV effectively uses the external perception information from the ITSS to resolve the occlusions, adequately reacts to other road users, makes reasonable decisions, and plans comfortable trajectories.

While in this thesis the proposed general motion planning concept was evaluated only on one application scenario, it is conceivable to apply the concept to many other situations. Hence, in future work, the proposed concept can be evaluated for further traffic scenarios, such as on-ramping, left turns, or signalized intersections. Moreover, in this thesis, the information flow was unidirectional from the external perception to the CAV. However, it has been shown in Section 5.3 that with the proposed reliability estimation schemes, trust can be established within a V-MAS. Furthermore, the proposed parallel use of the external and ego perception in the motion planning prevents loop-closing and thus self-fulfilling prophecies of ghost objects by design. Therefore, it is a reasonable next step to introduce a feedback channel, where the CAVs share their perception with other road users. If the agents in a V-MAS can trust each other, cooperative control strategies can be applied to further enhance the efficiency of connected automated driving. For example, trust management can be used to secure cooperative adaptive cruise control [WWM+20]. With the new urn scheme representation of SL, SL has been expressed mostly in terms of set theory. Thus, many further properties of SL can be proven using theorems from set theory. Furthermore, the new intuition of SL allows to extend the application of SL to many more fields, such as self-assessment in information fusion, where a first step was already done in [GMBD20].

Appendix A

Augmented Lagrangian Approach and Numerical Optimization Strategies

In this appendix section, details on the augmented Lagrangian approach are discussed. The approach is the basis for the optimization tool GRAMPC [EVM+19; KG14], which is used in this thesis for motion planning. Moreover, details on numerical optimization strategies are presented. Particularly, details on the GRAMPC optimization algorithm are given.

A.1 Augmented Lagrangian Approach

In Section 3.2, the equality and inequality constraints (3.13c) and (3.13d) have been neglected. To account for these constraints, first, the inequality constraints are transformed into equality constraints using slack variables $\boldsymbol{\xi}$, $\boldsymbol{\xi}_T \geq \mathbf{0}$:

$$\mathbf{0} = \check{\boldsymbol{g}}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\xi}(t), t) = \begin{bmatrix} \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \\ \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) + \boldsymbol{\xi} \end{bmatrix}, \tag{A.1a}$$

$$\mathbf{0} = \check{\boldsymbol{g}}_T(\boldsymbol{x}(t_f), \boldsymbol{\xi}_T, t_f) = \begin{bmatrix} \boldsymbol{g}_T(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \\ \boldsymbol{h}_T(\boldsymbol{x}(t_f), t_f) + \boldsymbol{\xi}_T \end{bmatrix}. \tag{A.1b}$$

Then, the augmented Lagrangian approach [EVM+19] is applied to integrate the equality constraints into the cost function:

$$\begin{aligned} \check{J}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\mu}(t), \boldsymbol{\mu}_T, \upsilon, \upsilon_T, \boldsymbol{\xi}(t), \boldsymbol{\xi}_T, t_f) = \check{V}(\boldsymbol{x}(t_f), \boldsymbol{u}(t), \boldsymbol{\mu}_T, \upsilon_T, \boldsymbol{\xi}_T) \\ + \int_0^{t_f} \check{l}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\xi}(t), \boldsymbol{\mu}(t), \upsilon, t)\, \mathrm{d}t\,. \end{aligned} \tag{A.2}$$

In (A.2), the modified terminal cost function $\check{V}(\cdot)$ and the modified integral cost $\check{l}(\cdot)$ are given by

$$\begin{aligned}\check{V}(\boldsymbol{x}(t_f), \boldsymbol{\mu}_T, \boldsymbol{\upsilon}_T, \boldsymbol{\xi}_T) &= V(\boldsymbol{x}(t_f), t_f) + \boldsymbol{\mu}_T^T \check{\boldsymbol{g}}_T(\boldsymbol{x}(t_f), t_f, \boldsymbol{\xi}_T) \\ &\quad + \check{\boldsymbol{g}}_T^T(\boldsymbol{x}(t_f), t_f, \boldsymbol{\xi}_T) \cdot \Upsilon_T \cdot \check{\boldsymbol{g}}_T(\boldsymbol{x}(t_f), t_f, \boldsymbol{\xi}_T)\,, \qquad \text{(A.3a)} \\ \check{l}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\xi}(t), \boldsymbol{\mu}(t)\boldsymbol{\upsilon}, \boldsymbol{\xi}_T, t) &= l(\boldsymbol{x}(t), \boldsymbol{u}(t), t) + \boldsymbol{\mu}^T \check{\boldsymbol{g}}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\xi}(t), t) \\ &\quad + \check{\boldsymbol{g}}^T(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\xi}(t), t) \cdot \Upsilon \cdot \check{\boldsymbol{g}}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\xi}(t), t)\,, \qquad \text{(A.3b)}\end{aligned}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_T$ are the Lagrange multipliers for the constraints and the terminal constraints, respectively. Furthermore, $\boldsymbol{\upsilon}$ as well as $\boldsymbol{\upsilon}_T$ with $\Upsilon = \mathrm{diag}(\boldsymbol{\upsilon})$ and $\Upsilon_T = \mathrm{diag}(\boldsymbol{\upsilon}_T)$ are penalty weights describing how strong a violation of the constraints is penalized. Here, $\mathrm{diag}(\cdot)$ spans the diagonal matrix from its vector argument. Thus, the OCP turns into the max-min problem

$$\begin{aligned}\boldsymbol{u}^*(t) &= \arg\min_{\boldsymbol{u}} \max_{\boldsymbol{\mu}, \boldsymbol{\mu}_T} \min_{\boldsymbol{u}, t_f, \boldsymbol{\xi}, \boldsymbol{\xi}_T} \left\{ \check{J}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\mu}(t), \boldsymbol{\mu}_T, \boldsymbol{\upsilon}, \boldsymbol{\upsilon}_T, \boldsymbol{\xi}(t), \boldsymbol{\xi}_T, t) \right\} \qquad \text{(A.4a)} \\ \text{subject to } \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\,, \quad \boldsymbol{x}^*(0) = \boldsymbol{x}_0\,, \qquad \text{(A.4b)} \\ \mathbf{0} &\leq \boldsymbol{\xi}(t),\, \boldsymbol{\xi}_T(t)\,, \quad \boldsymbol{u}(t) \in [\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}]\,, \quad t_f \in [t_{f,\min}, t_{f,\max}]\,. \qquad \text{(A.4c)}\end{aligned}$$

The strictly convex, quadratic structure of the max-min problem with respect to the slack variables then allows for a closed form solution of the max-min problem with respect to the slack variables [EVM+19]

$$\begin{aligned}\boldsymbol{\xi} &= \mathbf{max}\{\mathbf{0},\, -\Upsilon_{\boldsymbol{h}}^{-1} \boldsymbol{\mu}_{\boldsymbol{h}} - \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}, t)\}\,, \qquad \text{(A.5a)} \\ \boldsymbol{\xi}_T &= \mathbf{max}\{\mathbf{0},\, -\Upsilon_{\boldsymbol{h}_T}^{-1} \boldsymbol{\mu}_{\boldsymbol{h}_T} - \boldsymbol{h}_T(\boldsymbol{x}(t_f), t_f)\}\,, \qquad \text{(A.5b)}\end{aligned}$$

where $\mathbf{max}\{\cdot\}$ has to be understood element-wise and $\Upsilon_{\boldsymbol{h}}$, $\Upsilon_{\boldsymbol{h}_T}$, $\boldsymbol{\mu}_{\boldsymbol{h}}$, $\boldsymbol{\mu}_{\boldsymbol{h}_T}$ describe the parts of the respective quantities that relate to the inequality constraints. Hence, the max-min problem (A.4) is simplified to

$$\begin{aligned}\boldsymbol{u}^*(t) &= \arg\min_{\boldsymbol{u}} \max_{\boldsymbol{\mu}, \boldsymbol{\mu}_T} \min_{\boldsymbol{u}, t_f} \left\{ \tilde{J}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\mu}(t), \boldsymbol{\mu}_T, \boldsymbol{\upsilon}, \boldsymbol{\upsilon}_T, t_f) \right\} \qquad \text{(A.6a)} \\ \text{subject to } \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\,, \quad \boldsymbol{x}^*(0) = \boldsymbol{x}_0\,, \qquad \text{(A.6b)} \\ \boldsymbol{u}(t) &\in [\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}]\,, \quad t_f \in [t_{f,\min}, t_{f,\max}]\,. \qquad \text{(A.6c)}\end{aligned}$$

In (A.6), the cost function $\tilde{J}(\cdot)$ is

$$\tilde{J}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\mu}(t), \boldsymbol{\mu}_T, \boldsymbol{\upsilon}, \boldsymbol{\upsilon}_T, t_f) = \tilde{V}(\boldsymbol{x}(t_f), \boldsymbol{\mu}_T, \boldsymbol{\upsilon}_T) + \int_0^{t_f} \tilde{l}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\mu}(t) \mathrm{d}t, \qquad \text{(A.7)}$$

where the terminal cost and the integral costs are

$$\begin{aligned}\tilde{V}(\boldsymbol{x}(t_f), \boldsymbol{\mu}_T, \boldsymbol{\upsilon}_T) = V(\boldsymbol{x}(t_f), t_f) + \boldsymbol{\mu}_T^T \tilde{\boldsymbol{g}}_T(\boldsymbol{x}(t_f), t_f) \\ + \tilde{\boldsymbol{g}}_T^T(\boldsymbol{x}(t_f), t_f)\, \Upsilon_T\, \tilde{\boldsymbol{g}}_T(\boldsymbol{x}(t_f), t_f)\,, \qquad \text{(A.8a)}\\ \tilde{l}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\mu}(t), \boldsymbol{\upsilon}, t) = l(\boldsymbol{x}(t), \boldsymbol{u}(t), t) + \boldsymbol{\mu}^T \tilde{\boldsymbol{g}}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \\ + \tilde{\boldsymbol{g}}^T(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\, \Upsilon\, \tilde{\boldsymbol{g}}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\,. \qquad \text{(A.8b)}\end{aligned}$$

Accordingly, the Hamiltonian, the equality and inequality constraints are

$$\tilde{H}(\boldsymbol{x}(t), \boldsymbol{u}(t), \lambda(t), \boldsymbol{\mu}(t), \boldsymbol{\upsilon}, t) = \tilde{l}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\mu}(t), \boldsymbol{\upsilon}, t) + \lambda(t)^T\, \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\,, \qquad \text{(A.9a)}$$

$$\tilde{\boldsymbol{g}}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) = \begin{bmatrix} \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \\ \mathbf{max}\{\boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t), t), -\Upsilon_{\boldsymbol{h}}^{-1} \boldsymbol{\mu}_{\boldsymbol{h}}\} \end{bmatrix}, \qquad \text{(A.9b)}$$

$$\tilde{\boldsymbol{g}}_T(\boldsymbol{x}(t_f), \boldsymbol{\xi}_T, t_f) = \begin{bmatrix} \boldsymbol{g}_T(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \\ \mathbf{max}\{\boldsymbol{h}_T(\boldsymbol{x}(t_f), t_f), -\Upsilon_{\boldsymbol{h}_T}^{-1} \boldsymbol{\mu}_{\boldsymbol{h}_T}\} \end{bmatrix}. \qquad \text{(A.9c)}$$

Under the assumption of strong duality, the solution of (A.6) solves the OCP (3.13).

A.2 Numerical Optimization Strategies

In the ideal case, the OCP (3.13) can be completely solved analytically. However, most OCPs that occur in practice are too complex to be solved in a closed form solution. Hence, in this thesis, two ways are pursued to solve such an OCP. The first approach is to simplify the OCP (3.13) until it can be solved analytically up to a set of remaining parameters. Then, a grid search is conducted on the remaining parameters and the resulting solution candidate is evaluated with respect to the full OCP (3.13) for each parameter combination. Finally, from all solution candidates that fulfill the constraints (3.13c) and (3.13d), the candidate that comes with minimal cost is selected. This approach is termed the sampling-based approach, since the grid search involves the sampling over a parameter space. A prominent example for such a sampling-based approach is the method presented by Werling et al. [WZKT10]. Sample-based approaches are especially advantageous if the search space is limited, the optimization horizon is large, or if the simplifications covered by the sampling would be difficult to handle in a numerical optimization. For example, a sampling-based approach can ignore non-differentiable constraints like (3.13c) or (3.13d) first and handle them afterwards by point-wise evaluation during the sampling process.

The second way is using the augmented Lagrangian approach to convert the full OCP (3.13) into the max-min-problem (A.6) and solve it by numerical methods. This approach is beneficial if no adequate analytical solution can be found or if the sampling space of a

sampling-based approach would be large. Furthermore, to apply the numerical strategy, the OCP (3.13) has to be fully differentiable to use gradient-based optimization strategies. Such gradient-based optimization strategy is efficiently implemented in the open-source software framework GRAMPC [EVM+19; KG14]. The underlying optimization algorithm is summarized in Algorithm A.1. For the sake of brevity, some implementation details, such as the utilized numerical integration methods, the line search, the convergence check or the update of the penalty weights through the functions $\boldsymbol{\zeta}(\cdot)$ and $\boldsymbol{\zeta}_T(\cdot)$, are neglected in this summary. Further details can be found in [EVM+19].

In Algorithm A.1, the inner minimization is carried out with the projected gradient method from [KG14] that comprises the component-wise projection functions

$$\boldsymbol{\psi}(\boldsymbol{u}(t)) = \begin{cases} u_{i,\min}, & u_i < u_{i,\min}, \\ u_i, & u_i \in [u_{i,\min}, u_{i,\max}], \\ u_{i,\max}, & u_i > u_{i,\max}, \end{cases} \quad \boldsymbol{\psi}(t_f) = \begin{cases} t_{f,\min}, & t_f < t_{f,\min} \\ t_f, & t_f \in [t_{f,\min}, t_{f,\max}] \\ t_{f,\max}, & t_f > t_{f,\max} \end{cases}. \tag{A.10}$$

Each iteration of the projected gradient method basically consists of six steps: the backward integration of the adjoint system, the calculation of the gradients, a line search that determines the step size, the update of the controlled variables, the forward integration of the system, and a convergence check. In turn, the outer maximization is conducted through a common steepest ascent and consists of three steps: the update of the Lagrange multipliers, the update of the penalty weights, and a convergence check.

Algorithm A.1 and its implementation GRAMPC has been reported to work extremely fast [GGD18]. This thesis focuses on the development of prototypes. Thus, runtime optimization and deployment to embedded devices is out of the scope for this thesis. Nonetheless, these results show the huge potential of the approach for a later serial product. Furthermore, for the core method without the augmented Lagrangian extension, good stability and convergence properties have been proven [GK10; Gra12; Gra15]. Thus, the gradient-based optimization strategy is well-suited to be used for the lateral trajectory tracking (see Section 6.2.5).

Algorithm A.1 Augemented Lagrangian Optimization [EVM+19]

Input: System description: $\boldsymbol{f}(\cdot)$, $\boldsymbol{g}(\cdot)$, $\boldsymbol{g}_T(\cdot)$, $\boldsymbol{h}(\cdot)$, $\boldsymbol{h}_T(\cdot)$,
and their derivatives $\frac{\delta \boldsymbol{f}(\cdot)}{\delta \boldsymbol{x}}$, $\frac{\delta \boldsymbol{f}(\cdot)}{\delta \boldsymbol{u}}$, $\frac{\delta \boldsymbol{g}(\cdot)}{\delta \boldsymbol{x}}$, $\frac{\delta \boldsymbol{g}(\cdot)}{\delta \boldsymbol{u}}$, $\frac{\delta \boldsymbol{g}_T(\cdot)}{\delta \boldsymbol{x}}$, $\frac{\delta \boldsymbol{g}_T(\cdot)}{\delta \boldsymbol{u}}$, $\frac{\delta \boldsymbol{h}(\cdot)}{\delta \boldsymbol{x}}$, $\frac{\delta \boldsymbol{h}(\cdot)}{\delta \boldsymbol{u}}$, $\frac{\delta \boldsymbol{h}_T(\cdot)}{\delta \boldsymbol{x}}$, $\frac{\delta \boldsymbol{h}_T(\cdot)}{\delta \boldsymbol{u}}$,
system bounds $[\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}]$, $[t_{f,\min}, t_{f,\max}]$, cost functionals $V(\cdot)$, $l(\cdot)$,
and their derivatives $\frac{\delta V(\cdot)}{\delta \boldsymbol{x}}$, $\frac{\delta V(\cdot)}{\delta t}$, $\frac{\delta l(\cdot)}{\delta \boldsymbol{x}}$, $\frac{\delta l(\cdot)}{\delta \boldsymbol{u}}$,
initial values $\boldsymbol{x}_0$, $\boldsymbol{u}_0(t)$, $t_{f,0}$, $\boldsymbol{\mu}_0$, $\boldsymbol{\mu}_{T,0}$, $\boldsymbol{\upsilon}_0$, $\boldsymbol{\upsilon}_{T,0}$,
and algorithm parameters $i_{\text{mult,max}}$, $i_{\text{grad,max}}$
Output: Optimal trajectory $\mathcal{T}^* : [\boldsymbol{x}^*(t)^T, \boldsymbol{u}^*(t)^T]^T,\ t \in [0, t_f^*]$

$\boldsymbol{x}(t) \leftarrow \boldsymbol{x}_0 + \int_0^{t_f} \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}_0(t), t)\mathrm{d}t$
for $i_{\text{mult}} = 1$ **to** $i_{\text{mult,max}}$ **do**
  for $i_{\text{grad}} = 1$ **to** $i_{\text{grad,max}}$ **do**
    % Backward integration of the adjoint system
    $\boldsymbol{\lambda}(t) \leftarrow \frac{\delta \tilde{V}}{\delta t}(\boldsymbol{x}(t_f), \boldsymbol{\mu}_T, \boldsymbol{\upsilon}_T, t)\Big|_{t=t_f} + \int_{t_f}^{t} \frac{\delta \tilde{H}}{\delta \boldsymbol{x}}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \boldsymbol{\lambda}(\tau), \boldsymbol{\mu}, \boldsymbol{\upsilon}, \tau)\mathrm{d}\tau$
    % Calculate gradients
    $\nabla \boldsymbol{u}(t) \leftarrow \frac{\delta \tilde{H}}{\delta \boldsymbol{u}}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\lambda}(t), \boldsymbol{\mu}, \boldsymbol{\upsilon}, t)$
    $\nabla t_f \leftarrow \tilde{H}(\boldsymbol{x}(t_f), \boldsymbol{u}(t_f), \boldsymbol{\lambda}(t_f), \boldsymbol{\mu}(t_f), \boldsymbol{\upsilon}, t_f) + \frac{\delta \tilde{V}}{\delta t}(\boldsymbol{x}(t_f), \boldsymbol{\mu}_T, \boldsymbol{\upsilon}_T, t)\Big|_{t=t_f}$
    % Line Search
    $\eta^* \leftarrow \arg\min_\eta \left\{\tilde{J}(\ \boldsymbol{x}(t), \boldsymbol{\psi}(\boldsymbol{u}(t) - \eta \nabla \boldsymbol{u}(t)), \boldsymbol{\mu}(t), \boldsymbol{\mu}_T, \boldsymbol{\upsilon}, \boldsymbol{\upsilon}_T, \psi(t_f - \eta \nabla t_f)\ )\right\}$
    $\boldsymbol{u}(t) \leftarrow \boldsymbol{\psi}(\boldsymbol{u}(t) - \eta^* \nabla \boldsymbol{u}(t))$
    $t_f \leftarrow \psi(t_f - \eta \nabla t_f)$
    % Forward integration of the system function
    $\boldsymbol{x}(t) \leftarrow \boldsymbol{x}_0 + \int_0^{t_f} \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\mathrm{d}t$
    if HasConverged() **then**
      break gradient iteration
    end if
  end for
  % Update Lagrange multipiler (gradient ascent)
  $\tilde{\boldsymbol{g}}^{\text{last}} \leftarrow \tilde{\boldsymbol{g}}, \quad \tilde{\boldsymbol{g}}_T^{\text{last}} \leftarrow \tilde{\boldsymbol{g}}_T$
  $\tilde{\boldsymbol{g}}^T \leftarrow [\ \boldsymbol{g}^T(\boldsymbol{x}(t), \boldsymbol{u}(t), t), \quad \mathbf{max}\{\boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t), t), -\Upsilon_{\boldsymbol{h}}^{-1} \boldsymbol{\mu}_{\boldsymbol{h}}\}^T\ ]^T$
  $\tilde{\boldsymbol{g}}_T^T \leftarrow [\ \boldsymbol{g}_T^T(\boldsymbol{x}(t_f), t_f), \quad \mathbf{max}\{\boldsymbol{h}_T(\boldsymbol{x}(t_f), t_f), -\Upsilon_{\boldsymbol{h}_T}^{-1} \boldsymbol{\mu}_{\boldsymbol{h}_T}\}^T\ ]^T$
  $\boldsymbol{\mu}(t) \leftarrow \boldsymbol{\mu}(t) + \Upsilon \tilde{\boldsymbol{g}}$
  $\boldsymbol{\mu}_T(t) \leftarrow \boldsymbol{\mu}_T(t) + \Upsilon_T \tilde{\boldsymbol{g}}_T$
  % Update of penalty weights
  $\boldsymbol{\upsilon} \leftarrow \zeta(\boldsymbol{\upsilon}, \tilde{\boldsymbol{g}}, \tilde{\boldsymbol{g}}^{\text{last}})$
  $\boldsymbol{\upsilon}_T \leftarrow \zeta(\boldsymbol{\upsilon}, \tilde{\boldsymbol{g}}_T, \tilde{\boldsymbol{g}}_T^{\text{last}})$
  if HasConverged() **then**
    break multiplier iteration
  end if
end for
return $\mathcal{T}^* : [\boldsymbol{x}^*(t)^T, \boldsymbol{u}^*(t)^T]^T,\ t \in [0, t_f^*]$

Appendix B

Additional Parameters

In this appendix chapter, the full set of parameters used for the channel estimation from Chapter 5.1 is provided. For this, Table B.1 shows the parameter of the simulation study from Chapter 5.1.3, while Table B.2 contains the parameter for the real-world experiments of Chapter 5.1.4. Furthermore, a C++ implementation of the analytical solution for time-weighted jerk optimal trajectories from Chapter 6.2.4 is provided.

Symbol	Value(s)	Description
$\boldsymbol{A}$	$\begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$	statistical prior
l_w	100	window length
$\boldsymbol{\theta}_{DC}$	$\begin{bmatrix} 0.2 \\ 0.3 \end{bmatrix}$	consistency threshold for DC
p_d	0.999	discount probability for ω_i
$\tilde{p}_d$	1 - $\frac{2\,N_{A,i}}{N_{A,i}+4}$	discount probability for $\tilde{\omega}_i$, $N_{A,i}$: transitions from this state
t_1	19081	time / packet number of first step
t_2	30851	time / packet number of second step
t_3	54973	time / packet number of third step

Table B.1: Parameters for the simulation of the SL-based channel estimation method (see Section 5.1.3).

Symbol	Value(s)	Description
$\boldsymbol{A}$	$\begin{bmatrix} 0.9 & 0.1 & 0.0 \\ 0.9 & 0.0 & 0.1 \\ 1.0 & 0.0 & 0.0 \end{bmatrix}$	statistical prior
l_w	100	window length
p_d	0.999	discount probability for ω_i
$\tilde{p}_d$	1.0	discount probability for $\tilde{\omega}_i$
$\boldsymbol{\theta}_{DC}$	$\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$	consistency threshold for DC
$\theta_{\Delta t_1}$	3.5 ms	first threshold
$\theta_{\Delta t_2}$	10.5 ms	second threshold

Table B.2: Parameters for the experimental SL-based channel estimation (see Section 5.1.4).

```
void TimeWeightedTrajectory(const Eigen::VectorXd& x_ego,
                            const Eigen::VectorXd& xf,
                            Eigen::MatrixXd& trajectory,
                            double tf,
                            double w_t,
                            double dt)

    Eigen::VectorXd xf_copy = xf;
    if (std::abs(xf(1) - v_max) < 1e-1) xf_copy(1) -= 1e-1;

    double t2 = w_t * w_t;
    double t3 = tf * tf;
    double t4 = 180.0 * t2 * w_t;
    double t21 = 180.0 * w_t;
    double t5 = t4 - t21;
    double t6 = t5 * tf;
    double t7 = t3 * t3;
    double t8 = 9.0 * w_t;
    double t9 = t8 - 9.0;
    double t10 = t7 * t9;
    double t11 = 36.0 * t2;
    double t12 = t11 - 36.0;
    double t13 = t3 * t12 * tf;
    double t14 = t2 * 162.0;
    double t15 = 30.0 * w_t * t2;
    double t16 = t14 + t15 - 162.0 * w_t - 30.0;
    double t17 = t3 * t16;
    double t18 = 180.0 * t2;
    double t19 = 360.0 * x_ego[0];
    double t20 = 360.0 * w_t * xf_copy[0];
```

```
        double t22 = t4 + t6 + t10 + t13 + t17 - t18;
        double t23 = 360.0 * w_t * x_ego[0];

        double b_param =
        -(-360.0 * xf_copy[0] + t19 + t20 - t23 +
        t7 * (x_ego[2] * 9.0 - xf_copy[2] * 3.0 - x_ego[2] * w_t * 9.0 +
            xf_copy[2] * w_t * 3.0) -
        tf * (xf_copy[0] * 360.0 - t19 - t20 + t23 - x_ego[1] * 180.0 -
            xf_copy[1] * 180.0 + x_ego[1] * w_t * 180.0 +
        xf_copy[1] * w_t * 180.0) +
        t3 * (x_ego[2] * 30.0 - xf_copy[2] * 30.0 + x_ego[0] * 60.0 -
            xf_copy[0] * 60.0 + x_ego[1] * 192.0 +
        xf_copy[1] * 168.0 - x_ego[2] * w_t * 30.0 + xf_copy[2] * w_t *
            30.0 - x_ego[0] * w_t * 60.0 +
        xf_copy[0] * w_t * 60.0 - x_ego[1] * w_t * 192.0 - xf_copy[1] * w_t
             * 168.0) +
        t3 * tf *
        (36.0 * x_ego[2] - xf_copy[2] * 24.0 + x_ego[1] * 36.0 + xf_copy[1]
             * 24.0 - 36.0 * x_ego[2] * w_t +
        xf_copy[2] * w_t * 24 - x_ego[1] * w_t * 36.0 - xf_copy[1] * w_t *
            24.0)) /
        (21.0 * t7 * (w_t - 1) + t7 * tf + tf * (t18 - t21) - t22 * log(tf
            + w_t) + t22 * log(w_t) -
        t3 * (90.0 * w_t - t18 + 90.0) + t3 * tf * (30.0 * t2 + 72.0 * w_t
            - 102.0));
        t2 = log(w_t);
        t3 = tf * tf;
        t4 = w_t * w_t;
        t5 = tf + w_t;
        t6 = log(t5);
        t7 = 1.0 / (tf * tf * tf * tf * tf);
        t8 = 1.0 / (tf * tf * tf * tf);
        t9 = 1.0 / (tf * tf * tf);

        Eigen::VectorXd coeff(7);
        coeff[0] = -10.0 * t7 *
        (12.0 * (x_ego[0] - xf_copy[0]) + (x_ego[2] - xf_copy[2]) * t3 +
            6.0 * tf * (x_ego[1] + xf_copy[1])) +
        10.0 * b_param * t7 *
        (3.0 * t3 + t2 * t3 + 6.0 * t2 * t4 - t3 * t6 - t4 * t6 * 6.0 + tf
            * w_t * 6.0 + t2 * tf * w_t * 6.0 -
        t6 * tf * w_t * 6.0);
        coeff[1] = 3.0 * t8 *
        (60.0 * (x_ego[0] - xf_copy[0]) + 6.0 * x_ego[2] * t3 - 4.0 *
            xf_copy[2] * t3 + 32.0 * tf * x_ego[1] +
        28.0 * tf * xf_copy[1]) -
        3.0 * b_param * t8 *
        (17.0 * t3 + 6.0 * t2 * t3 + 30.0 * t2 * t4 - 6.0 * t3 * t6 - 30.0
            * t4 * t6 + 30.0 * tf * w_t +
        32.0 * t2 * tf * w_t - 32.0 * t6 * tf * w_t);
        coeff[2] = -3.0 * t9 *
        (20.0 * (x_ego[0] - xf_copy[0]) + 3.0 * x_ego[2] * t3 - xf_copy[2]
            * t3 + 12.0 * tf * x_ego[1] +
        8.0 * tf * xf_copy[1]) +
```

```
        3.0 * b_param * t9 *
        (7.0 * t3 + 3.0 * t2 * t3 + 10.0 * t2 * t4 - 3.0 * t3 * t6 - 10.0 *
             t4 * t6 + 10.0 * tf * w_t +
        12.0 * t2 * tf * w_t - 12.0 * t6 * tf * w_t);
        coeff[3] = x_ego[2] - b_param * t2;
        coeff[4] = x_ego[1] - b_param * t2 * w_t;
        coeff[5] = x_ego[0] - 0.5 * b_param * t2 * t4;
        coeff[6] = b_param;

        double t_traj = 0.0;
        int traj_length = (int)(tf / dt) + 2;
        trajectory.resize(5, traj_length);
        for (int i = 0; i < traj_length; i++)
        {
                trajectory(0, i) = t_traj;
                t_traj = t_traj + dt;
                if (t_traj > tf)
                t_traj = tf;
        }
        for (int i = traj_length - 1; i >= 0; i--)
        {
                t_traj = trajectory(0, i);
                t2 = t_traj * t_traj;
                t3 = t2 * t2;
                t4 = t_traj + w;
                t5 = log(t4);
                t6 = b_param * t5 * w;
                t7 = b_param * t5;

                trajectory(1, i) = coeff[5] + (1.0 / 12) * coeff[1] * t3 +
                    t_traj * (coeff[4] + t6 - 0.5 * b_param * w) +
                t2 * (0.5 * coeff[3] - 0.75 * b_param + 0.5 * b_param * t5)
                     + 0.5 * b_param * t5 * w * w +
                0.05 * coeff[0] * t_traj * t3 + (1.0 / 6.0) * coeff[2] *
                    t_traj * t2;
                trajectory(2, i) = coeff[4] + t6 + 0.25 * coeff[0] * t3 +
                    0.5 * coeff[2] * t2 +
                t_traj * (coeff[3] - b_param + t7) + (1.0 / 3.0) * coeff[1]
                     * t_traj * t2;
                trajectory(3, i) = coeff[3] + t7 + coeff[2] * t_traj +
                    coeff[1] * t2 + coeff[0] * t_traj * t2;
                trajectory(4, i) = coeff[2] + 2.0 * coeff[1] * t_traj + 3.0
                     * coeff[0] * t2 + b_param / t4;
        }
}
```

Listing B.1: C++ Implementation of Time-Wighted Jerk Optimal Trajectory

Appendix C

Supervised Bachelor and Master Theses

Aguilar, Julian: *Lernen von Einfädelmanövern für die Handlungsplanung automatisierter Fahrzeuge*. Bachelor thesis, Ulm University, 2019.

Dehler, Robin: *Value Function Learning zum Lernen von Einfädelmanövern für die Handlungsplanung automatisierter Fahrzeuge*. Bachelor thesis, Ulm University, 2019.

Düster, Lukas: *Resiliente Formationsregelung unter eingeschränkter Beobachtbarkeit und Steuerbarkeit*. Master thesis, Ulm University, 2020.

Geisler, Paul: *Anwendung der Subjective Logic in der Selbstbewertung des Single Object Trackings in Clutter*. Master thesis, Ulm University, 2021.

Holzbock, Adrian: *Deep Software-in-the-Loop Simulator für hochautomatisiertes Fahren*. Master thesis, Ulm University, 2020.

Huo, Lu: *Auto calibration of a Multi Sensor Network*. Master thesis, Ulm University, 2017.

Joos, David: *Modellbildung von Lichtsignalanlagen für die Bewegungsplanung automatisierter Fahrzeuge*. Bachelor thesis, Ulm University, 2021.

Oesterlen, Alexander: *Bewegungsplanung für automatisierte Fahrzeuge zum Einfädeln in fließenden Verkehr*. Master thesis, Ulm University, 2018.

Renftle, Eva Mareike: *Identifikation von Hidden Markov Models mit Subjective Logic*. Bachelor thesis, Ulm University, 2020.

Strohm, Mona: *Entwicklung und Evaluation optimaler Trajektorienplanungsverfahren für vernetzte automatisierte Fahrzeuge innerhalb sicherer Zustandsmengen*. Master thesis, Ulm University, 2021.

Tsaregorodtsev, Alexander: *Risk Assessment of Merging Scenarios for Highly Automated Vehicles*. Master thesis, Ulm University, 2019.

Appendix D

Author's Publications

Parts of this thesis were pre-published in author's publications.

Müller, J. and Buchholz, M.: *A Risk and Comfort Optimizing Motion Planning Scheme for Merging Scenarios*. In: *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, pages 3155–3161, Auckland, New Zealand, 2019.

Müller, J. and Buchholz, M.: *Subjective Logic Reasoning: An Urn Model Intuition and Application to Connected Automated Driving*. In: *at - Automatisierungstechnik*, volume 69, no. 2, pages 111–121, 2021.

Müller, J.; Gabb, M.; and Buchholz, M.: *A Subjective-Logic-based Reliability Estimation Mechanism for Cooperative Information with Application to IV's Safety*. In: *Proc. IEEE Intell. Vehicles Symp. (IV)*, pages 1940–1946, Paris, France, France, 2019.

Müller, J.; Griebel, T.; Gabb, M.; and Buchholz, M.: *Subjective Logic-Based Identification of Markov Chains and Its Application to CAV's Safety*. In: *Proc. IEEE Connected and Automated Vehicles Symp. (CAVS)*, pages 1–5, Honolulu, HI, USA, USA, 2019.

Müller, J.; Herrmann, M.; Strohbeck, J.; and Buchholz, M.: *Motion Planning for Connected Automated Vehicles at Occluded Intersections With Infrastructure Sensors*. arXiv: https://arxiv.org/abs/2110.11246. 2021.

Müller, J.; Meuser, T.; Steinmetz, R.; and Buchholz, M.: *A Trust Management and Misbehaviour Detection Mechanism for Multi-Agent Systems and its Application to Intelligent Transportation Systems*. In: *Proc. IEEE Int. Conf. Control Automation (ICCA)*, pages 325–331, Edinburgh,UK, 2019.

Strohbeck, J.; Müller, J.; Holzbock, A.; and Buchholz, M.: *DeepSIL: A Software-in-the-Loop Framework for Evaluating Motion Planning Schemes Using Multiple Trajectory Prediction Networks*. IEEE/RSJ Int. Conf. on Intell. Robots Systems (IROS). Prague, Czech Republic, 2021.

Furthermore, some parts of the chapter *Background* will be published in the following article:

Buchholz, M.; Müller, J.; Herrmann, M.; et al.: *Handling Occlusions in Automated Driving Using a MEC Server-based Environment Model From Infrastructure Sensors*. accepted for publication in Intell. Transp. Syst. Mag. 2021.

The author shares the following articles with Thomas Griebel and Jan Strohbeck, respectively, as *first author with equal contribution*:

Müller, J.; Griebel, T.; Gabb, M.; and Buchholz, M.: *Subjective Logic-Based Identification of Markov Chains and Its Application to CAV's Safety*. In: *Proc. IEEE Connected and Automated Vehicles Symp. (CAVS)*, pages 1–5, Honolulu, HI, USA, USA, 2019.

Strohbeck, J.; Müller, J.; Holzbock, A.; and Buchholz, M.: *DeepSIL: A Software-in-the-Loop Framework for Evaluating Motion Planning Schemes Using Multiple Trajectory Prediction Networks*. IEEE/RSJ Int. Conf. on Intell. Robots Systems (IROS). Prague, Czech Republic, 2021.

Publication List

The author has contributed to the following publications.

Buchholz, M.; Herrmann, M.; Müller, J.; et al.: *A Digital Mirror: A Mobile Edge Computing Service Based on Infrastructure Sensors*. In: *25th ITS World Congr.* Copenhagen, Denmark, 2018.

Buchholz, M.; Müller, J.; Herrmann, M.; et al.: *Handling Occlusions in Automated Driving Using a MEC Server-based Environment Model From Infrastructure Sensors*. accepted for publication in Intell. Transp. Syst. Mag. 2021.

Griebel, T.; Müller, J.; Buchholz, M.; and Dietmayer, K.: *Kalman Filter Meets Subjective Logic: A Self-Assessing Kalman Filter Using Subjective Logic*. In: *Proc. IEEE Int. Conf. Inf. Fusion (FUSION)*, pages 1–8, Rustenburg, South Africa, 2020.

Gabb, M.; Maier, M.; Völz, B.; et al.: *Urban Automated Driving using Environment Perception as a Service*. In: *25th ITS World Congr.* Copenhagen, Denmark, 2018.

Herrmann, M.; Müller, J.; Strohbeck, J.; and Buchholz, M.: *Environment Modeling Based on Generic Infrastructure Sensor Interfaces Using a Centralized Labeled-Multi-Bernoulli Filter*. In: *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, pages 2414–2420, Auckland, New Zealand, 2019.

Herrmann, M.; Piroli, A.; Strohbeck, J.; Müller, J.; and Buchholz, M.: *LMB Filter Based Tracking Allowing for Multiple Hypotheses in Object Reference Point Association*. In: *Proc. Int. Conf. on Multisensor Fusion and Integration for Intel. Sys. (MFI)*, pages 197–203, Karlsruhe, Germany, 2020.

Müller, J. and Buchholz, M.: *A Risk and Comfort Optimizing Motion Planning Scheme for Merging Scenarios*. In: *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, pages 3155–3161, Auckland, New Zealand, 2019.

Müller, J. and Buchholz, M.: *Subjective Logic Reasoning: An Urn Model Intuition and Application to Connected Automated Driving*. In: *at - Automatisierungstechnik*, volume 69, no. 2, pages 111–121, 2021.

Müller, J.; Gabb, M.; and Buchholz, M.: *A Subjective-Logic-based Reliability Estimation Mechanism for Cooperative Information with Application to IV's Safety*. In: *Proc. IEEE Intell. Vehicles Symp. (IV)*, pages 1940–1946, Paris, France, France, 2019.

Müller, J.; Griebel, T.; Gabb, M.; and Buchholz, M.: *Subjective Logic-Based Identification of Markov Chains and Its Application to CAV's Safety*. In: *Proc. IEEE Connected and Automated Vehicles Symp. (CAVS)*, pages 1–5, Honolulu, HI, USA, USA, 2019.

Müller, J.; Herrmann, M.; Strohbeck, J.; Belagiannis, V.; and Buchholz, M.: *LACI: Low-effort Automatic Calibration of Infrastructure Sensors*. In: *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, pages 3928–3933, Auckland, New Zealand, 2019.

Müller, J.; Herrmann, M.; Strohbeck, J.; and Buchholz, M.: *Motion Planning for Connected Automated Vehicles at Occluded Intersections With Infrastructure Sensors*. arXiv: https://arxiv.org/abs/2110.11246. 2021.

Müller, J.; Krebs, S.; Buchholz, M.; and Hohmann, S.: *Interval MRAC with Proven Stability Properties and Application on Induction Motor Control*. In: *Proc. IEEE Int. Conf. Control Automation (ICCA)*, pages 173–179, 2018.

Müller, J.; Meuser, T.; Steinmetz, R.; and Buchholz, M.: *A Trust Management and Misbehaviour Detection Mechanism for Multi-Agent Systems and its Application to Intelligent Transportation Systems*. In: *Proc. IEEE Int. Conf. Control Automation (ICCA)*, pages 325–331, Edinburgh,UK, 2019.

Strohbeck, J.; Belagiannis, V.; Müller, J.; et al.: *Multiple Trajectory Prediction with Deep Temporal and Spatial Convolutional Neural Networks*. In: *Proc. IEEE/RSJ Int.Conf. Intell. Robots Syst. (IROS)*, pages 1992–1998, Virtual, 2020.

Strohbeck, J.; Müller, J.; Holzbock, A.; and Buchholz, M.: *DeepSIL: A Software-in-the-Loop Framework for Evaluating Motion Planning Schemes Using Multiple Trajectory Prediction Networks*. IEEE/RSJ Int. Conf. on Intell. Robots Systems (IROS). Prague, Czech Republic, 2021.

Wolf, M.; Willecke, A.; Müller, J.; et al.: *Securing CACC: Strategies for Mitigating Data Injection Attacks*. In: *IEEE Veh. Netw. Conf. (VNC)*, pages 1–7, New York, NY, USA, 2020.

Acronyms

ABF	aleatory average belief fusion 24, 39, 49, 69, 80, 81
ACADO	Automatic Control and Dynamic Optimization 28
ARQ	Automated Response Request 57
BWP	Boundary Value Problem 29
C-ITS	cooperative intelligent transportation systems 79
C-V2X	cellular vehicle to anything 77
CAM	cooperative awareness message 64, 77
CAV	connected automated vehicle . . 1–5, 7, 9–11, 13–16, 18, 19, 33, 53, 54, 58, 62–64, 66, 67, 70–76, 79, 88–101, 104, 105, 107, 108, 110, 111, 113–120
CBF	aleatory cummulative belief fusion 24, 36, 38, 45, 46, 51, 56, 67, 68, 80, 120
CPM	cooperative perception message 77
CV	constant velocity 15, 70
DC	Degree of Conflict 23, 40, 50, 51, 56, 65, 81, 82, 127, 128
DENM	decentralized environmental notification message 77, 86
E2E	end-to-end 7–9
EOS	end of sight 99, 100
FDIR	fault detection, isolation, and recovery 4, 6
FOV	field of view 9, 16, 66, 68, 70, 73, 75, 96
GPS	Global Positioning System 15, 58
GRAMPC	Gradient Based Nonlinear Model Predictive Control 29, 105, 118, 119, 121, 124
HARQ	Hybrid Automated Response Request 57, 58, 60
IoT	Internet of Things 4, 5

ISS invariably safe set . . . . . xi, 3, 9, 30, 31, 92–94, 98, 99, 119
ITSS infrastructure-based traffic surveillance system 13–15, 17, 18, 53, 62, 66, 70, 72, 73, 75, 76, 78, 81, 83–85, 120

LIDAR light detection and ranging 15–18, 70, 71
LMB Labeled Multi-Bernoulli 15, 16, 18
LOS line of sight 57
LTE Long Term Evolution 57, 59, 61

MAB embedded real-time computing system for rapid prototyping 19
MAP maximum a posteriori 34
MAS multi-agent systems 4, 6, 10
MEC multi access edge computing 3, 14–16, 53, 58, 62, 71, 96, 120
ML maximum likelihood 34
MMSE minimum mean square error 41, 44
MOT multi-object tracking 18
MPC Model Predictive Control 3, 25, 28, 89, 94, 96, 101, 104, 118, 119

NIS normalized innovation squared 4, 6
NLOS non-line of sight 57

OCP Optimal Control Problem 9, 10, 26, 28, 29, 90, 95, 101–105, 118, 119, 122–124

PDF probability density function 71–74, 80
PGA Point of Guaranteed Arrival 30, 31, 94, 95, 99, 104
PKI Public Key Infrastructure 77, 78
PNR Point of No Return 30, 31, 94, 95, 99, 104, 108
POMDP partially observable Markov decision process 3, 7–9, 90

ROI region of interest 70, 71
RRT rapidly exploring random tree 8
RTK real time kinematic 18, 113, 115

SL Subjective Logic 2, 4, 6, 7, 11, 21–23, 29, 33, 35–38, 40, 41, 43, 46, 51–55, 59, 61, 62, 68, 73, 77, 80, 81, 83, 88, 120, 127, 128
SNR Signal-to-Noise ratio 57
SOTIF safety of the intended functionality 3–5
SPU sensor processing unit 14, 15

T2T track-to-track 3, 10, 19, 89, 96, 97

V-MAS vehicular multi-agent system 2, 4, 5, 11, 62, 76–79, 85, 88, 119, 120
V2X vehicle-to-anything . . . . . xi, 1, 3, 18, 53, 88, 118, 119

WABF weighted average belief fusion 24, 120

List of Symbols

General notation

$\square_1$	first element of a vector; 2 would indicate the second one, and so on
$\lvert\square\rvert$	absolute value
$\check{\square}$	modified quantity in augmented Lagrangian approach
$\Delta\square$	increment of the quantity
$\square\Big\rvert_{a=a_1}$	term or function evaluated at $a = a_1$
$\frac{\delta\square}{\delta\square}$	partial derivative
$\dot{\square}$	time derivative
$\square^*$	optimal quantity
$\square_f$	final
$\square(\cdot)$	function of the respective quantity, where the arguments are omitted
$\boldsymbol{A}$	multi-dimensional quantity
$\hat{\square}$	estimated quantity
$\square_i$, $\square_j$	integer indices
$\square_k$	time step k
$\square_{\max}$	maximum value of the quantity
$\square_{\min}$	minimum value of the quantity
$\square_{\text{ref}}$	reference quantity
$\{\square_i\}_{i=1,\dots,N}$	stacking the entries of a quantity into a set
$\square^T$	transposed quantity
$\square_T$	terminal quantity
$\tilde{\tilde{\square}}$	double modified quantity
$\tilde{\square}$	modified quantity
$[\square_i]_{i=1,\dots,N}$	stacking the entries of a quantity into a vector

Quantities, and Functions related to Control Theory

A_i, A_j	agents in a multi-agent system

$\alpha_1, \ldots, \alpha_6$	coefficients of the optimal longitudinal trajectory
γ	dependent auxiliary coefficients of the optimal longitudinal trajectory
$\mathcal{B}$	behavior option
$c_1,\ c_2,\ c_3$	integration constants related to the optimal longitudinal trajectory
$c_{\min}$	minimal cost
$\mathcal{C},\ \mathcal{C}_{[s_j, s_{j+1}]}$	situation context for the road segment indicated by the interval $[s_j, s_{j+1}]$ along the route
c_{tmp}	temporary cost of the trajectory segments aggregated so far
$\Delta t_{f,i}$	time difference the CAV needs to travel from the target state $\boldsymbol{x}_{f,i-1}$ to the target state $\boldsymbol{x}_{f,i}$
$d_{FR}, d_{FL}, d_{RL}, d_{RR}$	lateral distance of the respective corner points to the corresponding lane boundary
d_{Lane}	minimal distance to lane boundary
$d_{\perp,\text{ref}}$	offset to the lateral reference position
$\eta,\ \eta^*$	step size for the gradient iteration (result of the line search)
$\boldsymbol{f}(\cdot)$	state dynamics function
$\boldsymbol{f}_{\text{NLVM}}(\cdot)$	system dynamics of the nonlinear vehicle model
$\boldsymbol{g}(\cdot)$	equality constraints
$\boldsymbol{g}_T(\cdot)$	terminal equality constraints
$\tilde{\boldsymbol{g}}^{\text{last}},\ \tilde{\boldsymbol{g}}_T^{\text{last}}$	last value of the equality constraints
$H(\cdot)$	Hamiltonian
$\boldsymbol{h}(\cdot)$	inequality constraints
$\boldsymbol{h}_T(\cdot)$	terminal inequality constraints
I	importance of a behavior option
i_{grad}	current gradient iteration
$i_{\text{grad,max}}$	maximum gradient iterations
i_{mult}	current multiplier iteration
$i_{\text{mult,max}}$	maximum multiplier iterations
$J(\cdot)$	cost function
J_{long}	cost function for the longitudinal trajectory
$l(\cdot)$	integral cost function
$\boldsymbol{\lambda}$	costates
$\mathcal{L}_{\mathcal{B}}$	list of behavior options
$\mathcal{L}_{\text{ego}}$	list of objects in the ego environment model
$\mathcal{L}_{\text{ext}}$	list of objects in the external environment model
$\mathcal{L}_{\text{rel}}$	list of relevant objects

$\mathcal{L}_{\text{virt}}$	list of virtual objects
$\mathcal{M}_{\text{local}}$	local map
$\boldsymbol{\mu}_{\boldsymbol{h}}, \boldsymbol{\mu}_{\boldsymbol{h}_T}$	Lagrange multipliers for inequality constraints
$\boldsymbol{\mu}, \boldsymbol{\mu}_T$	Lagrange multipliers
o_j	dynamic object
$\mathcal{P}$	reference path
$\boldsymbol{\Phi}_{\text{safe}}$	safe feedback control law
$\mathcal{P}_{\text{LB}}$	path corresponding to the left bound of the street
$\mathcal{P}_{\text{RB}}$	path corresponding to the right bound of the street
$\mathcal{P}$	reference path
$\mathcal{R}$	route of the CAV
$\mathcal{S}_C$	set of constraints
$\mathcal{S}_{\text{correct}}$	set of correctly predicted states
$\mathcal{S}_{\text{incorrect}}$	set of incorrectly predicted states
s_{ego}	ego position
s_f, v_f	final position and velocity
$\mathcal{S}_{\text{ISS}}^{t}$	invariably safe set at a given time t
$\mathcal{S}_{\text{ISS}}^{\text{fin},\tilde{t}_f}$	final invariably safe set at a final time $\tilde{t}_f$
$\hat{\mathcal{S}}_{\text{ISS}}^{t}$	estimated invariably safe set at a given time t
$\mathcal{S}_{\text{ISS}}^{\text{init},t}$	initial invariably safe set at a given time t
s^{-}	safety distance to an object behind
s_{PGA}	position of the point of guaranteed arrival
$s^{+}(t)$	safety distance to an object ahead
s_{PNR}	position of the point of no return
s_{pred}	predicted position
$\mathcal{S}_{\text{RA}}$	set of regularizing assumptions
$\mathcal{S}_{\text{reach}}^{t}$	reachable set at a given time t
$\mathcal{S}_{\text{safe}}^{t}$	safe set: maximum set of collision-free states at a given time t
$\hat{\mathcal{S}}_{\text{safe}}^{\text{stat}}$	estimated set of static safe states
$s_{x,\text{ref}}$	reference x coordinate in a Cartesian frame
$\mathcal{S}_{\boldsymbol{x}_0}$	set of initial states
$s_{y,\text{ref}}$	reference y coordinate in a Cartesian frame
s_{yield}	position of the yield line
$\mathcal{T}$	trajectory
t	time

t_0	initial time
τ	alternative time variable
t_f	final time
$t_{f,\max}$	maximum final time
$t_{f,\min}$	minimum final time
$t_{f,\perp}$	final time for the lateral trajectory tracking
t_f^*	optimal final time
$\mathcal{T}_{\text{long}}$	longitudinal trajectory
$\mathcal{T}_{\text{long}}^*$	optimal longitudinal trajectory
$\tilde{\mathcal{T}}_{\text{long}}^{*,N-1,N}$	Nth segment of the optimal longitudinal trajectory
t_{PGA}	time at which the point of guaranteed arrival is reached
t_{PNR}	time at which the point of no return is reached
T_{pred}	prediction horizon
$\mathcal{T}_{\text{ref}}$	reference trajectory
$\mathcal{T}^*$	optimal trajectory
$\boldsymbol{u}$	control variables vector (input)
$\boldsymbol{u}_{\max}$	maximum input
$\boldsymbol{u}_{\min}$	minimum input
$\boldsymbol{\Upsilon}, \boldsymbol{\Upsilon}_T$	penalty weight matrices
$\boldsymbol{\upsilon}, \boldsymbol{\upsilon}_T$	penalty weights
$\boldsymbol{u}^*$	optimal control variable vector
$V(\cdot)$	terminal cost function
v_{PGA}	velocity at the point of guaranteed arrival
v_{PNR}	velocity at the point of no return
v_{pred}	predicted velocity
v_{sl}	speed limit velocity
w_1, w_2, w_3	optimization weight for curvature MPC
w_a	acceleration weight for the lateral trajectory tracking
$w_\perp$	lateral distance weight for the lateral trajectory tracking
w_φ	weight of the course angle for the lateral trajectory tracking
w_{risk}	weight of the residual risk
w_t	time weight
w_{t_f}	weight that balances the priority of time optimality versus time-weighted jerk optimality
w_{u_1}	steering wheel actuation weight for the lateral trajectory tracking

w_v	velocity weight for the lateral trajectory tracking
$\boldsymbol{x}$	state vector
$\boldsymbol{x}_0$	initial state vector
$\boldsymbol{x}_{\text{ego}}$	ego state vector
$\boldsymbol{x}_{\text{ego,i}}, d_{\text{min},i}$	state vector of the ith object in the ego perception and minimum assotiation distance to an object from the external perception
$\boldsymbol{x}_f$	final state vector
$\boldsymbol{x}_{FR}, \boldsymbol{x}_{FL}, \boldsymbol{x}_{RL}, \boldsymbol{x}_{RR}$	front right, front left, rear left, and rear right corner points of the CAV
$\boldsymbol{x}_f^*$	optimal final state vector
$\langle(\boldsymbol{x}_f, t_f)\rangle_i$	target sequence
$\boldsymbol{\xi}$	slack variables
$\boldsymbol{\xi}_T$	slack variables for terminal inequality constraints
$\boldsymbol{x}_{\text{meas}}$	measured states
$\boldsymbol{x}_{\text{pred}}$	predicted states
$\boldsymbol{x}^*$	optimal state vector
$\mathcal{X}(\cdot)$	solution of system dynamics initial value problem
$\boldsymbol{\zeta}(\cdot)$, $\boldsymbol{\zeta}_T(\cdot)$	penalty weight update functions
$\boldsymbol{\psi}(\cdot)$	component-wise projection function
z_i	ith measurement

Physical Quantities

a	acceleration
$a_\perp$	lateral acceleration
$a_{\perp,\text{max}}$	maximum lateral acceleration
a_{max}	maximum acceleration
a_{min}	minimum acceleration or maximum deceleration
Δt	time delay
κ_{ref}	reference curvature
κ	curvature
l_{veh}	vehicle length
φ	course angle
s	1D position, e.g. along a path or lane
t_{MA}	moving average of time delay
v	velocity
v_{char}	characteristic velocity of the vehicle

$v_{\max}$	maximum velocity

Probabilistic Quantities, Functions, and Operators

η	lower bound of error probability
$\boldsymbol{A}$	statistical prior matrix
$\boldsymbol{a}$	base rate
$\boldsymbol{a}_A$	base rate of opinion ω_A
$\boldsymbol{a}_{A\oplus B}$	base rate of opinion $\omega_{A\oplus B}$
$\boldsymbol{a}_B$	base rate of opinion ω_B
A, B, C	information sources, e.g. samples drawn from first urn
$AC(\mathcal{S}_{\omega,A})$	average conflict in set $\mathcal{S}_{\omega,A}$
$\boldsymbol{a}_C$	base rate of opinion ω_C
α	scaling factor, a coefficient
$\boldsymbol{a}_{\text{td}}$	trust discounted base rate
$\boldsymbol{b}$	belief mass
$\boldsymbol{b}_A$	belief mass of opinion ω_A
•, •, ∘	red, black and white balls which are drawn from the urns in the urn model. Furthermore, checkered balls ◒ are used to model ignorance.
$\boldsymbol{b}_{A\oplus B}$	belief mass of opinion $\omega_{A\oplus B}$
$\boldsymbol{b}_B$	belief mass of opinion ω_B
$\boldsymbol{b}_C$	belief mass of opinion ω_C
$\beta(\cdot)$	beta distribution
$\boldsymbol{b}_{\text{td}}$	belief mass of trust discounted opinion
c_A^A, c_A^B	weight of ω_A^A and ω_A^B for weighted ABF operator
c_{cons}	weight of consistency tests in overall fusion
c_{plaus}	weight of plausibility tests in overall fusion
c_{under}	weight penalizing uncertainty under-estimation
$\mathcal{D}(\cdot)$	trust discounting
$DC(\cdot)$	degree of conflict
DC^{A_j,ref^*}	degree of conflict between jth agent and the reference opinion with the most support
$DC_{\text{map}}^{\text{online}}$	degree of conflict between the online gathered opinion and the reference opinion of the map test
$\text{Dir}(\cdot)$	Dirichlet distribution
d_M	Mahalanobis distance
$\mathbb{E}\{\cdot\}$	expectation
$\text{erf}(\cdot)$	error function corresponding to the Gaussian distribution

$\boldsymbol{P}_{\text{MMSE}}$	minimum mean square error probability
$f_{\mathcal{L}_{\text{rel}} \cup \mathcal{L}_{\text{virt}}}(s,t)$	probability distribution of the states of the list of relevant and the list of virtual objects
$\Gamma(\cdot)$	Gamma function
$\mathcal{L}_{\text{sub}}$, $\mathcal{S}_{\text{sub}}$	list of subsets S_{sub}, where each subset contains all balls of a specific color from $\mathcal{S}_A$
l_w	window length for statistics
$\{X_{t_k} \mid t_k > 0, m \in \mathbb{N}\}$	finite Markov chain
μ	mean value
$\hat{\mu}_{\text{ITSS}}$, $\hat{\sigma}_{\text{ITSS}}$	estimated mean and variance of the measurements from the ITSS
$\text{Mult}(\cdot)$	multinomial distribution
μ_{o_j}, σ_{o_j}	mean and variance corresponding to the object $o_j \in \mathcal{L}_{\text{rel}} \cup \mathcal{L}_{\text{virt}}$
$\mathcal{N}(\cdot)$	normal distribution
N_A	sample size, number of balls in $\mathcal{S}_A$
N_B	sample size, number of balls in $\mathcal{S}_B$
N_{DC}	number of draws to determine consistency between two opinions through the degree of conflict
N_{up}	number of upsampling steps in the urn model
o_{ij}	observed state transition
$\{o_m\}_{m \in \mathbb{N}}$	observations
$\omega_{A,\text{dep}}$	part of ω_A that is statistically dependent on ω_B
$\omega_{A,\text{ind}}$	part of ω_A that is statistically independent of ω_B
$\omega_{B,\text{dep}}$	part of ω_B that is statistically dependent on ω_A
$\omega_{B,\text{ind}}$	part of ω_B that is statistically independent of ω_A
ω_{conf}	confirmative opinion that increases the estimated reliability
ω_{cons}	merged opinion of the consistency tests
ω_{ho}	opinion of a hidden observer
ω_{input}	opinion used as input to the fusion with the last step. Thus a simple temporal filtering is realized.
$\omega_i^{t_k}$	opinion on state i at time step t_k
ω_{loc}	opinion of the ego localization test on the external environment model
ω_{map}	opinion of the map test on the external environment model
ω_{mis}	opinion of the ego perception test that the external environment model misses a detection
ω_{perc}	opinion of the perception test
ω_{plaus}	merged opinion of the plausibility tests

ω_{pred}	opinion of the prediction test on the external environment model
ω^*_{ref}	reference opinion
$\omega_{\text{ref},j}$	jth reference opinion
$\mathbf{\Omega}^{t_k}$	SL opinion vector corresponding to transition matrix at time t_k
ω_{total}	merged opinion of prediction test, map test, ego localization test, and ego perception test
ω_{TR}	opinion on agent that truly reported an incidence
ω_{under}	opinion that the covariances in the position are under-estimated
ω	Subjective Logic opinion
ω_{A_i}	SL opinion on agent A_i
ω_A	SL opinion of source A
$\omega_{A \oplus B}$	resulting opinion from merging A and B with the CBF operator
$\omega_{A \tilde{\oplus} B}$	resulting opinion from merging A and B with the weighted ABF operator
ω_B	SL opinion of source B
ω_C	SL opinion of source C
ω_{td}	resulting opinion after trust discounting
$\omega_{\emptyset}$, $\mathcal{S}_{\emptyset}$	vacuous opinion and corresponding empty set of drawn balls
$\oplus$	Aleatory Cummulative Belief Fusion operator
$\underline{\oplus}$	Average Belief Fusion operator
$\bigoplus_{\forall \omega \in \mathcal{S}_\omega} \omega$	Aleatory Cummulative Belief Fusion operator for multiple opinions
$\underline{\bigoplus}_{\forall \omega \in \mathcal{S}_\omega} \omega$	Average Belief Fusion operator for multiple opinions
$\hat{\oplus}$	Uncertainty-Weighted Average Belief Fusion operator
$\tilde{\oplus}$	Weighted Average Belief Fusion operator
$\boldsymbol{p}$	probability vector, free variable of Dir($\cdot$)
$\boldsymbol{p}_A$	probability vector to draw a specific color from $\mathcal{U}_1$ after observing $\mathcal{S}_A$
$\boldsymbol{p}_{\text{age,loc}}$	trust discounting due to aging and poor observation location
p_{aging}	trust discount probability modeling the aging of trust
$\boldsymbol{p}_{A_i,\text{trust}}$	reliability trust of the *data source*
$p_{BB}(t_k, t_{k+1})$	transition probability from bad to bad in a Gilbert-Elliot channel model
$P(\cdot \mid \cdot)$	conditional probability
p_d	discount probability
$PD^{A_j}_{A_i}$	probability distance between the opinions of agent A_i and A_j

$\hat{\boldsymbol{p}}_{\mathrm{DC}}$	estimated probability vector corresponding to the degree of conflict
p_{dm}	probability of detecting a misbehavior
p_{dynamic}	probability of dynamic merging according to Monte-Carlo simulation
$p_{\text{fail-safe}}$	probability of applying a fail-safe maneuver according to Monte-Carlo simulation
p_{fp}	probability of false positives
$p_{\text{gentle stop}}$	probability of stopping gently at the yield line according to Monte-Carlo simulation
$p_{GG}(t_k, t_{k+1})$	transition probability from good to good in a Gilbert-Elliot channel model
$p_{ij}(t_k, t_{k+1})$	state transition probability
p_{im}	probability that the respective object is important to the decision making
$\boldsymbol{p}_{\mathrm{indep}}$	probability that the new data point adds independent information
$\hat{\boldsymbol{p}}_{A,\mathrm{red,MAP}}$	MAP estimate of probability to draw a red ball from the first urn
$\hat{\boldsymbol{p}}_{A,\mathrm{red,ML}}$	ML estimate of probability to draw a red ball from the first urn
$\boldsymbol{P}_{A,\perp}$	projected probability vector corresponding to ω_A
p_{RA}	probability that the regularizing assumptions hold
$\boldsymbol{P}_{\mathrm{ref}}$	probability that a reference measurement is found in a respective histogram bin
p_{rel}	probability that the external perception is reliable
p_{risk}	residual risk (probability)
$\tilde{\tilde{p}}_{\mathrm{risk},j}$, $\tilde{p}_{\mathrm{risk},j}$	intermediate quantities to approximate the residual risk (probability)
$\boldsymbol{P}_{\mathrm{ITSS}}$	probability that a ITSS measurement is found in a respective histogram bin
p_{tp}	probability of true positives
$\boldsymbol{P}(t_k, t_{k+1})$	transition matrix of a finite Markov chain
$\boldsymbol{p}^*_{\mathcal{U}1}$	unknown probability vector to draw a specific color from $\mathcal{U}_1$
$\boldsymbol{p}_{\mathrm{up}}$	probability vector after upsampling
p_{wb}	probability of a successful wrong blame
p_{white}	probability of drawing a white ball from the 4th urn in the urn model
$\boldsymbol{r}$	statistical evidence
$\boldsymbol{r}_A$	statistical evidence corresponding to opinion ω_A
$\boldsymbol{r}_B$	statistical evidence corresponding to opinion ω_B
$\boldsymbol{r}_C$	statistical evidence corresponding to opinion ω_C
$RF(A_i)$	revision factor for agent A_i after misbehavior
$p_{\mathrm{risk,max}}$	maximum acceptable residual risk

$\boldsymbol{r}_W$	obtained number of balls of a specific colors when drawing W balls from the third urn
$\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$, $\mathcal{S}_4$	sets of balls in $\mathcal{U}_1, \dots, \mathcal{U}_4$
$\mathcal{S}_A$, $\mathcal{S}_B$	sets of balls drawn from the first urn
$\tilde{\mathcal{S}}_A^A$, $\tilde{\mathcal{S}}_A^B$	observations of the set $\mathcal{S}_A$ in the partially observable case
$\mathcal{S}_{A,\text{up}}^{A\cup B}$, $\boldsymbol{r}_{A,\text{up}}^{A\cup B}$	joint upscaled sets and corresponding evidence count
$\mathcal{S}_{3,\text{up}}^A$, $\mathcal{S}_{3,\text{up}}^B$	upscaled sets of balls in the third urn
$\mathcal{S}_3^A$, $\mathcal{S}_3^B$	sets of balls in the third urn for the respective samples $\mathcal{S}_A^A$, $\mathcal{S}_A^B$
$\mathcal{S}_{A,\text{up}}^A$, $\mathcal{S}_{A,\text{up}}^B$	upscaled sets corresponding to $\mathcal{S}_A^A$ and $\mathcal{S}_A^B$
$\mathcal{S}_3^{A\tilde{\oplus}B}$	sets of balls in third urn when applying the weighted ABF fusion operator
$\mathcal{S}_{A,\text{black}}$, $N_{A,\text{black}}$	set of black balls in $\mathcal{S}_A$ and the number of black balls therein
$\mathcal{S}_{\text{noise}}$	single ball drawn from 4th urn (noisy observation) represented as set
$\mathcal{S}_{DC}^A$, $\mathcal{S}_{DC}^B$	set of balls from which the degree of conflict is calculated
$\mathcal{S}_{A,\perp}$, $\mathcal{S}_{B,\perp}$	set of balls distributed according to the projected probabilities $\boldsymbol{P}_{A,\perp}$ and $\boldsymbol{P}_{B,\perp}$.
$\mathcal{S}_{A,\text{red}}$, $N_{A,\text{red}}$	set of red balls in $\mathcal{S}_A$ and the number of red balls therein
$\mathcal{S}_{A\tilde{\oplus}B}$, $\boldsymbol{r}_{A\tilde{\oplus}B}$	sets of balls resulting from applying the weighted ABF fusion operator and its corresponding evidence count
$\mathcal{S}_{\text{draw}}$	ball randomly drawn e.g. from $\mathcal{S}_1$, represented as set with one element
$\mathcal{S}_{\text{draw}}^A$, $\mathcal{S}_{\text{draw}}^B$	ball randomly drawn from $\mathcal{S}_{A,\perp}$, $\mathcal{S}_{B,\perp}$, respectively, represented as set with one element
$\mathcal{S}_{\text{ho}}$	set of opinions from hidden observers
$\mathcal{S}_\omega^{\text{ref},i}$, $\mathcal{S}_\omega^{\text{ref},j}$	set of reference opinions corresponding to the ith or jth opinion cluster
$\mathcal{S}_{\mathcal{G},\text{sub}}$	set of connected subgraphs from which the opinion clusters are derived
$\mathcal{S}_{PD}$	set of pair-wise probability distances that do not exceed a threshold
$\mathcal{S}_{\omega,A}$	set of opinions from the agents involved in a reported incidence
$\mathcal{S}_{\omega,\text{online}}$	set of map test opinions based on online gathered data
$\mathcal{S}_{\omega,\text{ref}}$	set of reference opinions
$\mathcal{S}_{A,D}$, $N_{A,D}$	trust dicounted set of balls and the number of balls in it
$\tilde{\mathcal{S}}_A$, $\tilde{\omega}_A$	observed sets of balls drawn from the first urn and corresponding opinion
σ	standard deviation
σ_a	standard deviation of the acceleration noise

σ_v	standard deviation of the velocity noise
s_{ij}	state transition histogram
θ_{DC}	threshold for degree of conflict
$\theta_{DC}^{\text{clas}}$	classification threshold for degree of conflict
$\theta_{\text{DC}}^{\text{map}}$	map test threshold for degree of conflict
$\theta_{\text{DC}}^{\text{ref}}$	threshold for degree of conflict between an agent's opinion and a reference opinion
$\theta_{\Delta t,1}$	first decision border for channel identification
$\theta_{\Delta t,2}$	second decision border for channel identification
θ_{EPT}	threshold for the ego perception test
$\theta_{\text{M,loc}}$	threshold for the Mahalanobis distance used for the ego localization test
$\theta_{\text{M,pred}}$	threshold for the Mahalanobis distance used for the prediction test
θ_{PD}	threshold for probability distance
u	statistical uncertainty
u_A	statistical uncertainty of opinion ω_A
u_B	statistical uncertainty of opinion ω_B, statistical uncertainty for *bad* state in Gilbert-Elliot model
$u_{A\oplus B}$	statistical uncertainty of opinion $\omega_{A\oplus B}$
u_C	statistical uncertainty of opinion ω_C
u_G	statistical uncertainty for *good* state in Gilbert-Elliot model
$\mathcal{U}_1$, $\mathcal{U}_2$, $\mathcal{U}_3$, $\mathcal{U}_4$	the four urns in the urn model
u_{td}	trust discounted statistical uncertainty
W	cardinality of the domain
$\mathcal{W}^{t_{k+1}}$	sliding window for time step t_{k+1}
X	random variable, e.g. describing that the reliability estimation works
$\mathbb{X}$	domain
$\mathbb{X}_{\text{po}}$	domain for the partially observable case
$\overline{X}$	random variable describing that the reliability estimation does not work
$\boldsymbol{x}_{t_k}$	state distribution at discrete time step t_k
Z	discrete state space of Markov chain
$\overline{\boldsymbol{z}}$	center of histogram bins

Other Symbols

α	coefficient

$\mathrm{card}\{\cdot\}$	cardinality operator
d_E	Euclidean distance
$\emptyset$	empty set
$\mathcal{G}(\cdot)$	graph
N	natural number indicating the diminsion of a vector, list, Markov chain, or finite set
$\mathbb{N}$	natural numbers
$\mathbb{R}$	real numbers
$\mathbb{R}^+$	positive real numbers
$\mathcal{L}$	general list
$\mathcal{S}$	general set
$\mathcal{S}_{\text{honest}}$	set of honest agents
$\mathcal{S}_{\text{misb}}$	set of misbehaving agents
θ	general threshold
w	weight

Bibliography

[17] *Intelligent Transport Systems (ITS); Security; Security header and certificate formats*. ETSI, 2017.

[AAB+19] Ambrosin, M.; Alvarez, I.; Buerkle, C.; et al.: *Object-level Perception Sharing Among Connected Vehicles*. In: *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, pages 1566–1573, Auckland, New Zealand, 2019.

[AG17] Amin, S. and Galasso, F.: *Geometric proposals for faster R-CNN*. In: *Proc. IEEE Int. Conf. on Adv. Video Signal Based Surveillance (AVSS)*, pages 1–6, Lecce, Italy, 2017.

[AG19] Azzedin, F. and Ghaleb, M.: *Internet-of-Things and Information Fusion: Trust Perspective Survey*. In: *Sensors*, volume 19, no. 8, page 1929, 2019.

[AGH+20] Alanwar, A.; Gassmann, V.; He, X.; et al.: *Privacy Preserving Set-Based Estimation Using Partially Homomorphic Encryption*. In: *arXiv preprint arXiv:2010.11097*, 2020.

[AY20] Afrin, T. and Yodo, N.: *A survey of road traffic congestion measures towards a sustainable and resilient transportation system*. In: *Sustainability*, volume 12, no. 11, page 4660, 2020.

[BFC90] Bar-Shalom, Y.; Fortmann, T.; and Cable, P.: *Tracking and data association*. 1990.

[BFG18] Belagiannis, V.; Farshad, A.; and Galasso, F.: *Adversarial network compression*. In: *Proc. Europ. Conf. Computer Vision (ECCV) Workshops*, Munich, Germany, 2018.

[Bis06] Bishop, C.: *Pattern recognition and machine learning*. Springer, 2006.

[BMH+21] Buchholz, M.; Müller, J.; Herrmann, M.; et al.: *Handling Occlusions in Automated Driving Using a MEC Server-based Environment Model From Infrastructure Sensors*. accepted for publication in Intell. Transp. Syst. Mag. 2021.

[BNKZ17] Banzhaf, H.; Nienhüser, D.; Knoop, S.; and Zöllner, J. M.: *The future of parking: A survey on automated valet parking with an outlook on high density parking*. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1827–1834, Los Angeles, CA, USA, 2017.

[BSMM07] Bronshtein, I.; Semendyayev, K.; Musiol, G.; and Mühlig, H.: *Handbook of Mathematics*. Springer Berlin Heidelberg, 2007.

[CD19] Caillet, P. and Dupuis, Y.: *Efficient LiDAR data compression for embedded V2I or V2V data handling*. In: *arXiv preprint arXiv:1904.05649*, 2019.

[CRC+19] Cui, H.; Radosavljevic, V.; Chou, F.; et al.: *Multimodal Trajectory Predictions for Autonomous Driving using Deep Convolutional Networks*. In: *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 2090–2096, Montreal, QC, Canada, 2019.

[DGBD19] Danzer, A.; Griebel, T.; Bach, M.; and Dietmayer, K.: *2D Car Detection in Radar Data with PointNets*. In: *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, pages 61–66, Auckland, New Zealand, 2019.

[DHDK14] Dietzel, S.; van der Heijden, R.; Decke, H.; and Kargl, F.: *A Flexible, Subjective Logic-Based Framework for Misbehavior Detection in V2V Networks*. In: *Pro. IEEE Int. Sympo. World Wireless, Mobile and Multimedia Netw. (WoWMoM)*, pages 1–6, Sydney, NSW, Australia, 2014.

[DKJ18] Dorri, A.; Kanhere, S.; and Jurdak, R.: *Multi-Agent Systems: A Survey*. In: *IEEE Access*, volume 6, pages 28573–28593, 2018.

[DML+16] Du, M.; Mei, T.; Liang, H.; et al.: *Drivers visual behavior-guided RRT motion planner for autonomous on-road driving*. In: *Sensors*, volume 16, no. 1, page 102, 2016.

[DTHT14] Dezert, J.; Tchamova, A.; Han, D.; and Tacnet, J.-M.: *Can we trust subjective logic for information fusion?* In: *Proc. IEEE Int. Conf. Inform. Fusion (FUSION)*, pages 1–8, Salamanca, Spain, 2014.

[EKSX+96] Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X.; et al.: *A density-based algorithm for discovering clusters in large spatial databases with noise.* In: *Kdd*. Volume 96, pages 226–231, 1996.

[Ell63] Elliott, E.: *Estimates of error rates for codes on burst-noise channels*. In: *The Bell System Technical Journal*, volume 42, no. 5, pages 1977–1997, 1963.

[EP23] Eggenberger, F. and Pólya, G.: *Über die Statistik verketteter Vorgänge*. In: *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, volume 3, no. 4, pages 279–289, 1923.

[ETS14] ETSI EN 302 637-2: *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*. Final draft V1.3.1. 2014.

[ETS19] ETSI TR 103 562: *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2*. V2.1.1. 2019.

[EVM+19] Englert, T.; Völz, A.; Mesmer, F.; Rhein, S.; and Graichen, K.: *A software framework for embedded nonlinear model predictive control using a gradient-based augmented Lagrangian approach (GRAMPC)*. In: *Optimization and Engineering*, volume 20, no. 3, pages 769–809, 2019.

[EWS21] Eskandarian, A.; Wu, C.; and Sun, C.: *Research Advances and Challenges of Autonomous and Connected Ground Vehicles*. In: *IEEE Trans. Intell. Transp. Syst.*, volume 22, no. 2, pages 683–711, 2021.

[FHW16] Fassbender, D.; Heinrich, B.; and Wuensche, H.: *Motion planning for autonomous vehicles in highly constrained urban environments*. In: *Proc. IEEE/RSJ Int. Conf. on Intell. Robots Systems (IROS)*, pages 4708–4713, Daejeon, South Korea, 2016.

[FXWW20] Fu, F.; Xue, T.; Wu, Z.; and Wang, D.: *A Fault Diagnosability Evaluation Method for Dynamic Systems Without Distribution Knowledge*. In: *IEEE Trans. Cybernetics*, pages 1–11, 2020.

[GDMH19] Gabb, M.; Digel, H.; Müller, T.; and Henn, R.: *Infrastructure-supported Perception and Track-level Fusion using Edge Computing*. In: *Proc. IEEE Intell. Vehicles Symp. (IV)*, pages 1739–1745, Paris, France, 2019.

[GGD18] Gritschneder, F.; Graichen, K.; and Dietmayer, K.: *Fast Trajectory Planning for Automated Vehicles Using Gradient-Based Nonlinear Model Predictive Control*. In: *Proc. IEEE/RSJ Int.Conf. Intell. Robots Syst. (IROS)*, pages 7369–7374, Madrid, Spain, 2018.

[GHT+16] Gritschneder, F.; Hatzelmann, P.; Thom, M.; Kunz, F.; and Dietmayer, K.: *Adaptive learning based on guided exploration for decision making at roundabouts*. In: *Proc. IEEE Intell. Vehicles Symp. (IV)*, pages 433–440, Gothenburg, Sweden, 2016.

[Gib13] Gibbs, R.: *New Kalman filter and smoother consistency tests*. In: *Automatica*, volume 49, no. 10, pages 3141–3144, 2013.

[Gil60] Gilbert, E.: *Capacity of a burst-noise channel*. In: *The Bell System Technical Journal*, volume 39, no. 5, pages 1253–1265, 1960.

[GK10] Graichen, K. and Kugi, A.: *Stability and Incremental Improvement of Suboptimal MPC Without Terminal Constraints*. In: *IEEE Trans. Automat. Contr.*, volume 55, no. 11, pages 2576–2580, 2010.

[GMBD20] Griebel, T.; Müller, J.; Buchholz, M.; and Dietmayer, K.: *Kalman Filter Meets Subjective Logic: A Self-Assessing Kalman Filter Using Subjective Logic*. In: *Proc. IEEE Int. Conf. Inf. Fusion (FUSION)*, pages 1–8, Rustenburg, South Africa, 2020.

[GMT+16] Günther, H.; Mennenga, B.; Trauer, O.; Riebl, R.; and Wolf, L.: *Realizing collective perception in a vehicle*. In: *Proc. IEEE Veh. Netw. Conf. (VNC)*, pages 1–8, Columbus, OH, USA, 2016.

[Gra12] Graichen, K.: *A fixed-point iteration scheme for real-time model predictive control*. In: *Automatica*, volume 48, no. 7, pages 1300–1305, 2012.

[Gra15] Graichen, K.: *Nonlinear MPC with fixed-point iterations: numerical and experimental validation*. In: *IFAC-PapersOnLine*, volume 48, no. 25, pages 62–67, 2015.

[GSMA19] Ge, J.; Schürmann, B.; Murray, R.; and Althoff, M.: *Risk-aware motion planning for automated vehicle among human-driven cars*. In: *Proc. IEEE Amer. Control Conf. (ACC)*, pages 3987–3993, Philadelphia, PA, USA, 2019.

[HDLK18] van der Heijden, R.; Dietzel, S.; Leinmüller, T.; and Kargl, F.: *Survey on Misbehavior Detection in Cooperative Intelligent Transportation Systems*. In: *IEEE Communications Surveys Tutorials*, volume 21, no. 1, pages 779–811, 2018.

[HFD11] Houska, B.; Ferreau, H.; and Diehl, M.: *ACADO Toolkit - An Open Source Framework for Automatic Control and Dynamic Optimization*. In: *Optimal Control Applications and Methods*, volume 32, no. 3, pages 298–312, 2011.

[HGS+14] Hartmann, O.; Gabb, M.; Schüle, F.; Schweiger, R.; and Dietmayer, K.: *Robust and Real-Time Multi-Cue Map Verification for the Road Ahead*. In: *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, pages 894–899, Qingdao, China, 2014.

[HMSB19] Herrmann, M.; Müller, J.; Strohbeck, J.; and Buchholz, M.: *Environment Modeling Based on Generic Infrastructure Sensor Interfaces Using a Centralized Labeled-Multi-Bernoulli Filter*. In: *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, pages 2414–2420, Auckland, New Zealand, 2019.

[HNR68] Hart, P.; Nilsson, N.; and Raphael, B.: *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*. In: *IEEE Trans. Sys. Science and Cybernetics*, volume 4, no. 2, pages 100–107, 1968.

[HP20] Hosseini, M. and Parvania, M.: *Computationally Efficient Formulations for Fault Isolation and Service Restoration in Distribution Systems*. In: *IEEE Power Energy Society General Meeting (PESGM)*, pages 1–5, Montreal, QC, Canada, 2020.

[HPS+20] Herrmann, M.; Piroli, A.; Strohbeck, J.; Müller, J.; and Buchholz, M.: *LMB Filter Based Tracking Allowing for Multiple Hypotheses in Object Reference Point Association*. In: *Proc. Int. Conf. on Multisensor Fusion and Integration for Intel. Sys. (MFI)*, pages 197–203, Karlsruhe, Germany, 2020.

[HSX+18] Hubmann, C.; Schulz, J.; Xu, G.; Althoff, D.; and Stiller, C.: *A Belief State Planner for Interactive Merge Maneuvers in Congested Traffic*. In: *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, pages 1617–1624, Maui, HI, USA, 2018.

[HY20] Haydari, A. and Yilmaz, Y.: *Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey*. In: *IEEE Trans. Intell. Transp. Syst.*, pages 1–22, 2020.

[HZ04] Hartley, R. and Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[HZ19] Hussain, R. and Zeadally, S.: *Autonomous Cars: Research Results, Issues, and Future Challenges*. In: *IEEE Commun. Surveys Tuts.*, volume 21, no. 2, pages 1275–1313, 2019.

[IFE17] Iacovazzi, A.; Frassinelli, D.; and Elovici, Y.: *On packet loss modeling: An empirical assessment*. In: *Proc. Int. Conf. Netw. of the Future (NoF)*, pages 33–39, London, UK, 2017.

[JIB07] Jøsang, A.; Ismail, R.; and Boyd, C.: *A survey of trust and reputation systems for online service provision*. In: *Decision support systems*, volume 43, no. 2, pages 618–644, 2007.

[JMP06] Jøsang, A.; Marsh, S.; and Pope, S.: *Exploring Different Types of Trust Propagation*. In: *Trust Management*, pages 179–192, Berlin, Heidelberg, 2006.

[Jøs16] Jøsang, A.: *Subjective Logic*. Springer International Publishing, 2016.

[Jøs97] Jøsang, A.: *Artificial reasoning with subjective logic*. In: *Proc. of the second Australian workshop on commonsense reasoning*. Volume 48, page 34, Australia, 1997.

[JSSZ19] Jesenski, S.; Stellet, J.; Schiegg, F.; and Zöllner, M.: *Generation of Scenes in Intersections for the Validation of Highly Automated Driving Functions*. In: *Proc. IEEE Intell. Vehicles Symp. (IV)*, pages 502–509, Paris, France, 2019.

[JTSZ20] Jesenski, S.; Tiemann, N.; Stellet, J.; and Zöllner, M.: *Scalable Generation of Statistical Evidence for the Safety of Automated Vehicles by the Use of Importance Sampling*. In: *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, pages 1–8, Rhodes, Greece, 2020.

[JZW17] Jøsang, A.; Zhang, J.; and Wang, D.: *Multi-source trust revision*. In: *Proc. IEEE Int. Conf. Inf. Fusion (FUSION)*, pages 1–8, Xi'an, China, 2017.

[KA20] Koschi, M. and Althoff, M.: *Set-based Prediction of Traffic Participants Considering Occlusions and Traffic Rules*. In: *IEEE Trans. Intel. Veh.*, pages 1–17, 2020.

[KEBB12] Krajzewicz, D.; Erdmann, J.; Behrisch, M.; and Bieker, L.: *Recent development and applications of SUMO-Simulation of Urban MObility*. In: *Int. Jour. Advances Syst. and Meas.*, volume 5, no. 3&4, pages 128–138, 2012.

[Keh07] Kehl, S.: *Querregelung eines Versuchsfahrzeugs entlang vorgegebener Bahnen*. PhD thesis, University of Stutgart, 2007.

[KFZ20] Kurzer, K.; Fechner, M.; and Zöllner, M.: *Accelerating Cooperative Planning for Automated Vehicles with Learned Heuristics and Monte Carlo Tree Search*. In: *arXiv preprint arXiv:2002.00497*, 2020.

[KG14] Käpernick, B. and Graichen, K.: *The gradient based nonlinear model predictive control software GRAMPC*. In: *2014 European Control Conference (ECC)*, pages 1170–1175, Strassbourg, France, 2014.

[KHJ+19] Kendall, A.; Hawke, J.; Janz, D.; et al.: *Learning to Drive in a Day*. In: *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 8248–8254, Montreal, QC, Canada, Canada, 2019.

[Kir04] Kirk, D.: *Optimal control theory: an introduction*. Courier Corporation, 2004.

[KMB+18] Kumar, A.; Modh, A.; Babu, M.; Gopalakrishnan, B.; and Krishna, K.: *A Novel Lane Merging Framework with Probabilistic Risk based Lane Selection using Time Scaled Collision Cone*. In: *Proc. IEEE Intell. Vehicles Symp. (IV)*, pages 1406–1411, Changshu, China, 2018.

[KNW+15] Kunz, F.; Nuss, D.; Wiest, J.; et al.: *Autonomous Driving at Ulm University: A Modular, Robust, and Sensor-Independent Fusion Approach*. In: *Proc. IEEE Intell. Vehicles Symp. (IV)*, pages 666–673, Seoul, South Korea, 2015.

[KPMA19] Koschi, M.; Pek, C.; Maierhofer, S.; and Althoff, M.: *Computationally Efficient Safety Falsification of Adaptive Cruise Control Systems*. In: *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, pages 2879–2886, Auckland, New Zealand, 2019.

[KS12] Kamien, M. and Schwartz, N.: *Dynamic optimization: the calculus of variations and optimal control in economics and management*. Courier Corporation, 2012.

[KZZ18] Kurzer, K.; Zhou, C.; and Zöllner, M.: *Decentralized Cooperative Planning for Automated Vehicles with Hierarchical Monte Carlo Tree Search*. In: *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 529–536, Changshu, China, 2018.

[Liu20] Liu, B.: *A Survey on Trust Modeling from a Bayesian Perspective*. In: *Wireless Pers. Commun.*, volume 112, pages 1–23, 2020.

[LVS12] Lewis, F.; Vrabie, D.; and Syrmos, V.: *Optimal control*. John Wiley & Sons, 2012.

[LXP20] Li, Q.; Xue, H.; and Pan, Y.: *Fault Detection Observer Design for Nonlinear Discrete-Time Systems with Measurement Outliers*. In: *Int. Conf. Syst. Science and Engineering (ICSSE)*, pages 1–6, Kagawa, Japan, 2020.

[Mah36] Mahalanobis, Prasanta Chandra: *On the generalized distance in statistics*. In: *Proc. Nat. Inst. Sciences (Calcutta)*, volume 2, pages 49–55, 1936.

[MB19] Müller, J. and Buchholz, M.: *A Risk and Comfort Optimizing Motion Planning Scheme for Merging Scenarios*. In: *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, pages 3155–3161, Auckland, New Zealand, 2019.

[MB21] Müller, J. and Buchholz, M.: *Subjective Logic Reasoning: An Urn Model Intuition and Application to Connected Automated Driving*. In: *at - Automatisierungstechnik*, volume 69, no. 2, pages 111–121, 2021.

[MBSR19] Meuser, T.; Bischoff, D.; Steinmetz, R.; and Richerzhagen, B.: *Simulation Platform for Connected Heterogeneous Vehicles*. In: *Proc. Int. Conf. Veh. Tech. Intel. Trans. Sys. (VEHITS)*, pages 412–419, Heraklion,Greece, 2019.

[MEC20] MEC-View Consortium: *Verbundprojekt MEC-View Mobile Edge Computing basierte Objekterkennung für hoch- und vollautomatisiertes Fahren - Gemeinsamer Schlussbericht*. Technical Report. 2020.

[MGB19] Müller, J.; Gabb, M.; and Buchholz, M.: *A Subjective-Logic-based Reliability Estimation Mechanism for Cooperative Information with Application to IV's Safety*. In: *Proc. IEEE Intell. Vehicles Symp. (IV)*, pages 1940–1946, Paris, France, France, 2019.

[MGGB19] Müller, J.; Griebel, T.; Gabb, M.; and Buchholz, M.: *Subjective Logic-Based Identification of Markov Chains and Its Application to CAV's Safety*. In: *Proc. IEEE Connected and Automated Vehicles Symp. (CAVS)*, pages 1–5, Honolulu, HI, USA, USA, 2019.

[MHS+19] Müller, J.; Herrmann, M.; Strohbeck, J.; Belagiannis, V.; and Buchholz, M.: *LACI: Low-effort Automatic Calibration of Infrastructure Sensors*. In: *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, pages 3928–3933, Auckland, New Zealand, 2019.

[MHSB20] Müller, J.; Herrmann, M.; Strohbeck, J.; and Buchholz, M.: *Video: Motion Planning for Connected Automated Vehicles at Occluded Intersections With Infrastructure Sensors*. 2020.

[MHSB21] Müller, J.; Herrmann, M.; Strohbeck, J.; and Buchholz, M.: *Motion Planning for Connected Automated Vehicles at Occluded Intersections With Infrastructure Sensors*. arXiv: https://arxiv.org/abs/2110.11246. 2021.

[MKBH18] Müller, J.; Krebs, S.; Buchholz, M.; and Hohmann, S.: *Interval MRAC with Proven Stability Properties and Application on Induction Motor Control*. In: *Proc. IEEE Int. Conf. Control Automation (ICCA)*, pages 173–179, 2018.

[MLH+19] Meghjani, M.; Luo, Y.; Ho, Q.; et al.: *Context and Intention Aware Planning for Urban Driving*. In: *Proc. IEEE/RSJ Int.Conf. Intell. Robots Syst. (IROS)*, pages 2891–2898, Macau, China, 2019.

[MLN+19] Meuser, T.; Lieser, P.; Nguyen, T.; Böhnstedt, D.; and Steinmetz, R.: *Lifetime-aware Information Aggregation under Uncertainty for Advanced Driver Assistance Systems*. In: *Ad Hoc Networks*, volume 82, pages 56–67, 2019.

[MMSB19] Müller, J.; Meuser, T.; Steinmetz, R.; and Buchholz, M.: *A Trust Management and Misbehaviour Detection Mechanism for Multi-Agent Systems and its Application to Intelligent Transportation Systems*. In: *Proc. IEEE Int. Conf. Control Automation (ICCA)*, pages 325–331, Edinburgh,UK, 2019.

[MMT+21] Müller, M.; Müller, T.; Talkhestani, B.; et al.: *Industrial autonomous systems: a survey on definitions, characteristics and abilities*. In: *at - Automatisierungstechnik*, volume 69, no. 1, pages 3–13, 2021.

[NSV+18] Nguyen, T.; Spehr, J.; Vock, D.; et al.: *A General Reliability-Aware Fusion Concept Using DST and Supervised Learning with Its Applications in Multi-Source Road Estimation*. In: *Proc. IEEE Intell. Vehicles Symp. (IV)*, pages 597–604, Changshu, China, 2018.

[NSX+17] Nguyen, T.; Spehr, J.; Xiong, J.; et al.: *Online Reliability Assessment and Reliability-Aware Fusion for Ego-Lane Detection Using Influence Diagram and Bayes Filter*. In: *Proc. Int. Conf. on Multisensor Fusion and Integration for Intel. Sys. (MFI)*, pages 7–14, Daegu, South Korea, 2017.

[OSR20] OSRAM Ingenieurbüro Spies, Universität Ulm: *MEC-View Abschlusspräsentation: Überblick AP4 Infrastruktursensorik*. MEC-View Final Presentation. 2020.

[OZZ20] Orf, S.; Zofka, M.; and Zöllner, M.: *From Level Four to Five: Getting rid of the Safety Driver with Diagnostics in Autonomous Driving*. In: *Proc. Int. Conf. on Multisensor Fusion and Integration for Intel. Sys. (MFI)*, pages 19–25, Karlsruhe, Germany, 2020.

[PA18a] Pek, C. and Althoff, M.: *Computationally Efficient Fail-safe Trajectory Planning for Self-driving Vehicles Using Convex Optimization*. In: *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, pages 1447–1454, Maui, HI, USA, 2018.

[PA18b] Pek, Christian and Althoff, Matthias: *Efficient Computation of Invariably Safe States for Motion Planning of Self-Driving Vehicles*. In: *IEEE/RSJ Int. Conf. on Intell. Robots Systems (IROS)*, pages 3523–3530, Madrid, Spain, 2018.

[PAPP17] Pereira, A.; Anany, H.; Přibyl, O.; and Přikryl, J.: *Automated vehicles in smart urban environment: A review*. In: *Proc. Smart City Symp. Prague (SCSP)*, pages 1–8, Prague, Czech Republic, 2017.

[PCY+16] Paden, B.; Cap, M.; Yong, S. Z.; Yershov, D.; and Frazzoli, E.: *A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles*. In: *IEEE Trans. Intell. Veh.*, volume 1, no. 1, pages 33–55, 2016.

[Pic16] Piché, R.: *Online tests of Kalman filter consistency*. In: *International Journal of Adaptive Control and Signal Processing*, volume 30, no. 1, pages 115–124, 2016.

[Pis14] Pishro-Nik, H.: *Introduction to Probability, Statistics, and Random Processes*. Kappa Research, LLC, 2014.

[PKA19] Pek, C.; Koschi, M.; and Althoff, M.: *An online verification framework for motion planning of self-driving vehicles with safety guarantees*. In: *AAET-Automatisiertes und vernetztes Fahren*, pages 260–274, Braunschweig, Germany, 2019.

[PKWA17] Pek, C.; Koschi, M.; Werling, M.; and Althoff, M.: *Enhancing motion safety by identifying safety-critical passageways*. In: *Proc. IEEE Annu. Conf. Decis. Control (CDC)*, pages 320–326, Melbourne, VIC, Australia, 2017.

[Pou14] Pouyanne, Nicolas: *Pólya urn models*. In: *Proceedings of Nablus*, volume 14, pages 65–87, 2014.

[Pul11] Pullen, S.: *Augmented GNSS: Fundamentals and Keys to Integrity and Continuity*. In: *Proc. of the Institute of Navigation GNSS+*, Portland, OR, USA, 2011.

[RLNH19] Radtke, S.; Li, K.; Noack, B.; and Hanebeck, U.: *Comparative Study of Track-to-Track Fusion Methods for Cooperative Tracking with Bearings-only Measurements*. In: *Proc. IEEE Int. Conf. on Ind. Cyber Physical Syst. (ICPS)*, pages 236–241, Taipei, Taiwan, 2019.

[RM17] Rios-Torres, J. and Malikopoulos, A. A.: *A Survey on the Coordination of Connected and Automated Vehicles at Intersections and Merging at Highway On-Ramps*. In: *IEEE Trans. Intell. Transp. Syst.*, volume 18, no. 5, pages 1066–1077, 2017.

[RPL+20] Riedmaier, S.; Ponn, T.; Ludwig, D.; Schick, B.; and Diermeyer, F.: *Survey on Scenario-Based Safety Assessment of Automated Vehicles*. In: *IEEE Access*, volume 8, pages 87456–87477, 2020.

[RVVD14] Reuter, S.; Vo, B.; Vo, B.; and Dietmayer, K.: *The Labeled Multi-Bernoulli Filter*. In: *IEEE Signal Process. Lett.*, volume 62, no. 12, pages 3246–3260, 2014.

[SAR18] Schwarting, W.; Alonso-Mora, J.; and Rus, D.: *Planning and Decision-Making for Autonomous Vehicles*. In: *Annual Review of Control, Robotics, and Autonomous Systems*, volume 1, no. 1, pages 187–210, 2018.

[Sau14] Sauter, M.: *From GSM to LTE-advanced: an introduction to mobile networks and mobile broadband*. John Wiley & Sons, 2014.

[SBM+20] Strohbeck, J.; Belagiannis, V.; Müller, J.; et al.: *Multiple Trajectory Prediction with Deep Temporal and Spatial Convolutional Neural Networks*. In: *Proc. IEEE/RSJ Int.Conf. Intell. Robots Syst. (IROS)*, pages 1992–1998, Virtual, 2020.

[SES18] Siegel, J.; Erb, D.; and Sarma, S.: *A Survey of the Connected Vehicle Landscape - Architectures, Enabling Technologies, Applications, and Development Areas*. In: *IEEE Trans. Intell. Transp. Syst.*, volume 19, no. 8, pages 2391–2406, 2018.

[Sha76] Shafer, G.: *A mathematical theory of evidence*. Volume 42. Princeton university press, 1976.

[SHP09] Simon, M.; Hermitte, T.; and Page, Y.: *Intersection road accident causation: A European view*. In: *Proc. Int. Tech. Conf. Enhanced Safety of Vehicles (ESV)*, pages 1–10, Stuttgart, Germany, 2009.

[SK16] Siciliano, B. and Khatib, O.: *Springer Handbook of Robotics 2nd Ed.* Springer, 2016.

[SMHB21] Strohbeck, J.; Müller, J.; Holzbock, A.; and Buchholz, M.: *DeepSIL: A Software-in-the-Loop Framework for Evaluating Motion Planning Schemes Using Multiple Trajectory Prediction Networks*. IEEE/RSJ Int. Conf. on Intell. Robots Systems (IROS). Prague, Czech Republic, 2021.

[SRD17] Stübler, M.; Reuter, S.; and Dietmayer, K.: *Consistency of feature-based random-set Monte-Carlo localization*. In: *Europ. Conf. on Mobile Robots (ECMR)*, pages 1–6, 2017.

[The17] The MathWorks Inc.: *MATLAB 9.2 and Symbolic Math Toolbox 7.2*. The MathWorks, Inc., Natick, Massachusetts, United States. 2017.

[THH00] Treiber, M.; Hennecke, A.; and Helbing, D.: *Congested traffic states in empirical observations and microscopic simulations*. In: *Physical review E*, volume 62, no. 2, page 1805, 2000.

[TK05] Thompson, S. and Kagami, S.: *Continuous Curvature Trajectory Generation with Obstacle Avoidance for Car-Like Robots*. In: *Proc. Int. Conf. Comp. Intell. Model., Contr. and Aut. Int. Conf. Intell. Ag., Web Techn. and Inter. Comm. (CIMCA-IAWTIC'06)*, pages 863–870, Vienna, Austria, 2005.

[UMR+15] Ulbrich, S.; Menzel, T.; Reschka, A.; Schuldt, F.; and Maurer, M.: *Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving*. In: *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, pages 982–988, Gran Canaria, Spain, 2015.

[WF18] Ward, E. and Folkesson, J.: *Towards Risk Minimizing Trajectory Planning in On-Road Scenarios*. In: *Proc. IEEE Intell. Vehicles Symp. (IV)*, pages 490–497, Changshu, China, 2018.

[WW16] Wachenfeld, W. and Winner, H.: *The Release of Autonomous Vehicles*. In: *Autonomous Driving: Technical, Legal and Social Aspects*. Ed. by Maurer, M.; Gerdes, J.; Lenz, B.; and Winner, H. Springer Berlin Heidelberg, 2016, pages 425–449.

[WWM+20] Wolf, M.; Willecke, A.; Müller, J.; et al.: *Securing CACC: Strategies for Mitigating Data Injection Attacks*. In: *IEEE Veh. Netw. Conf. (VNC)*, pages 1–7, New York, NY, USA, 2020.

[WZKT10] Werling, M.; Ziegler, J.; Kammel, S.; and Thrun, S.: *Optimal trajectory generation for dynamic street scenarios in a Frenét Frame*. In: *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 987–993, Anchorage, AK, USA, 2010.

[XZLD20] Xue, T.; Zhong, M.; Li, L.; and Ding, S.: *An Optimal Data-Driven Approach to Distribution Independent Fault Detection*. In: *IEEE Trans. Indust. Informatics*, volume 16, no. 11, pages 6826–6836, 2020.

[YLCT20] Yurtsever, E.; Lambert, J.; Carballo, A.; and Takeda, K.: *A Survey of Autonomous Driving: Common Practices and Emerging Technologies*. In: *IEEE Access*, volume 8, pages 58443–58469, 2020.

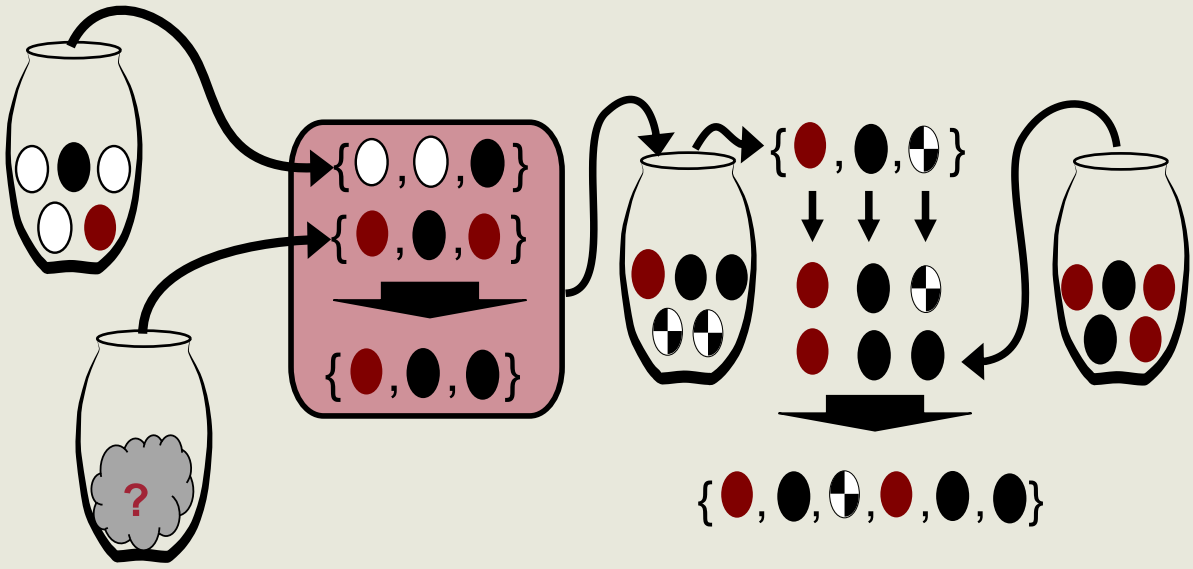
[YSL+13] Yu, H.; Shen, Z.; Leung, C.; Miao, C.; and Lesser, V.: *A Survey of Multi-Agent Trust Management Systems*. In: *IEEE Access*, volume 1, pages 35–50, 2013.

[Zad84] Zadeh, L.: *Review of a mathematical theory of evidence*. In: *AI magazine*, volume 5, no. 3, pages 81–81, 1984.

[ZBDS14] Ziegler, J.; Bender, P.; Dang, T.; and Stiller, C.: *Trajectory planning for Bertha — A local, continuous method*. In: *Proc. IEEE Intell. Vehicles Symp. (IV)*, pages 450–457, Dearborn, MI, USA, 2014.

[ZKA+17] Zidan, A.; Khairalla, M.; Abdrabou, A.; et al.: *Fault Detection, Isolation, and Service Restoration in Distribution Systems: State-of-the-Art and Future Trends*. In: *IEEE Trans. Smart Grid*, volume 8, no. 5, pages 2170–2185, 2017.

[ZTW+19] Zhou, J.; Tian, D.; Wang, Y.; et al.: *Reliability-Optimal Cooperative Communication and Computing in Connected Vehicle Systems*. In: *IEEE Trans. Mobile Comput.*, volume 19, no. 5, pages 1216–1232, 2019.



ISBN 978-3-941543-63-8

ISBN 978-3-941543-64-5 (E-Book)