JOHANNES KEPLER
UNIVERSITY LINZ

Submitted by
Admir Sovtic, BSc.

Submitted at
Institute for Design and Control of Mechatronical Systems

Supervisor
Univ.-Prof. DI Dr. Luigi del Re

Co-Supervisor
Prof. Dr. Meng Wang
DI Daniel Adelberger

March 2022

Probabilistic Prediction of Driving Behavior on Country Roads

Master's Thesis
to obtain the academic degree of
Diplom-Ingenieur
in the Master's Program
Mechatronics

JOHANNES KEPLER
UNIVERSITY LINZ
Altenbergerstraße 69
4040 Linz, Österreich
www.jku.at
DVR 0093696

Abstract

Advanced Driver Assistance Systems (ADAS) are already an important contributor to safety in conventional vehicles and thus in road traffic. This is obviously necessary, as in Austria in 2020, almost 99% of all accidents with personal injury were presumably caused by misconduct of involved persons or impairments [1]. Furthermore, country roads seem to be a safety-relevant environment, as they account for more than 50% of the fatalities in road traffic in Austria.

In order to increase the contribution of ADAS and therefore the safety in road traffic, ADAS are developed further with the goal of becoming more "intelligent" and gaining abilities like perception and assessment of road traffic situations as well as planning and execution of the further actions. One important fact in this process is the correct and precise prediction of the driving behavior of other traffic participants. While on highways and in cities, it is primarily the surrounding traffic that influences the behavior of vehicles, on country roads it is often necessary for a driver to adapt the driving behavior to the road topology, making it challenging for behavior prediction. Therefore, the aim of this work is to build a prediction model that uses the characteristics of the road to estimate the future range where vehicles will be located on country roads.

For this purpose, different motion prediction types are introduced, whereby Bayesian Networks are chosen as the prediction model type for the present thesis. Subsequently, the Bayesian Network is created and adapted to the present task and, in contrast to other works, extended with known attributes of the road topology, in order to consider them for the prediction of the driving behavior. To train the Bayesian nets and to validate the prediction concept, real world data is used and analysed regarding its topological attributes. Subsequently, the influence of the particular topological attributes on the quality of the prediction models is determined. For the validation, two performance measures concerning the accuracy and conservativeness are introduced, in order to assess the results of the prediction and to compare different approaches with each other. It is shown that especially the consideration of the curvature in combination with the sign of the slope has a positive influence on the prediction results.

In a next step, the influence of the identification strategy on the prediction results is observed, where for the identification a measure for the "richness" of a training data

set is introduced. It becomes apparent, that the “richness” and also the volume of the identification dataset influences the quality of the prediction models.

Finally, the Bayesian network model is used for an Adaptive Cruise Controller (ACC) in simulation, in order to predict the behavior of the preceding vehicle. The ACC is formulated as an optimal control problem and solved within a MPC framework, while the validation is based on a measure for the fuel economy and driving comfort. The result was, that the prediction with the Bayesian net improves the fuel economy and driving comfort in comparison to other prediction methods, which are based only on first principles.

Kurzfassung

Fahrerassistenzsysteme leisten schon heute einen wichtigen Beitrag zur Sicherheit im Straßenverkehr. Dies ist offensichtlich auch notwendig, da in Österreich im Jahr 2020 fast 99% aller Unfälle mit Personenschaden durch mutmaßliches Fehlverhalten der beteiligten Personen oder durch Beeinträchtigungen wie Alkohol oder Müdigkeit verursacht wurden [1]. Darüber hinaus scheinen Landstraßen ein sicherheitsrelevantes Umfeld zu sein, da jährlich mehr als 50% der Todesfälle im Straßenverkehr in Österreich auf Landstraßen entfallen.

Um den Einfluss von Fahrerassistenzsystemen und damit die Sicherheit im Straßenverkehr zu erhöhen, werden Fahrerassistenzsysteme mit dem Ziel weiterentwickelt, "intelligenter" zu werden. Das bedeutet, dass sie Anwendungen wie die Wahrnehmung und Bewertung von Situationen im Straßenverkehr und die Planung und Entscheidung des weiteren Vorgehens abdecken sollen. Ein wichtiger Faktor ist dabei die richtige und präzise Vorhersage des Fahrverhaltens anderer Verkehrsteilnehmer. Während auf Autobahnen und in Städten vor allem der Umgebungsverkehr das Verhalten der Fahrzeuge beeinflusst, ist es auf Landstraßen oft notwendig, dass ein Fahrer sein Fahrverhalten an die topologischen Eigenschaften der Strecke anpasst, was die Verhaltensprognose zu einer Herausforderung macht. Ziel dieser Arbeit ist es daher, ein Prädiktionsmodell zu entwickeln, das die topologischen Eigenschaften der Straße nutzt, um den zukünftigen Bereich abzuschätzen, in dem sich Fahrzeuge auf Landstraßen befinden werden.

Zu diesem Zweck werden verschiedene Ansätze zur Prädiktion vorgestellt, wobei Bayes'sche Netze für die weitere Verwendung in der vorliegenden Arbeit gewählt werden. Im Gegensatz zu anderen Arbeiten werden die Bayes'schen Netze um bekannte Attribute der Straßentopologie erweitert, um diese bei der Vorhersage des Fahrverhaltens zu berücksichtigen. Zum Training der Bayes'schen Netze und zur Validierung des Prädiktionskonzepts werden reale Daten verwendet und hinsichtlich ihrer topologischen Eigenschaften analysiert. Anschließend wird der Einfluss der einzelnen topologischen Attribute auf die Qualität der Prädiktionsmodelle bestimmt. Für die Validierung werden zwei Leistungsmaße hinsichtlich der Genauigkeit und Konservativität eingeführt, um die Ergebnisse der Vorhersage zu bewerten und verschiedene Ansätze miteinander zu vergleichen. Es zeigt sich, dass insbesondere die Berücksichtigung der Straßenkrümmung in Kombination mit dem Vorzeichen der Steigung einen positiven Einfluss auf die Prädiktionsergebnisse hat.

Im nächsten Schritt wird der Einfluss der Identifikationsstrategie auf die Prädiktionsergebnisse betrachtet, wobei für die Identifikation ein Maß für die "Reichhaltigkeit" eines Trainingsdatensatzes eingeführt wird. Es zeigt sich, dass die "Reichhaltigkeit" als auch der Umfang des Identifikationsdatensatzes die Qualität der Prädiktionsmodelle beeinflusst.

Schließlich wird ein ausgewähltes Bayes'sches Netzwerk in Simulation für einen Abstandsregeltempomat (ACC) verwendet, um das Verhalten des vorausfahrenden Fahrzeugs vorherzusagen. Der ACC wird als optimal-control Problem formuliert und im Rahmen eines MPC gelöst, während die Validierung auf einem Maß für den Kraftstoffverbrauch und Fahrkomfort basiert. Es zeigt sich bei der Prädiktion mit dem Bayes'schen Netz eine Verbesserung des Kraftstoffverbrauchs und Fahrkomforts im Vergleich zu anderen Prädiktionsmethoden, die keine topologischen Eigenschaften berücksichtigen.

Contents

Chapter 1

Introduction

This chapter will give introductory information about the background and motivation of this work. Furthermore, the topic of motion prediction will be discussed, in order to give an overview of possible approaches, existing works and relations to the current work. Finally, the structure of the thesis is shortly described.

1.1 Background

In the past decades, the trend to more "intelligent" vehicles evolved to the goal of developing autonomously or self-driving cars. This goal was partly achieved by some car manufacturers, but it will still need time until these cars will be integrated in regular traffic due to remaining technological and legal challenges. These include finding the optimal sensing modality for localization, mapping and perception, accuracy and efficiency lacking algorithms and the need for proper online assessment, as it was shown in [2]. Besides the technological issues, there are also ethical and especially legal concerns, e.g., when it comes to accidents, whereby one main advantage of self driving cars actually could be the reduction of accidents. In Austria in 2020, almost 99% of all accidents with personal injury were presumably caused by misconduct of involved persons or impairments, as it can be seen in Table 1.1. The main contributors to this figure are carelessness or distraction, failure to give way, non-adapted speed and lack of safe distance.

In order to address this issue and to support the driver in conventional non-self-driving cars, Advanced Driver Assistance Systems (ADAS) are deployed. Certain ADAS have been utilized since decades and already belong to the standard equipment of conventional vehicles. These are, for example:

- Anti-lock Braking System (ABS)
- Traction Control System (TCS)
- Electronic Stability Control (ESC)
- Emergency Brake Assist, Limited-slip differential etc.

Table 1.1: Overview of the presumed main causes of accidents in Austria in 2020 (from [1])

Presumed main causes of accidents		Total
Misconduct by persons involved		90,6%
thereof	Non-adapted speed	16,1%
	Failure to give way (also to pedestrians), disregarding red lights	25,0%
	Overtaking	2,8%
	Carelessness / distraction	28,4%
	Misbehaviour of pedestrians	2,5%
	Lack of safe distance	10,9%
	Disregarding bans and prohibitions (e.g., driving against one-way system, bans on turning)	4,9%
Impairments		8,4%
thereof	Alcohol, drugs or medication	6,3%
	Fatigue	0,9%
	Health impairments	1,2%
Vehicle-specific causes		1,1%
	Technical defect, inadequate load securing	1,1%
Total		100,0%

Other newly developed ADAS support the driver and take over more and more tasks in order to increase the safety, raise the driving comfort or lower the fuel consumption:

- Adaptive Cruise Control (ACC)
- Lane Keep or Lane Change Assistance (LKA)
- Autonomous Emergency Braking (AEB)
- etc.

These ADAS have different core functions like localization and mapping, perception, assessment, planning and decision making, vehicle control, and human-machine interface [2]. One of the most important tasks for planning and decision making is to estimate the location of other road users in future situations as accurately as possible, without being too conservative. The question is always, which factors influence a human driver and subsequently, which actions are set by the driver to cope with those factors. This of course depends highly on the respective type of environment. Additionally, not only influence factors in the observable area, but also limited observability may play a role. Perception limitations [3] pose a problem for ADAS systems [4] and are especially due to safety reasons subconsciously considered by human drivers [5]. These limitations can be overcome through vehicle-to-everything (V2X) communication [6] which, however, cannot yet be assumed to be realistically available in the near future [7]. Many prediction approaches are based on reachable sets [8, 9, 10], while

probabilistic methods which reduce conservativeness are also well known [11, 12]. Other authors combine first principles with possible maneuvers in the environment [13, 14] to improve prediction performance. A general overview of different motion prediction types will be given in Section 1.2.

Compared to urban areas or highways, the road topology seems to play a significant role for drivers on country roads, which are – as they account for 54.7 % of the fatalities in road traffic in Austria [1] – a very relevant environment. In Figure 1.1 the development over time of the fatalities in road traffic in Austria can be seen broken down by the road type between 2011 and 2020. It is obvious that country roads accounted for at least the half of all fatalities each year.

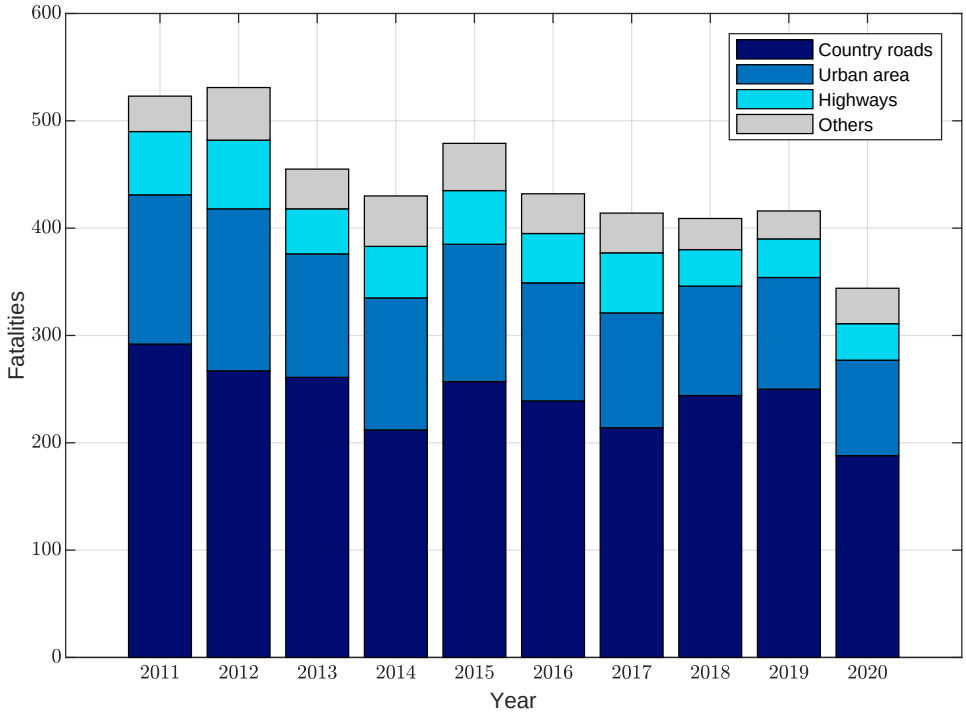

Figure 1.1: Temporal development of fatalities in road traffic in Austria broken down by road type (data from [1]).

When driving along a country road, the topology of the surroundings, such as the curvature of the road, the slope, and also the visible range, is – in most cases – subconsciously incorporated into the driving decision of human drivers to remain in a safe state [5]. The question arises, if it is possible to determine the significance of the respective influence factors and if this knowledge can be used to create a realistic prediction model.

1.2 Overview of motion prediction types

This section is mainly based on the research and findings of Lefèvre et al. in [15].

In order to assess particular situations in traffic regarding the ego-vehicle's risk of being endangered by an accident, mathematical models are needed, which are able to predict how these situation will evolve in future. These models can be classified in three groups with an increasing level of abstraction:

1. Physics-based motion models:
   These models represent the simplest way of modelling, as only physical laws are considered for the motion prediction.
2. Maneuver-based motion models:
   These models additionally include the estimated driver intention in order to determine the future maneuver of the vehicle.
3. Interaction-aware motion models:
   In this case the inter-dependencies between vehicles' maneuvers are considered to estimate the future motion.

In the following, these three groups or types of motion prediction will be discussed more detailed.

1.2.1 Physics-based motion models

Physics-based motion models (also known as first principles) predict the evolution of physical states of a vehicle like position and velocity based on particular inputs, e.g., acceleration or steering angle. Due to the mostly known underlying physical laws and therefore more or less complex modelling, these models are widely used for trajectory prediction and collision risk estimation in the context of road safety. The complexity of these models depends on the underlying problem and how fine and real-life-similar the representation of the vehicle has to be.

The modelling of the evolution of the physical states of a vehicle can be classified in two types:

1. Dynamic models:
   For these types of models the vehicle and its behavior are described by the use of Newton's laws or respectively Langrange's equations, in order to consider the different forces that affect the motion. These forces can, e.g., be the lateral and longitudinal tire force or the air resistance. As the behavior of car-like vehicles is based on complex coherences from the driver to the street, dynamic models can get very complex, dependent on the existing task and parameters which have to be considered. Usually, more complex models are needed for control tasks, while for trajectory prediction simpler models are used. For example, in [16]

such a "simple" dynamic model of a vehicle can be found, which is also directly compared to a kinematic vehicle model for the same task:

$$\ddot{x} = \dot{\psi}\dot{y} + a_x \tag{1.1}$$

$$\ddot{y} = -\dot{\psi}\dot{x} + \frac{2}{m}\left(F_{c,f}\cos\delta_f + F_{c,r}\right) \tag{1.2}$$

$$\ddot{\psi} = \frac{2}{I_z}\left(l_f F_{c,f} - l_r F_{c,r}\right) \tag{1.3}$$

$$\dot{X} = \dot{x}\cos\psi - \dot{y}\sin\psi \tag{1.4}$$

$$\dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi \tag{1.5}$$

In this case, $\dot{x}$ and $\dot{y}$ denote the longitudinal and lateral velocities in the body frame, and $\dot{X}$ and $\dot{Y}$ the respective velocities in the inertial frame. $\dot{\psi}$ denotes the yaw rate, m and I_z denote the vehicle's mass and yaw inertia, $F_{c;f}$ and $F_{c,r}$ denote the lateral tire forces at the front and rear wheels in coordinate frames aligned with the wheels. l_f and l_r represent the distance from the center of the mass of the vehicle to the front and rear axles. a_x and δ_f are the inputs of the model, and represent the acceleration in longitudinal direction and the steering angle.

2. Kinematic models:
Kinematic models are far more popular for trajectory prediction than dynamic models, as they describe the motion based on the mathematical relationships between the physical parameters of the motion like position, velocity and acceleration. Hence, kinematic models do not consider forces which affect the motion or complex dependencies inside the vehicle. Therefore, kinematic models are also better suited for trajectory prediction involving more traffic participants, like in ITS (Intelligent Transportation Systems), as no internal parameters of other participants are needed, as it would be necessary for dynamic evolution models. The simplest kinematic models are the Constant Velocity (CV) and Constant Acceleration (CA) models, which both assume, that the motion is straight and the velocity respectively the acceleration remain constant until the next time step. This might sound quite trivial, but it is sufficient for many cases. A more complex but still simple model can be found, e.g., in [17]:

$$\dot{x}(t) = v(t)\cos\left(\theta(t)\right) \tag{1.6}$$

$$\dot{y}(t) = v(t)\sin\left(\theta(t)\right) \tag{1.7}$$

$$\dot{v}(t) = a(t) \tag{1.8}$$

$$\dot{\theta}(t) = \frac{v(t)}{D_a}\tan\left(\delta(t)\right) \tag{1.9}$$

with a visualization of the quite common single-track or "bicycle" representation of a vehicle, as depicted in Figure 1.2.

The next question is now, how the trajectory prediction is performed and mainly how uncertainties are handled. Therefore, three different approaches were given in [15]:

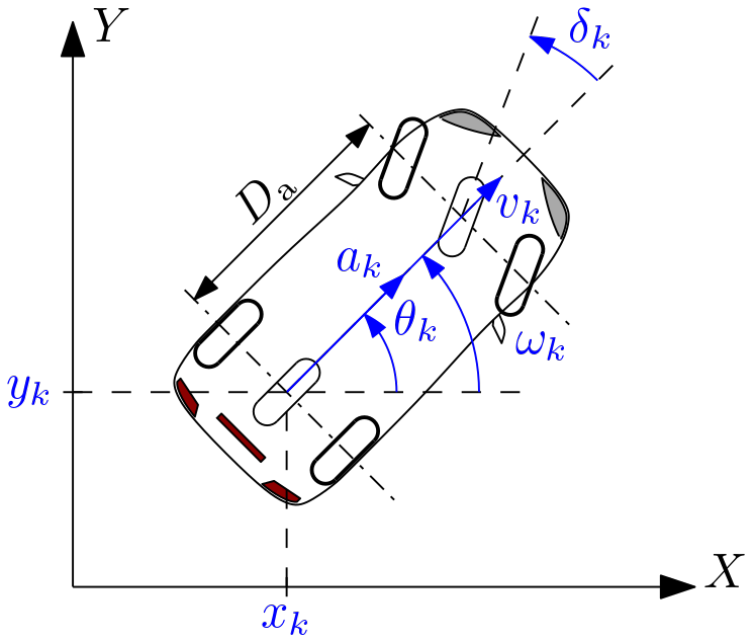

Figure 1.2: Representation of a car as a two-wheeled vehicle with the corresponding attributes for the kinematic model (from [17])

1. Single trajectory simulation:
   In this approach, it is assumed that the current state is perfectly known and the dynamic or kinematic evolution model is a perfect representation of the motion of the vehicle. The future trajectory is therefore predicted by simply applying the evolution model to the current state of the vehicle. On the one hand, this approach is quite straightforward and computationally efficient, but on the other hand, uncertainties on the current state or the shortcomings of the evolution model can not be considered. Therefore, such approaches are not reliable for long term predictions.

2. Gaussian noise simulation:
   The states of a vehicle like position and velocity can be measured with sensors, whereby these measurements appear to be noisy, which is an uncertainty as the state can not be determined exactly. This uncertainty can be modelled by a normal distribution, whereby the current state can be estimated with Kalman Filters. Firstly, the evolution model is applied to the current estimated state $x(t)$, resulting in a predicted state $\hat{x}(t+1)$ which is distributed normally. In the second step the sensor measurements at time $t+1$ are combined with the predicted state, resulting in an estimated state $x(t+1)$. By repeating the prediction in a certain state, one can obtain a mean trajectory with associated uncertainty at each future time step, as done in [18].

   The problem with these approaches is, that modelling the uncertainties with an unimodal normal distribution is insufficient to represent the different possible maneuvers. One solution are Switching Kalman Filters, which use mixtures of Gaussians to represent the uncertainty.

3. Monte Carlo simulation:
   Monte Carlo methods provide tools to approximate the distribution on the predicted states, when no assumption can be made about the Gaussianity of the uncertainties. Thereby, one randomly samples from the input variables of the evolution model in order to generate potential future trajectories starting from the current state. This results in a set of possible future trajectories, which do not all have to be feasible in real-life regarding the road topology or the feasibility of a maneuver. To take this into account, unfeasible trajectories can be removed or penalized with weights, or the physical limitations could be considered in the evolution model.

As stated so far, physics-based motion models are only based on low level properties of the motion like dynamic or kinematic attributes. Therefore, they are unable to anticipate any change in motion caused by a particular maneuver or by external factors like the change of the driver intention based on the road topology. This is the reason why this type of models is limited to short-term prediction and is not suitable for the present work.

1.2.2 Maneuver-based motion models

Maneuver-based motion models represent the motion of a vehicle on a road network as a series of consecutively executed maneuvers. These maneuvers can be, for example, stop, go straight, turn left, lane change left etc., and are assumed to be executed independently of other vehicles. The prediction of the future motion of a vehicle is therefore based on the early recognition of the maneuver which the driver is intending to perform. If the maneuver is identified, it is assumed that the future motion of the vehicle will match the maneuver. In order to recognize the intended maneuver properly, two different approaches are given in [15]:

1. Prototype trajectories:
   This approach is based on the idea, that the road network is a structured environment and therefore all possible trajectories can be clustered in groups, whereby every group represents a typical motion pattern or maneuver. These motion patterns are learned from previously observed data, where several trajectories of a motion pattern are agglomerated to one prototype trajectory for this particular maneuver. This can be done in various ways, e.g., with a stochastic representation meaning the mean trajectory and a standard deviation, with several prototype trajectories out of the training data or with Gaussian processes. These appeared to be well-suited for the representation of motion patterns (as seen in [19]) because of their robustness to noise and to variabilities in the velocity due to varying traffic conditions, with the disadvantage of a high computational complexity.

   After learning of the motion patterns, the prediction is based on finding the motion pattern which is the most similar to the partial trajectory executed by the vehicle so far. The similarity between the partial trajectory and the

motion patterns can be measured with the probability that the trajectory corresponds to a Gaussian Process, with the Euclidian distance between points of the trajectories, the Longest Common Subsequence etc. After the matching of the partial trajectory with the motion pattern, the simplest way to predict the future motion is to use the prototype trajectory of the motion pattern. Another possibility is to use a mixture of motion patterns and to combine them by weighing in terms of probability, or to pick a weighted subset of probable trajectories.

The main disadvantage of this approach is the adaption of the prototype trajectories to different road layouts, as each motion model is trained for one specific intersection geometry and topology. Therefore, they can only be reused at intersections with a similar layout as the training intersection.

2. Maneuver recognition and execution:
The main difference and advantage of this approach is, that the maneuver intention of the driver is estimated based on higher-level characteristics instead of matching a partial trajectory with a prototype trajectory, which makes it easier to generalize the learnt model to different layouts. Among these characteristics are, for example, the physical state of the vehicle, information about the road network or the driver behavior. These characteristics can then be used in heuristic approaches, discriminative learning algorithms like Multi-Layer Perceptrons, Logistic Regression or Support Vector machines, or in Hidden Markov Models [20] in order to determine what maneuvers are likely to be performed in the near future.

   After recognizing the maneuver, the future trajectory is predicted, so that it matches the identified maneuver. This can be done by deriving the input controls for the corresponding maneuver and applying it to a kinematic motion model, which leads to a deterministic future trajectory. The drawback of this approach is, that uncertainties on the current state or the maneuver can not be considered. A solution to this problem are Gaussian Processes or Rapidly-exploring Random Trees (RRT) [21].

The main limitation of maneuver-based motion models is the assumption that vehicle's motion is independent of the other traffic participants. In practice, the maneuvers of one vehicle necessarily affect the maneuvers of other vehicles. These inter-dependencies are particularly strong at road intersections, where priority rules force vehicles to take into account the maneuvers performed by the other vehicles.

1.2.3 Interaction-aware motion models

Interaction-aware motion models take up the drawback of maneuver-based motion models and represent vehicles as maneuvering entities which interact with each other. This leads to a better interpretation of the motion, a better understanding of situations in road traffic and a more reliable evaluation of the risk. These models are mainly based on prototype trajectories and Dynamic Bayesian Networks:

1. Trajectory prototypes:
   This approach is already known from maneuver-based motion models. Just like in this type of motion prediction, the inter-vehicle dependencies can not be considered during the training phase as the resulting number of motion patterns would quickly become intractable. The difference here is, that mutual influences can be taken into account when matching the partial trajectory with a motion pattern. The background is the assumption, that drivers have a strong tendency to avoid collisions when they can. Therefore, pairs of trajectories which lead to an unavoidable collision are penalized in the matching process, whereby safe trajectories are always considered more likely than hazardous ones. This is an elegant workaround, but still the mutual influences can not be modelled directly.
2. Dynamic Bayesian Networks:
   Most interaction-aware motion models are based on Dynamic Bayesian Networks. Other similar approaches are (Coupled) Hidden Markov Models [22], which are especially used for modelling of pairwise dependencies between multiple moving entities. These are usually made asymmetric, which means that it is assumed that the surrounding traffic affects the vehicle of interest, but not vice versa, which reduces the computational complexity significantly.

Interaction-aware motion models are the most comprehensive models proposed so far in literature. They allow longer-term predictions and are more reliable as they consider inter-vehicle dependencies. One fact, which is to be improved in future, is the computational complexity and real-time risk assessability.

1.3 Structure and aim of this work

In various works, Bayesian networks have been used to predict the behavior of surrounding vehicles in different traffic scenarios to improve fuel economy, driving comfort and safety [23, 24, 25] and also for overtaking [26], but the focus lays mostly on the vehicles, and topological factors of the environment are hardly considered.

This work aims to overcome this gap by exploiting the property of Bayesian networks to map the effect of variables on a predicted quantity in a targeted manner. Since these networks enable to map the effect of individual variables on a predicted variable, they are also used in this work, utilizing the toolbox from [27]. Different topological influence factors on country roads are discussed and their significance on driving behavior is studied by incorporating them in Bayesian network-based prediction models, which additionally rely on first principles. Subsequently, a combined model is applied on a simple ADAS in comparison with other prediction methods to investigate the advantages of considering topological factors for prediction.

The rest of the work is structured as follows:

- In a first step, the underlying environment is discussed in Chapter 2. Furthermore, a general introduction to Bayesian networks is given, in order to explain the deployment for this thesis, and to declare the applied prediction concept.
- Chapter 3 gives an overview of the collection and analysis of the data, which was used for identification and validation.
- In Chapter 4, the influence of particular topological attributes on the quality of the prediction is observed. Therefore, two performance measures are introduced.
- The influence of different identification strategies on the quality of the prediction is observed in Chapter 5.
- The prediction approach is subsequently implemented on an ACC system in Chapter 6, in order to compare its performance with other approaches.

A majority of the concepts in this thesis can also be found in [28], which was created in the course of the master thesis.

Chapter 2

Conception and conditions of the prediction

In this chapter, as a first basic step, the representation of the environment will be discussed. In a next step, several basic concepts, attributes and examples of Bayesian networks will be covered, which are relevant for the present work. Finally, the concrete concept for the prediction will be explained.

2.1 Representation of the environment

The prediction within this work is done based on topological influences like curvature or road grade and first principles like position and velocity. Therefore, in contrast to other works regarding behaviour prediction, the modelling of the ego-vehicle or potential obstacle vehicles is not considered. In order to describe the topological conditions, a local street-aligned coordinate system $(X_\mathrm{R}, Y_\mathrm{R}, Z_\mathrm{R})$ is used, where the curvature of the primary lane $c(j') = 1/r(j')$ as well as the road grade $\alpha(j')$ are assumed to be given for each longitudinal position $s(j')$ along the route, with j' indicating an arbitrary spatial index. This results in a representation where the whole route can be expressed using consecutive arcs with a well-defined radius of the centre-line, see Figure 2.1. If, at an arbitrary time $t(k')$, a vehicle moves along the lane with $s(j') \leq s(k') \leq s(j'+1)$, the arbitrary spatial attribute $x(k')$ of the enclosing segment is assumed to hold for the respective time instant $x(k') \approx x(j')$ which is a realistic assumption for small $\Delta s(j')$.

The representation of the road is done with common values of conventional country roads with a assumed width of $w_\mathrm{road} = 6\,\mathrm{m}$ and margins of $m_\mathrm{l} = m_\mathrm{r} = 1\,\mathrm{m}$. The margins m_l and m_r represent possible obstacles for the sight of an observer in a position $s(j')$, like walls, houses or bushes as depicted in green in Figure 2.1. Of course, this depends strongly on the type of the country road, i.e. the margin would be significantly higher when driving through lowlands without any trees or agriculture. In this work, the route, on which the used datasets base, runs mainly through forests and so the margin of 1 m is a good representation of the circumstances of the real driver. With these 2 parameters, street width and margin, it is possible to estimate

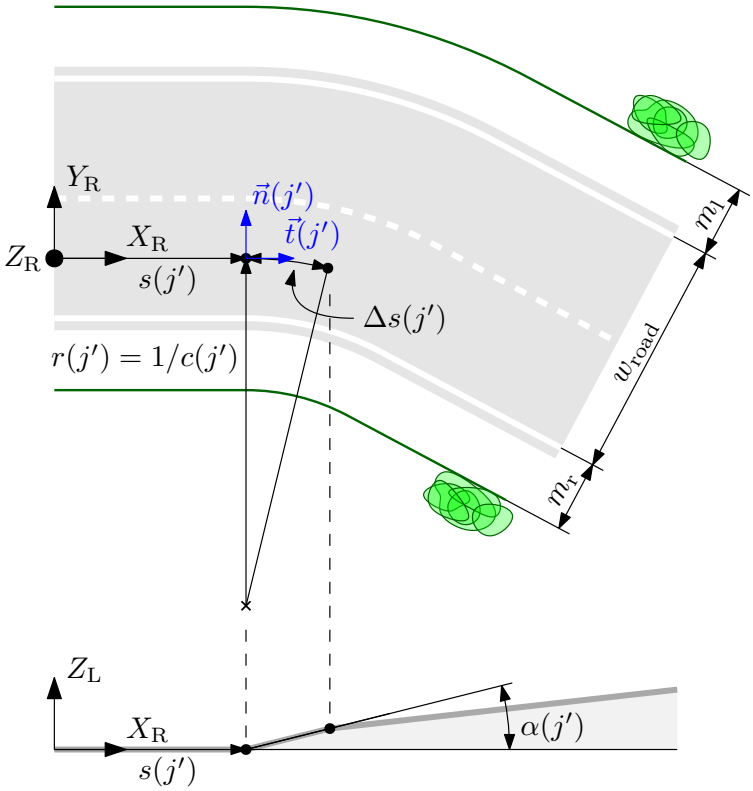

Figure 2.1: Coordinate representation as arc coordinates for the longitudinal position $s(j')$ with all respective characteristics of the road.

the visible distance $s_\mathrm{vis}(j')$ for each spatial position $s(j')$ along the route. The visible distance $s_\mathrm{vis}(j')$ is a representation of the maximum possible distance or area, which a real driver can overlook in a position $s(j')$, and so it is one of the main influences on drivers in real traffic situations, which is also taught in driver license classes. In the German-speaking world the visible distance describes the “Fahren auf Sicht”.

The estimation of the visible distance was first introduced in [29] and expanded by adding a consideration of the road grade in [5]. The idea of the estimation is to transform the Cartesian space to an ego-centered angular space, where the visible distance can be obtained by a constrained optimization, which is explained in more detail in Section 3.1.

2.2 Probabilistic Prediction with Bayesian Networks

The present work is based on a probabilistic prediction approach using Bayesian networks which contain random variables connected by conditional probabilities, as visualized in Figure 2.4. Using these networks, the parameters – i.e., the mean $\hat{s}_\mu(k'+n)$ and standard deviation $\hat{s}_\sigma(k'+n)$ – of a distribution function are estimated for an arbitrary time instant $k'+n$, as it can be seen in Figure 2.6. This distribution is then used to compute hard deterministic limits of the position range of a vehicle ($\hat{s}_{\mathrm{lb}(p)}(k'+n)$ for the lower bound of the possible position and $\hat{s}_{\mathrm{ub}(p)}(k'+n)$ for the

respective upper bound), which can be directly used in a controller. In the following two sub-chapters, Bayesian Networks will be discussed in general and it will be exemplified how they are applied and adjusted for the present problem.

2.2.1 General Introduction to Bayesian Networks

Before discussing Bayesian networks, several basic definitions of the probability theory are given in the following.

Random Variable

A Random Variable X is a variable with a fixed (finite or infinite) set of possible values $Val(X)$ – the so called domain – which represents some aspect of the system's world. Different types of random variables are:

- Boolean variables: $Val(X) = \{false, true\}$ or $\{0, 1\}$
- Discrete (categorical) variables: have a finite domain of symbolic values. Example: $Season$ with $Val(Season) = \{spring, summer, fall, winter\}$
- Continuous (real-valued) variables: $Val(X) \in \mathbb{R}$

Event

An Event is a fixed assignment of values to some or all of the variables in the system's world. An Atomic Event is an event where *all* random variables in the system's world have a specific value assigned.

Joint and Marginal Distribution

In a world defined by a set of random variables $\mathcal{X}$, the probability distribution over all atomic events possible over $\mathcal{X}$ is called Full Joint Distribution over $\mathcal{X}$. It assigns probabilities to all possible value combinations of the set $\mathcal{X}$. Marginal Distributions assign probabilities to all value combinations of a selected subset $X \subset \mathcal{X}$.

Conditional probability

Describes the probability $P(\alpha \mid \beta)$ that some (unobservable) event α is true when another observed event β is given.
Example: α describes if a patient has the corona virus and β describes if the patient has fever. Then $P(\alpha \mid \beta)$ describes the probability that the patient has corona if it is known that he has fever.

Bayesian networks

Bayesian networks are directed acyclic graphs with so called nodes and edges. The nodes stand for random variables, which are connected by edges representing conditional probability distributions.

- Directed: Every edge has a direction from a starting node X_i to a ending node X_j. This direction indicates that the starting node has an influence on the ending node, which is described by the conditional dependency resp. the connecting edge. In this case the starting node X_i is called a *Parent* of the ending node X_j, resp. X_j is a *Child* of X_i
- Acyclic: In a Bayesian net, in every path the starting node is always different to the ending node. Therefore no circles can occur in the graph, which is of vital importance to the factorization of the joint probability of a collection of nodes.
- Graph: Bayesian networks belong to the group of graphical models. Nodes are usually depicted as circles and edges are depicted as arrows.

All those concepts and the application of Bayesian nets will be clear in the following concrete example and the corresponding explanations, which are from [30].

The situation is the following, that a company wants to hire a recent college graduate. The goal is to hire intelligent persons, but it is not possible to measure or test intelligence directly. Instead, the company asks the applicants to supply the results of their SAT test (Scholastic Assessment Test), and their grade in some relevant course at the university, which could be more or less difficult. Furthermore, the student brings a recommendation letter of the professor of this course. According to this introduction, there are 5 random variables:

- Intelligence $I \in \{low, high\}$ or $\{i^0, i^1\}$
- SAT-Score $S \in \{low, high\}$ or $\{s^0, s^1\}$
- Grade $G \in \{A, B, C\}$ or $\{g^1, g^2, g^3\}$
- Difficulty of the course $D \in \{low, high\}$ or $\{d^0, d^1\}$
- Recommendation letter $L \in \{bad, good\}$ or $\{l^0, l^1\}$

These variables can be arranged to a Bayesian net as it is depicted in Figure 2.2. There we can see the causal dependencies from the introduction, where the difficulty and the intelligence affect the grade, the intelligence affects the SAT and the grade affects the recommendation letter. Of course these are not deterministic relationships, as there are uncertainties from real-life like having a bad day and therefore performing bad at the SAT, or other factors which influence the grade beside the difficulty and intelligence. These uncertainties can be modelled with probabilities, which are shown in the small tables in Figure 2.2. We distinguish two types of probabilities:

- Unconditional probabilities: associated with variables without parents
  Example: $P(i^1) = 0.3$ ... the probability that a random student is highly intelligent is 30%
- Conditional probabilities: for variables that depend on parent variables
  Example: $P(s^0 \mid i^1) = 0.2$ ... the probability that an intelligent student gets a low SAT score is 20%

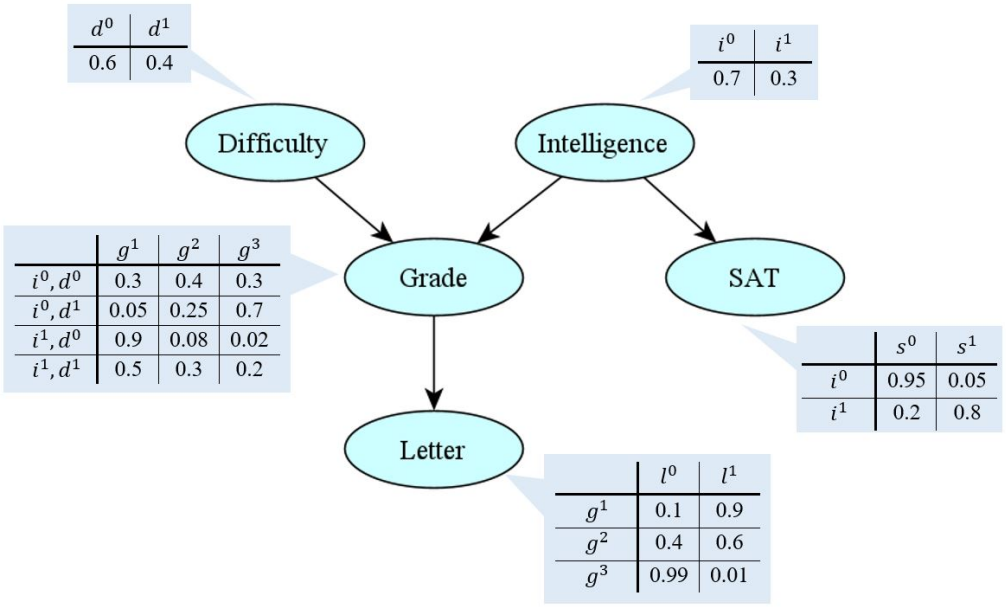

Figure 2.2: Bayesian network for the example from [30]

Each row in each of the tables is a (conditional) probability distribution CPD. With the help of these CPDs, different types of reasoning patterns in this network are possible:

1. *Causal Reasoning* or *Prediction*:
   Is a type of reasoning from causes to effects (or downwards) in the network.
   Example: We want to find out, how likely it is that a random student gets a good recommendation letter. We find out that the student is not so intelligent and therefore the probability of a strong recommendation goes down:

   $$P(l^1 \mid i^0) \approx 0.389$$

   Next we find out that the attended course was easy that year, which makes the probability go up again, because an easy class raises the probability for a good grade and a good grade raises the probability for a good recommendation letter:

   $$P(l^1 \mid i^0, d^0) \approx 0.513$$

2. *Evidential Reasoning* or *Explanation*:
   Is the counterpart to Causal Reasoning and describes the reasoning from effects (observations) to possible causes (upwards) in the network
   Example: The probability that a random student is intelligent is directly given by the model:

   $$P(i^1) = 0.3$$

If we now find out that this student got a C (g^3) in his class, the probability that the student is intelligent goes down significantly:

$$P(i^1 \mid g^3) \approx 0.079$$

Furthermore, the probability that the course is difficult goes up to from:

$$P(d^1) = 0.4 \text{ to } P(d^1 \mid g^3) \approx 0.63$$

3. *Intercausal Reasoning* or *Explaining Away*:
   This type of reasoning describes, that one causal factor between two variables X_i and X_j gives information about another casual factor between X_i and X_k.
   Example: The aforementioned student has a bad grade, but submits the SAT score which is surprisingly high (s^1). This raises the probability for intelligence, as it is more likely that an intelligent student gets a bad grade than that an intelligent student has a bad SAT score.

   $$P(i^1 \mid g^3, s^1) \approx 0.578 > P(i^1 \mid g^3) \approx 0.079$$

   Furthermore, the probability for difficulty of the class rises:

   $$P(d^1 \mid g^3, s^1) \approx 0.76 > P(d^1 \mid g^3) \approx 0.63$$

   A poor grade is a possible indicator of a difficult class, but could also be due to low intelligence. But when we learn that the student is (probably) highly intelligent, then probably the course must really have been difficult.

Special characteristics of Bayesian networks

- Conditional independencies:
  In a Bayesian network, every variable is conditionally independent of all it non-descendants, given its parents

  $$(X_i \perp \text{NonDesc}(X_i) \mid \text{Pa}(X_i)) \text{ for all } X_i \in \mathcal{X}$$

  This means in our example, if the grade is given, the letter can be determined without any other information. So, L is conditionally independent of D, I, S.

- Factorised Probability Distribution:
  Each node in the network is associated with a set of conditional probability distributions $(X_i \mid \text{Pa}(X_i))$ (the tables in Figure 2.2). The full joint distribution over all variables can be represented as a product of all local CPDs:

  $$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid \text{Pa}(X_i))$$

One instance from the example:

$$P(d^0, i^0, g^2, s^0, l^1) = P(d^0) \cdot P(i^0) \cdot P(g^2 \mid d^0, i^0) \cdot P(s^0 \mid i^0) \cdot P(l^1 \mid g^2) = 0.09576$$

So, the probability of a random student being less intelligent, the course being easy, the student achieving a B grade in this course and a low SAT score, and still receiving a good recommendation letter from the professor is 9.58%.

- Compactness of the representation:
  Consider a set $\mathcal{X} = \{X_1, \ldots, X_N\}$ of N boolean random variables in an arbitrary (not Bayesian) network. The full joint distribution $P(X_1, \ldots, X_N)$ over $\mathcal{X}$ has 2^N entries resp. different atomic events. To specify these, $2^N - 1$ independent parameters are required and therefore the representation of the full joint distribution is infeasible for reasonably large N.
  But, if we consider a sparsely connected Bayesian network with the same amount of variables N, where sparsely connected means that each variable has at most $k << N$ parents, the complexity becomes exponentially reduced from $O(2^N)$ to $O(N)$ for a fixed $k << N$. In our example, we would theoretically need $2 \cdot 2 \cdot 3 \cdot 2 \cdot 2 = 48$ independent parameters, but due to the sparse connectivity $(2 << 5)$ we only need 15 independent parameters.
- Linear Gaussian Networks:
  In our considerations so far, we assumed that all variables in a model are discrete with a finite domain $Val(X)$, which allowed us to represent the conditional probability distributions $P(X \mid \mathrm{Pa}(X))$ in the form of a table. The problem with this assumption is that many systems involve continuous aspects and measurements that are best modelled by real-valued variables, like position and velocity in our case.
  This problem can be solved with parametrisable density functions, whereby the parameters of a distribution $P(X \mid Y)$ are a function of the specific value y. As a consequence, it is not necessary to explicitly represent an infinite number of distributions $P(X \mid y)$. A special and very common case are linear gaussian networks, where all distributions in the net are modelled as Gaussians $\mathcal{N}(\mu_i; \sigma_i^2)$. For our case, we will also use Gaussian distributions, although we have measurements of the position and velocity. One could now easily say, that the next longitudinal position has to be $s(j' + 1) = s(j) + v(j) \cdot \Delta t$, which is true for a theoretic case. But in real life there are uncertainties and other influences (some of which we will consider), which affect the motion of the vehicle. This uncertainties are modelled with Gaussian distributions where the mean value, e.g., could be $s(j) + v(j) \cdot \Delta t$.
- Dynamic Bayesian networks:
  These are Bayesian networks, which don't relate fundamentally different variables like intelligence and grade, but relate variables over adjacent time steps.

For the prediction of driving behavior Dynamic Bayesian Networks are used, as it is standing to reason that the future position and velocity are dependent on the actual or

preceding position and velocity. One example can be found in [26], where the behavior of the surrounding vehicles was predicted with the Dynamic Bayesian Network in Figure 2.3. In this case the future velocity and lateral displacement were predicted based on the respective preceding values and an indicator signal.

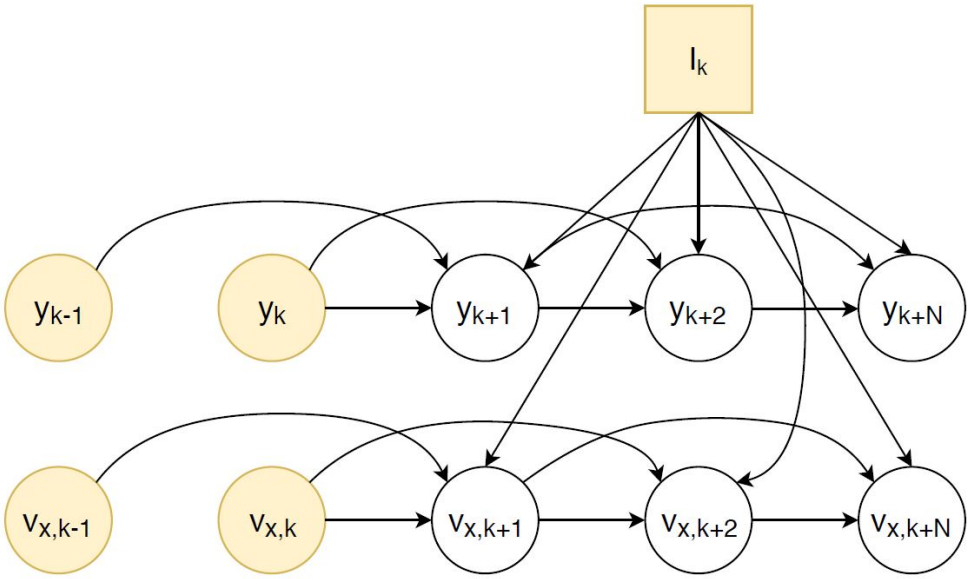

Figure 2.3: Bayesian network from [26] for prediction of future velocity $v_{x,i}$ and lateral displacement y_i. (I_K ... Indicator signal)

2.2.2 Inclusion of topological attributes and limitations

In Figure 2.3 and in previous works like [24, 26, 23] it can be seen, that mainly first principles are utilized as variables in the Bayesian network, in order to predict ones driving behavior. But thinking about experiences of real driving situations on country roads, one can imagine that there are a plenty of influences which affect the driver. Some of these influences are topological and hence only dependent on the actual position $s(j')$ of the road, like the curvature or the slope. In this work, we want to incorporate those topological influences in a Bayesian network, so as to achieve a more realistic prediction of the driving behavior. This results in a structure of the Bayesian network, which can be seen in Figure 2.4.

In the lower part of Figure 2.4, a trivial Dynamic Bayesian network can be seen, where the future velocity and longitudinal position (blue circles) are dependent on their preceding values, just as in Figure 2.3. Additional to this basic structure, in the upper part of the figure the incorporation of the topological influences is depicted. The yellow square with the label Σ stands for the additional interchangeable information which is used. Throughout this work we will construct several experiments and combinations of additional topological variables and informations, in order to determine their influence on the prediction.

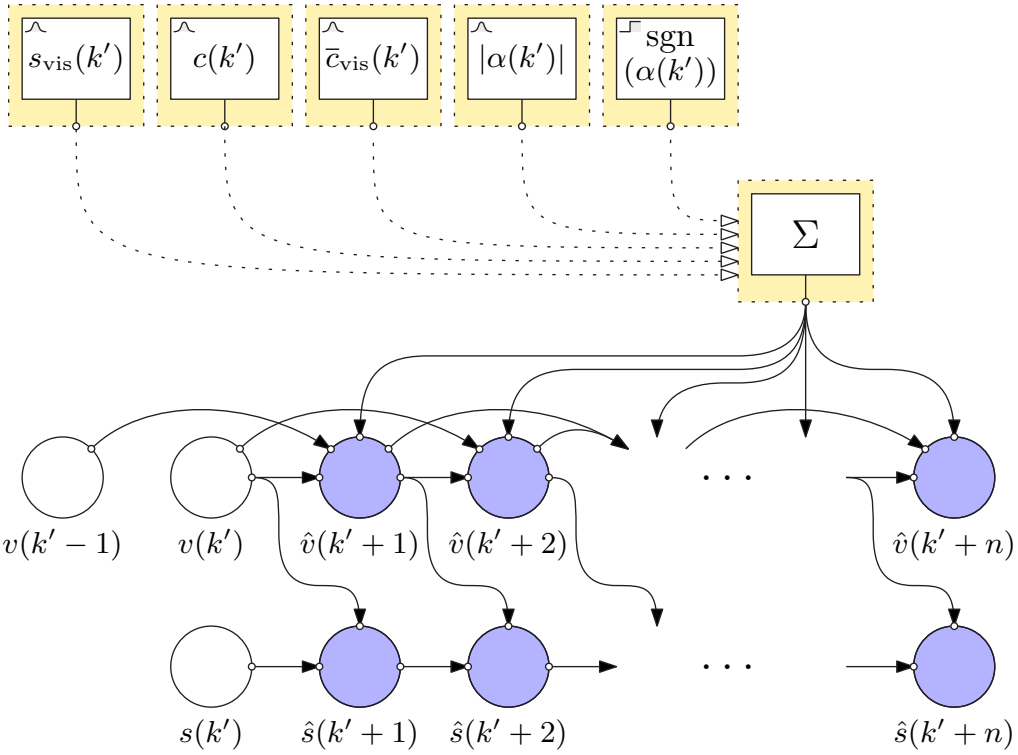

Figure 2.4: Basic structure of the Bayesian network for prediction, where the additionally used information (Σ) is interchangeable. Predicted values are displayed in blue and marked with a hat.

The topological attributes, which will be used in this thesis, are:

s_{vis}	...	Visible distance
c	...	Curvature
$\bar{c}_{vis}$	...	Maximum curvature within the visible distance
$\lvert\alpha\rvert$	...	Absolute value of the slope
$\text{sign}(\alpha)$	...	Sign of the slope

As the presence of a potential preceding vehicle was not recorded in the data, the neglect of such – especially in terms of visible distance $s_{\text{vis}}(k')$ – poses a limitation. On the other hand, it somehow also robustifies our method, as this knowledge cannot be expected to be given for all surrounding vehicles in reality.

If this circumstance had been considered and the prediction models had been identified only for vehicles without preceding vehicles – or with the actual visible distance – a more accurate prediction performance than that achieved in this work could be expected.

2.3 Concept of deterministic limits

Normal distributions are assumed within this work, which is theoretically to be considered as a limitation, since they feature infinite expansion, which is unrealistic for most physical quantities. However, in practice this is not relevant here, since we deduce deterministic limits from the parameters of the distribution. This approach is depicted in Figure 2.5:

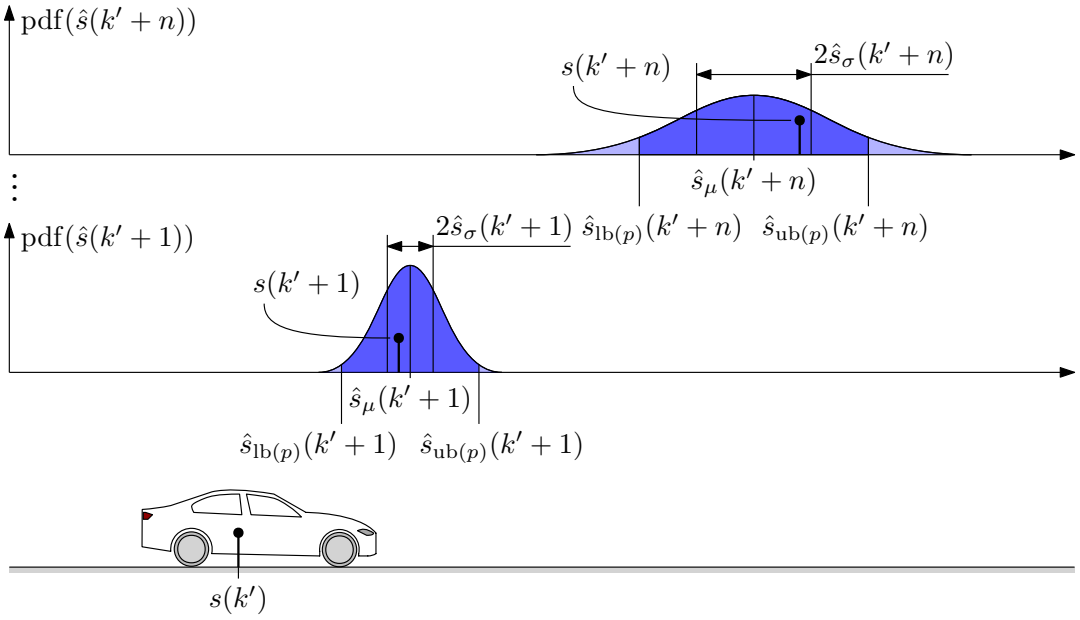

Figure 2.5: Concept of the prediction for n steps in the future with the respective parameters of the predicted distribution ($\hat{s}_\mu(\cdot)$, $\hat{s}_\sigma(\cdot)$) as well as predicted lower and upper boundary ($\hat{s}_{\mathrm{lb}(p)}(\cdot)$, $\hat{s}_{\mathrm{ub}(p)}(\cdot)$) and actual value $s(\cdot)$.

Assuming the prediction of the parameters of a normal distribution at an arbitrary temporal time instant $k'+n$ – with $\hat{s}_\mu(k'+n)$ as the spatial mean and $\hat{s}_\sigma(k'+n)$ as the spatial standard deviation – the lower and upper spatial border $\hat{s}_{\mathrm{lb}(p)}(k'+n)$ and $\hat{s}_{\mathrm{ub}(p)}(k'+n)$ which enclose the value with a probability p can be derived using the z-factor z_f^p related to the respective probability-level.

$$\hat{s}_{\mathrm{lb}(p)}(k'+n) = \hat{s}_\mu(k'+n) - z_f^p \cdot \hat{s}_\sigma(k'+n) \tag{2.1}$$

$$\hat{s}_{\mathrm{ub}(p)}(k'+n) = \hat{s}_\mu(k'+n) + z_f^p \cdot \hat{s}_\sigma(k'+n) \tag{2.2}$$

This way, the results of the prediction can be directly converted to a deterministic spatial interval based on which error-level $(1-p)$ one is willing to accept.

All previously discussed concepts can be seen jointly in Figure 2.6. The prediction is done based on first principles – previous position and velocities – and on topological attributes of the road like curvature, slope and visible distance of the driver. With the above explained approach we can predict a deterministic possible range of the

longitudinal position of the vehicle, with the lower bound $\hat{s}_{\mathrm{lb}(p)}$ and upper bound $\hat{s}_{\mathrm{ub}(p)}$.

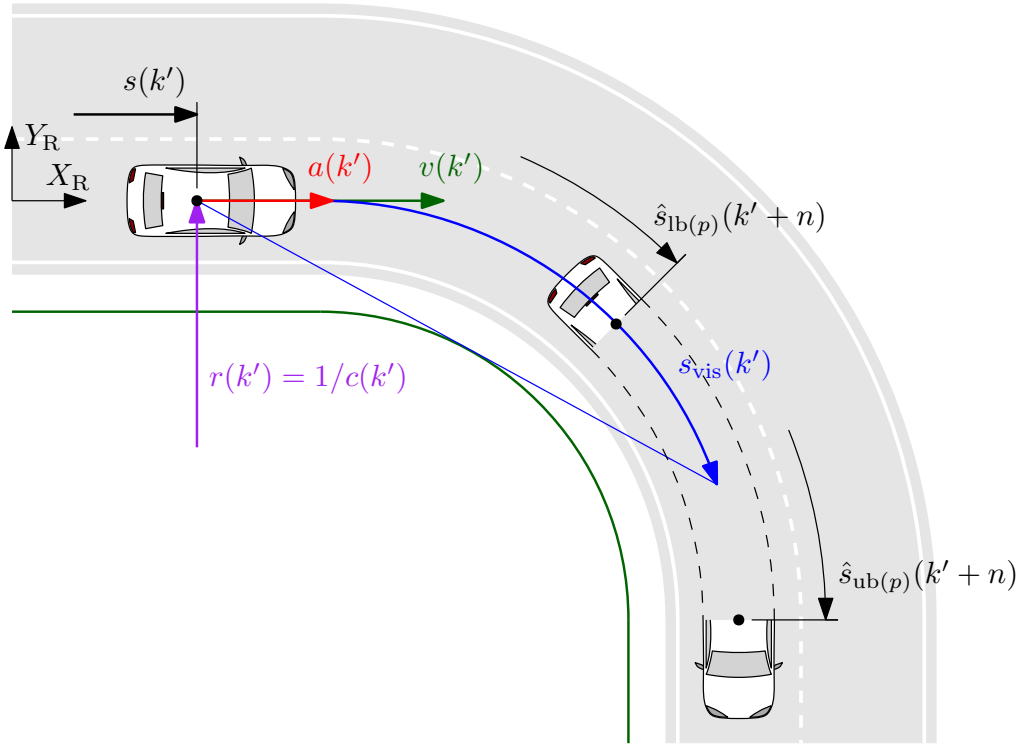

Figure 2.6: Concept of the prediction method including topological influence factors. It is assumed that the visible distance is limited only by the environment and not by a potential preceding vehicle.

2.4 Implementation in Matlab

For the computationally quite complex process of training a Bayesian Network, sampling from the net and validating the model, basically the Bayesian Net Toolbox for Matlab by Murphy et al. [27] was used. This toolbox supports many kinds of nodes (probability distributions), exact and approximate inference, parameter and structure learning, and static and dynamic models. In order to increase the usability, the toolbox was extended at the institute in the last years, resulting in a small number of needed functions. The use of these functions will be explained shortly in the following for a model which incorporates all topological attributes, in order to give an overview of the implementation.

2.4.1 Identification

```
%% Define net structure
% Define all variables in the Bayesian net
variables(1) = bayes.defineVariable('s',     's',    0, @gaussian_CPD);
variables(2) = bayes.defineVariable('v',     'v',    0, @gaussian_CPD);
variables(3) = bayes.defineVariable('vis',   'vis',  0, @gaussian_CPD);
variables(4) = bayes.defineVariable('curv',   'curv', 0, @gaussian_CPD);
variables(5) = bayes.defineVariable('viscurv',  'viscurv', 0, @gaussian_CPD);
```

```
variables(6) = bayes.defineVariable('Aalpha',  'Aalpha', 0, @gaussian_CPD);
variables(7) = bayes.defineVariable('Salpha',  'Salpha', 2, @tabular_CPD);

% Define all outputs and their dependencies
outputs(1) = bayes.defineOutput('s(k)', {'s(k-1)', 'v(k-1)'},   nPH);
outputs(2) = bayes.defineOutput('v(k)', {'v(k-1)', 'v(k-2)', 'vis(0)','curv(0)'
    ,'viscurv(0)','Aalpha(0)','Salpha(0)'}, nPH);
structure = bayes.defineNetStructure(variables, outputs);
```

Listing 2.1: Defining the structure of the Bayesian Network

In Listing 2.1 it can be seen, how the structure of the Bayesian network can be defined. First, with the function *defineVariable* one is able to include a variable to the network and define its name and its assumed conditional probability distribution. In this case, all variables are assumed to be distributed normally, except the sign of the slope, which can only adopt 2 values. The function *defineOutput* creates the connections within the net, as the name and the dependencies of the output variables are defined. Furthermore, the prediction horizon n_{PH} is to define. After defining the variables and outputs, the structure of the Bayesian net is to define with the function *defineNetStructure*. For the concrete case of $\mathcal{P}_{\Sigma}$, this leads to the Bayesian net in Figure 2.7, generated with a plot function of the toolbox. As it can be seen, a prediction horizon of 20 is chosen, which corresponds to a time span of 10 seconds.

```
%% Train model
% Segmentize training data
lenSegment = bayes.getSegmentLength(structure.nodes);
segments = bayes.segmentizeDatasets(data, lenSegment);
% Train
model = bayes.trainModel(structure, segments);
```

Listing 2.2: Defining the structure of the Bayesian Network

Listing 2.2 shows the process of training the Bayesian net. First the training data is segmented with the function *segmentizeDatasets* into segments with the length of the prediction horizon. Afterwards, these segments are used to train the net with *trainModel*, which returns the trained Bayesian net model.

2.4.2 Validation

The model can be sampled with the function *getSamples*, as it can be seen in Listing 2.3. The result are *NSamp* predicted positions and velocities over the prediction horizon, which can be converted into a mean course position respectively velocity and a standard deviation.

This is done for every data point in the validation dataset, resulting in a $1 \times n_{PH} = 1 \times 20$ predicted vector for every validation data point. Afterwards, the performance measures from Section 4.2 are calculated for every prediction step.

```
%% Sample model
input = bayes.getInput(model, val_input, val_ind);
samples  = bayes.getSamples(models.m{iExp}, input, NSamp);
```

Listing 2.3: Defining the structure of the Bayesian Network

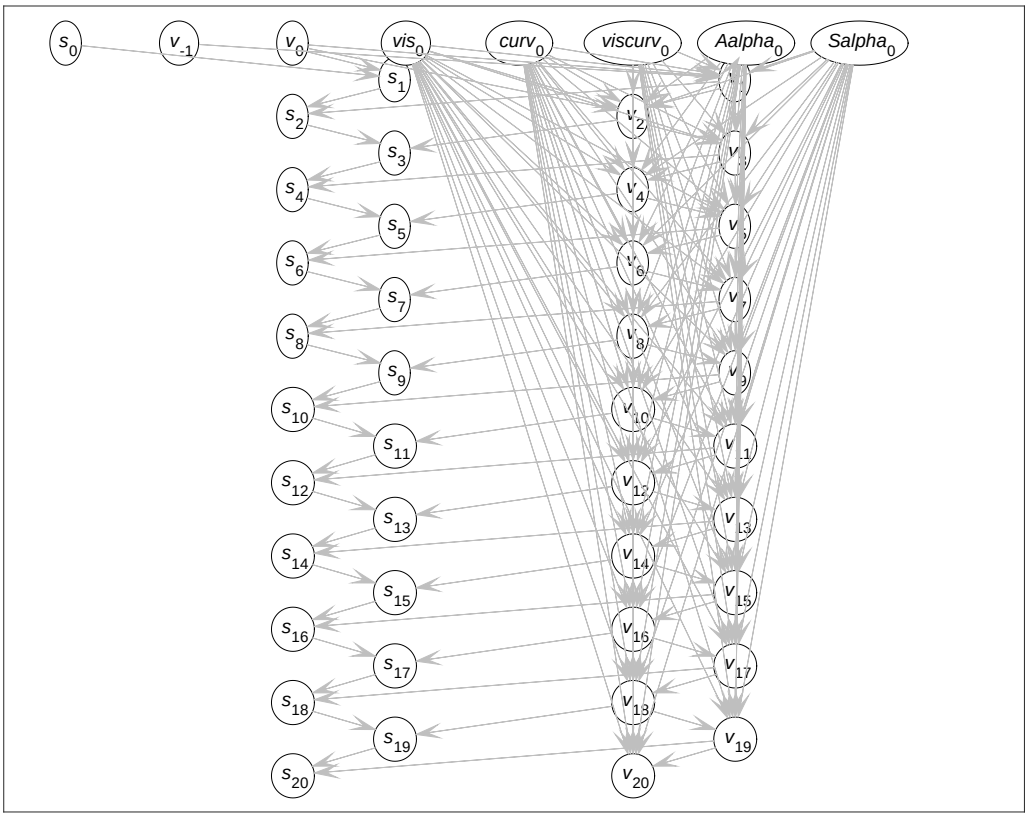

Figure 2.7: Structure of the Bayesian net for $\mathcal{P}_\Sigma$

Chapter 3

Collection and analysis of data

In this chapter, the origin and the methodology of the data collection will be discussed. Furthermore, the given data will be analyzed mainly visually through histograms, in order to get an insight into the data, which will be used for identification and validation in the following chapters.

3.1 Data Collection and Processing

To develop and especially also to validate the performance of prediction methods, data is required. In this case, the data is limited to trajectories of a measurement vehicle without information concerning the surrounding traffic. Although the surrounding traffic is not the main focus in this work, this poses a limitation as the influence of other vehicles cannot be modelled, leading to an unknown bias of the measurements.

3.1.1 Data Collection

The measurements were taken using a BMW F31 equipped with different sensors, of which the DGPS-sensor is relevant for the current work. This sensor measures the current position of the vehicle with the help of the DPGS (Differential Global Positioning Service), which is a improved version of the well-known GPS, whereby the measured position is corrected with the help of fixed ground-based reference stations that have a fixed known position.

On a country road (L1501 between Linz and Altenberg, AT) various drives were performed to measure trajectories of the vehicle. The initial purpose of these test drives was to create models for eco-driving control [31]. The drivers were not aware that their behavior would be analyzed also in terms of safety, leading to non-biased and natural behavior. In total, 39 drives were performed. For the current work, the datasets were truncated to 5 km, in order to have a uniform route length for all drives and to cut off the starting and ending part of the route, which are more similar to urban traffic than to country roads. The resulting truncated route is portrayed in Figure 3.1.

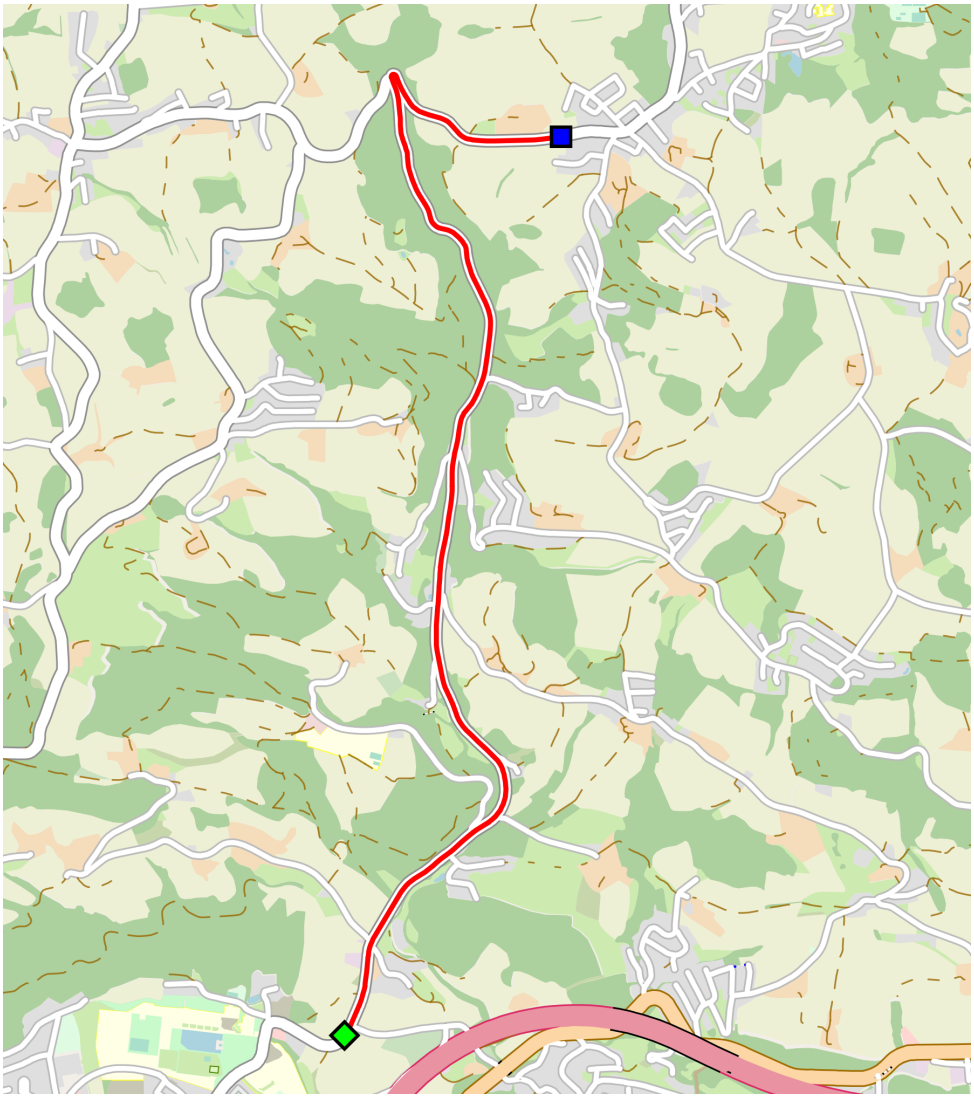

Figure 3.1: Route (Altenberger Straße - L1501 [32]) with a total length of 5 km where the test drives were performed. The green diamond represents the start in upward direction, while the blue box displays the end of the route – and vice versa.

3.1.2 Data Processing

Additional to the measurement of the trajectories of the test vehicle, topological attributes of the route in Figure 3.1 were determined. Those are

- slope α:
  The slope can be determined by taking 2 consecutive measurements and applying the trigonometric laws:

$$\tan\alpha = \frac{\Delta Z_R}{{\Delta X_R}^2 + {\Delta Y_R}^2} \text{ or } \sin\alpha = \frac{\Delta Z_R}{{\Delta X_R}^2 + {\Delta Y_R}^2 + {\Delta Z_R}^2}$$

- curvature c:
  The determination of the curvature is based on the assumption, that the whole route can be expressed using consecutive arcs with a well-defined radius of the

center-line, see Figure 2.1. The curvature is then calculated as the reciprocal value of the radius.

- visible distance s_{vis}: In Figure 3.2, the concept of the estimation of the visible distance is depicted, which was first introduced in [29] and expanded by adding a consideration of the road grade in [5].

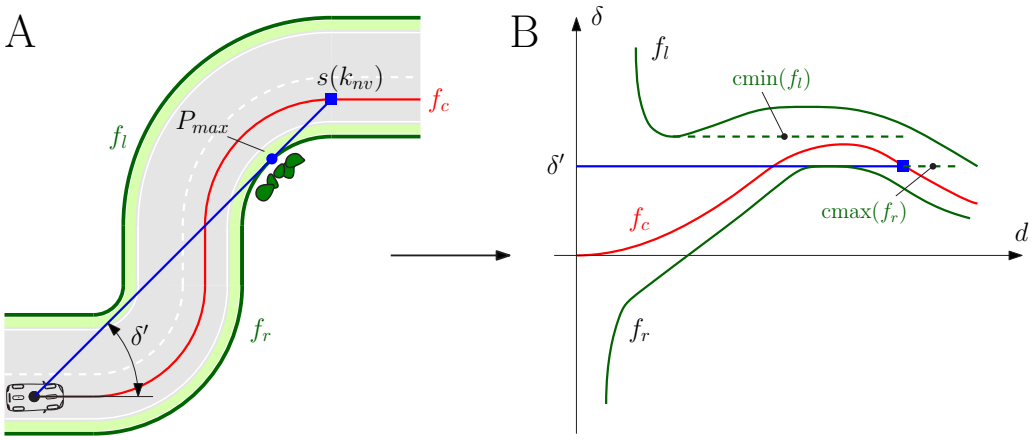

Figure 3.2: Concept of the estimation of the visible distance s_{vis} (from [29]) A ... representation in Cartesian space, B ... transformed representation in angular space

The idea of the estimation of the visible distance is to transform the Cartesian representation of the road (Figure 3.2-A) and its lanes and margins to an angular representation (Figure 3.2-B), which is defined by the relative angle to the longitudinal axis of the vehicle and the distance to the vehicle. For the estimation, the left road margin f_l and right road margin f_r are transformed to the angular space (darkgreen functions), as well as the center of the right lane (red function). In this new representation, the determination of the first non-visible position or index k_{nv} is possible by simply evaluating

$$k_{nv} = \arg\min_k k \tag{3.1}$$

$$\text{s.t.} \quad \text{cmax}(f_r(k)) \geq f_c(k) \mid \text{cmin}(f_l(k)) \leq f_c(k) \tag{3.2}$$

This means, that the determination of the visible distance is simplified to the determination of a cumulated minimum cmin(f_l), cumulated maximum cmax(f_r) and the intersection of the center of the right lane f_c with cmin(f_l) or cmax(f_r). In (Figure 3.2-B), the visible distance is depicted in blue at the point where f_c is lower than cmax(f_r) and so the constraints of the above stated problem is fulfilled. If the car in the figure would be some meters behind its depicted position, the transformed right lane f_c would first reach the cumulated minimum cmin(f_l), and so the visible distance would be much lower.

As the visible distance is not only dependent on the curvature of the road, and so the left and right margins of the road, but also on the height profile of the road, this concept was extended in [5]. The expansion with the height profile is

quite similar to the above explained concept, with the difference that the visible distance is only dependent on the height profile of the road itself. The first non-visible index is then the one after the cumulated maximum of the height profile.

The resulting estimated visible distance is then the minimum of the two separately estimated visible distances.

3.2 Data Analysis

After collecting and processing the necessary data, the data needs to be analysed, so that it can be used properly for identification and validation of the prediction models.

3.2.1 Analysis concerning the topological attributes

In order to get an imagination of their distribution along the route, the magnitudes of the three topological attributes slope α, curvature c and visible distance s_{vis} are plotted along the route in Figure 3.3. Green coloring stands for a small slope, a low curvature or a high visible distance, while red coloring stands for the opposite. Furthermore, it is to say, that for this depiction only the "upward" direction was considered. It can be seen that the slope is relatively high along the whole route, with only few road segments with no or negative slope, which stands for downhill segments. The curvature along the whole route is mainly low, as the green coloring suggests. But, there are some exceptions where the curvature is relatively high, mainly at the last third of the road, where even a almost 180-degree-curve occurs. As explained above, the visible distance is mainly determined by the slope or height profile and the curvature of the road. This fact reflects also in Figure 3.3-C. In segments where the curvature and the slope are low, the visible distance is relatively high, while it becomes relatively low if at least one of the two topological attributes is high.

Within the dataset, both the route and also the vehicles show a wide range in terms of the single attributes, which are visualized in Figure 3.4, where in the upper plots (A, C, E, G) both directions are shown separately, with "upwards" meaning starting from the green diamond towards the blue box in Figure 3.1. In the lower plots (B, D, F, H) the data is shown for both directions combined. It can be seen, that the velocity has an approximate form of a normal distribution with a mean of about $18\,\mathrm{m/s} \approx 65\,\mathrm{km/h}$. Furthermore, an interesting fact is that the velocity distributions for both directions are almost the same, which was not to expect, as the downwards direction consists mainly of downhill segments, where one could tend to drive at slightly higher velocities. The distribution of the visible distance has a range of 300 metres and its mean at approximately 100 metres, which shows that rather lower visible distances occur along the route. As mentioned above, the curvature along the route is mainly low, which is confirmed in Figure 3.4-F. The mean of the combined distribution of the curvature is almost exactly at $0\,\mathrm{m}^{-1}$, but the full range of the occuring curvatures accounts for about $0.04\,\mathrm{m}^{-1}$. For a better imagination of this figure, the curvature of the curved sides of a standardized running track is about $0.025\,\mathrm{m}^{-1}$. In contrast to the curvature,

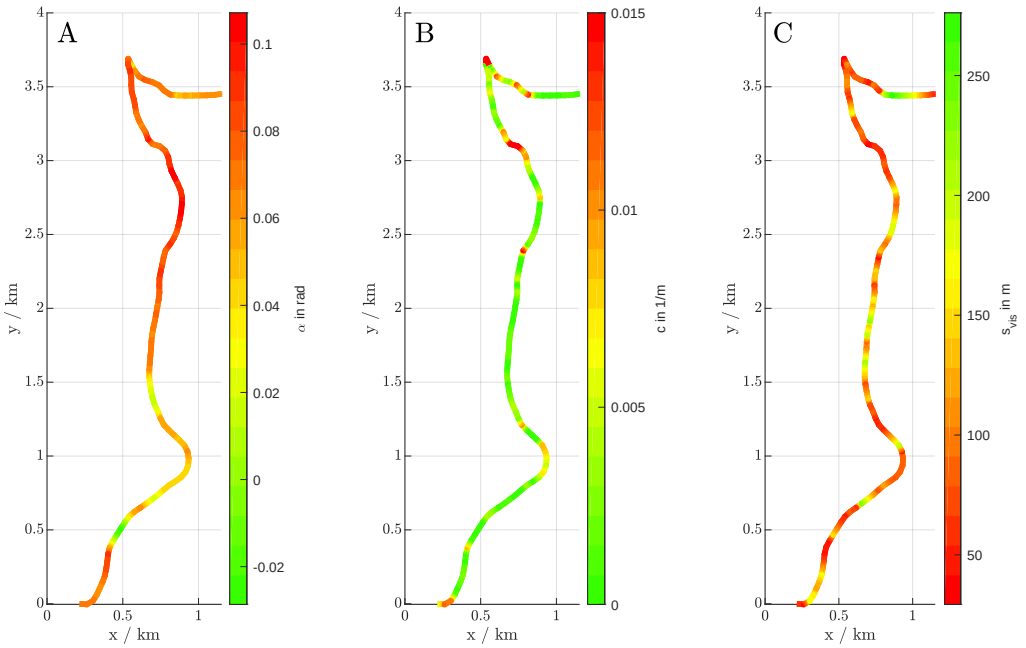

Figure 3.3: Heatmaps of the topological attributes, A) slope α, B) curvature c, C) visible distance s_{vis}

the slope is distributed around a mean of about 0.7 rad, which can be converted to the specification of slope used in road traffic of approximately 8.9%.

The wide ranges of the single attributes are also illustrated in Figure 3.5, where the slope, the curvature and the visible distance are plotted in relation to the driven velocity. There are no significant correlations to see, as the driven velocity is not affected by the 3 attributes separately, but by all 3 at the same time. Nevertheless, some intuitive expectations can also be found in these 2D-plots. For example, there are no data points at very low visible distances and high velocities. Furthermore, the higher the curvature, the lower was the driven velocity. The slope is very wide distributed in relation to the velocity, the only behavior that can be observed is that the range of the driven velocity is smaller in segments with no or very low slope. The majority of the data points or cases is at a driven velocity between 55 km/h and 70 km/h, a slope of $\pm 0.05 - 0.1$ rad, almost no curvature and a visible distance between 50 and 100 m.

Besides the visual analysis in the form of plots, the data is also analysed based on its statistical features in Table 3.1. The statistical features are the minimum and maximum occurring value in the dataset, and the 25-, 50- and 75-quantile. Quantiles are values that partition a finite set of values into subsets of almost equal sizes. For example, the quantile $\boldsymbol{Q_{25}}$ stands for the value, which is greater than 25% of the values in the dataset. In other words, 25% of the dataset's values are lower than $\boldsymbol{Q_{25}}$. According to this notation, the minimum could also be called $\boldsymbol{Q_0}$ and the maximum $\boldsymbol{Q_{100}}$, while the mean value is $\boldsymbol{Q_{50}}$. In Table 3.1, the findings of the visual observation

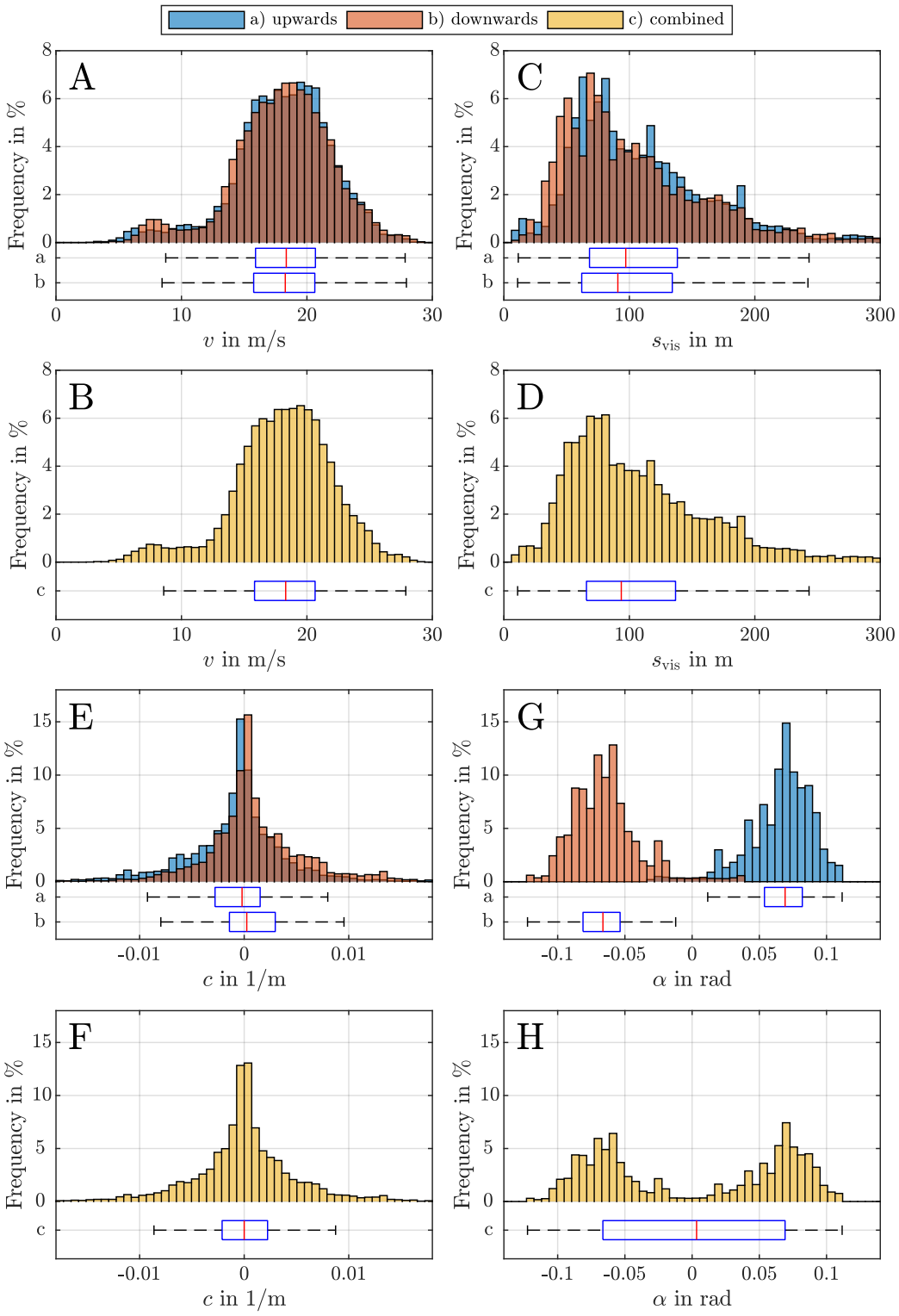

Figure 3.4: Histograms of the velocity (A-B) and the route attributes visible distance s_{vis} (C-D), curvature c (E-F) and slope α (G-H) in upwards, downwards and combined-direction each

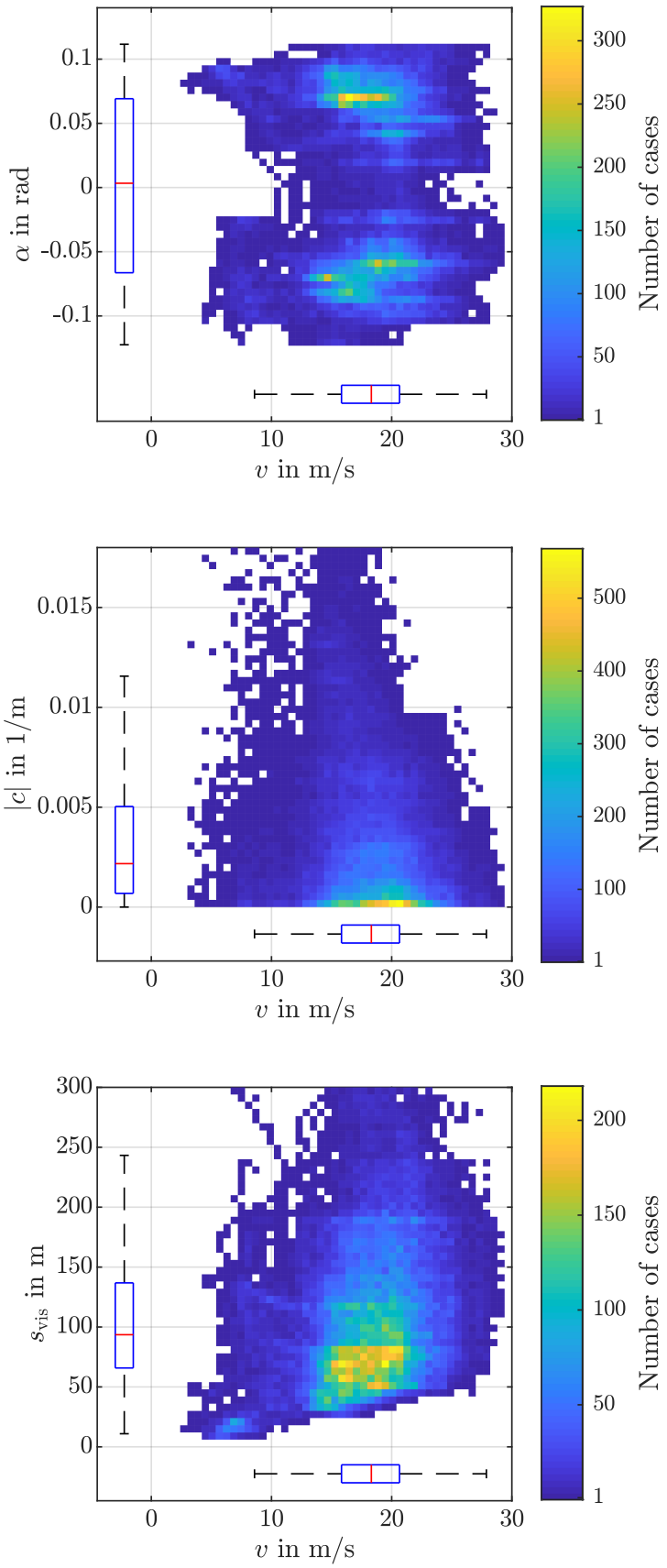

Figure 3.5: Histogram of the visible distance s_{vis}, related to the driven velocity

are confirmed. The distribution of the driven velocity is almost symmetric, while the distribution of the visible distance has a trend to lower values. $\boldsymbol{Q_{75}}$ accounts for about 137 m, while the maximum value is 300 m. This means that the width of the range of the first 75% of the data is almost equal to the width of the range of the last 25% of the data. Another unique behavior occurs at the curvature, where the 3 quantiles are equal zero. Of course, they are not exactly equal to zero, but differ in some digit behind the comma. Nevertheless, this circumstance shows that the curvature is mainly relatively low along the route. The slope is distributed symmetrically, as the 2 distributions of the upward and downward direction mainly differ by their sign.

Table 3.1: Statistical features of the recorded data.

Attribute		**min**	$\boldsymbol{Q_{25}}$	$\boldsymbol{Q_{50}}$	$\boldsymbol{Q_{75}}$	**max**
v	in m/s	2.83	15.83	18.31	20.65	29.35
s_{vis}	in m	10.93	65.81	93.57	136.82	300.00
c	in 1/m	-0.16	0.00	0.00	0.00	0.16
α	in rad	-0.12	-0.07	0.00	0.07	0.11

3.2.2 Analysis by a self-made test drive

In order to get a better feeling of the route and the various situations which can wait for a real driver, an additional test-drive was executed on the route and filmed with a conventional mobile phone camera. The most significant spots or situations with the expectation of the influence on a potential driver will be discussed in the following:

- Spots with high and low slope α:
  In this case "low slope" includes also downhill segments, which is shown in Figure 3.7. From an heuristic, subjective point of view, the slope may affect the drivers behavior in terms of velocity and fuel consumption. On the one hand, if the slope is high, acceleration and therefore fuel consumption is needed to at least maintain at constant velocity. On the other hand, if the slope is low, no acceleration would be needed. However, this depends on the type of driver.

Figure 3.6: Spot with highest slope $\alpha_{max} = 0.112\,\text{rad} = 6.39\,^\circ \mathrel{\widehat{=}} 14.2\,\%$

Figure 3.7: Spot with lowest slope $\alpha_{min} = -0.0289\,\text{rad} = 1.66\,^\circ \mathrel{\widehat{=}} -3.68\,\%$

- Spots with high and low curvature c:
  At spots with high curvature like in Figure 3.8, the driver has to adjust the velocity in order to pass the curve safely.

Figure 3.8: Spot with highest curvature $c_{max} = 0.156\,\mathrm{m}^{-1}$

Figure 3.9: Spot with lowest curvature $c_{min} \approx 0\,\mathrm{m}^{-1}$

- Spots with high and low visible distance s_{vis}:
  The same as in Figure 3.8 stands for spots with low visible distance, whereby Figure 3.8 and Figure 3.10 show the same spot. The driver has to adjust the velocity, as he has no appropriate overview of the situation.

Figure 3.10: Spot with highest visible distance $s_{vis,max} = 301.4\,\mathrm{m}$

Figure 3.11: Spot with lowest visible distance $s_{vis,min} = 10.9\,\mathrm{m}$

With these observations of the topological attributes, the question arises how the consideration of certain attributes influences the quality of a prediction model. This question is addressed by the approach presented in Chapter 4.

Chapter 4

Determination of the influence of the topological attributes

In this chapter the influence of the already introduced topological attributes will be analyzed. For this purpose, models with different combinations of considered topological attributes will be trained and compared with the help of performance measures, which will be introduced at the beginning of the chapter.

4.1 Experimental setup

As explained in Section 2.2.2, five topological attributes are deployed for the consideration in the prediction models: visible distance s_{vis}, curvature c, maximum curvature within the visible distance $\bar{c}_{vis}$, absolute value of the slope $|\alpha|$ and sign of the slope $\text{sign}(\alpha)$. In order to determine their single influence on the quality of the prediction models, as well as the influence of different combinations of the topological attributes, the Bayesian nets for this observation will be trained with all possible combinations of the above mentioned variables. With this approach it is also possible to determine how the inclusion or extraction of certain topological attributes affects the performance.

The number of possible combinations can be calculated with the binomial coefficient

$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$$

$$n_{comb} = \binom{1}{5} + \binom{2}{5} + \binom{3}{5} + \binom{4}{5} + \binom{5}{5} = 31 \text{ combinations}$$

Therefore, including the basic method with only first principles (position and velocity, without topological attributes), this results in a total of 32 different models. These 32 models will be analysed step by step: first, the models are compared based on their number of considered topological attributes in the prediction model. Subsequently, the models are compared based on the particular topological attributes which are considered in the certain models. As a next step, the influence of including a particular

variable will be observed. Finally, this observation will be done for short and long prediction horizons. The present database with a total of 39 drives is divided in two parts, whereby the first 20 drives are used for identification and the second 19 drives are used for validation of the prediction models.

4.2 Performance measures

In order to quantify and measure the prediction performance, two aspects are analyzed, on the one hand the accuracy of prediction and on the other hand the aspect, how conservative the prediction is. This is especially relevant in cases – such as the present one – where the result of the prediction is an interval and not a precise value. In case the predicted interval is overestimated, the first criterion would always evaluate very good, while yielding a practically not useful big interval. This problem is dealt with by additionally comparing a measure for the width of the interval in the second criterion. $n = [1, \ldots, 20]$ prediction steps are analyzed, whereby the sampling time is chosen with $T_S = 0.5\,\mathrm{s}$.

4.2.1 Prediction accuracy

As the primary objective of the validation is to find how well one can trust in the predicted interval, a measure to rate the accuracy of the interval, similar to the $P_{15/15}$ method from [33], is used. At first sight the term "accuracy" could sound confusing in combination with probabilistic prediction, but the accuracy in this case means merely to determine if a measured sample lies in the predicted interval.

$$P_{\mathrm{acc}}^{(\cdot)}(n,p) = \frac{100}{J}\sum_{j=1}^{J} p_{\mathrm{acc}}^{(\cdot)}(n,p) \tag{4.1}$$

$$p_{\mathrm{acc}}^{(\cdot)}(n,p) = \begin{cases} 1 \;\text{ if } \begin{array}{l} \hat{s}_{\mathrm{lb}(p)}^{(\cdot)}(k'+n) \le s(k'+n) \wedge \\ \hat{s}_{\mathrm{ub}(p)}^{(\cdot)}(k'+n) \ge s(k'+n) \end{array} \\ 0 \;\text{ otherwise} \end{cases} \tag{4.2}$$

High values of $P_{\mathrm{acc}}(n,p)$ indicate a good performance, while low values imply the opposite. An intuitive interpretation is, that for a very high number of validation datasets J and a perfect match of prediction method and actual behavior, the value of $P_{\mathrm{acc}}(n,p)$ should converge towards p. Values $> p$ indicate an underestimation of the behavior-variance, values $< p$ indicate the opposite. For all tests, one standard deviation was set as the bidirectional limit ($z_f^{68.26\,\%} = 1$), which poses a limitation as only a certain range of the probability distribution is observed. Nevertheless, with the assumption of Gaussian distributions the results of this approach have general

expressive power. In order to have a better comparison to the basic method $\mathcal{P}_0$, the following adaption of the accuracy will be used in some cases:

$$P^{\%}_{\text{acc}}(n,p) = \frac{P_{\text{acc}}(n,p) - P^{\mathcal{P}_0}_{\text{acc}}(n,p)}{P^{\mathcal{P}_0}_{\text{acc}}(n,p)} \tag{4.3}$$

yielding the absolute percentage deviation of the predicted accuracy from the accuracy of the basic method.

4.2.2 Prediction conservativeness

Consequently, a measure for the conservativeness of the predicted interval is calculated by comparing the predicted standard deviations to the one from the basic method $\mathcal{P}_0$.

$$P^{(\cdot)}_{\text{con}}(n) = \frac{100}{J} \sum_{j=1}^{J} \frac{\hat{s}^{(\cdot)}_{\sigma}(k'+n) - \hat{s}^{\mathcal{P}_0}_{\sigma}(k'+n)}{\hat{s}^{\mathcal{P}_0}_{\sigma}(k'+n)} \tag{4.4}$$

This reflects, how additional information influences the confidence measure of the Bayesian network in the predicted distribution, which is represented by the standard deviation.

4.3 Comparison based on number of considered attributes

In a first step, these 32 prediction models are compared based on the performance measures, as it could be seen in the previous section. The results of the validation are depicted in Figure 4.1 – 4.4.

There it can be seen in general, that the consideration of more topological attributes leads to higher deviations in the prediction accuracy between the respective models. This means, that some models perform better than the ones before, but there are also models which perform worse regarding accuracy. The main difference is, that the measure for conservativeness P_{Con} decreases, which means that the predicted intervals get smaller or tighter when more attributes are considered. Especially when 4 or more attributes are considered, the predicted intervals of all models are up to 3% tighter than the interval of the basic method, as it can be seen in Figure 4.4. Therefore, the main learning of this approach is, that by adding and considering more topological attributes in the Bayesian net, a smaller or tighter future spatial area of the vehicle can be predicted, while the accuracy remains almost constant.

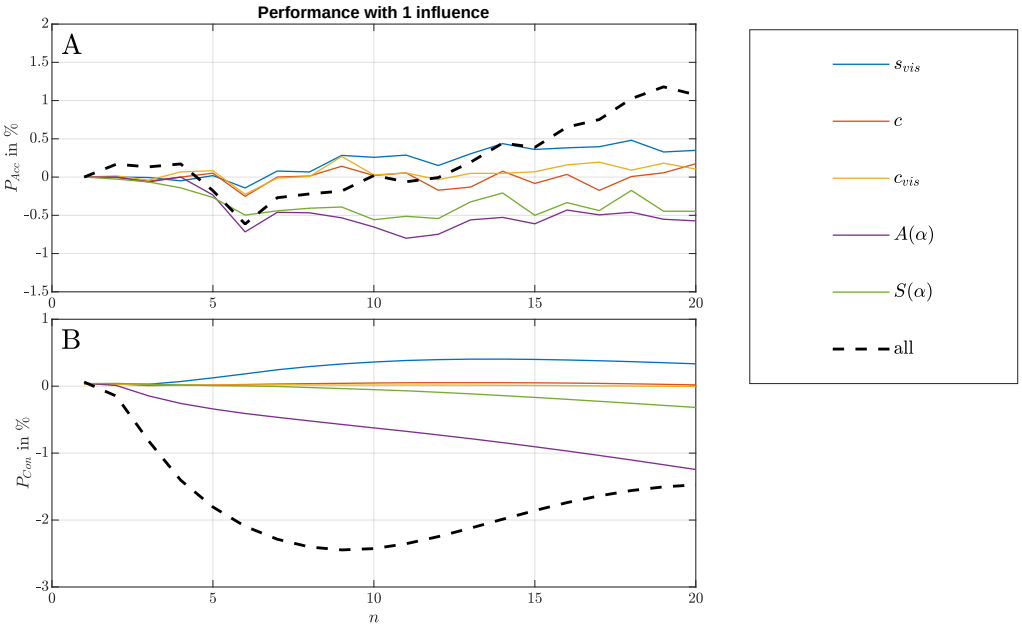

Figure 4.1: Performance measures of the approaches based on only one topological attribute (no combinations)

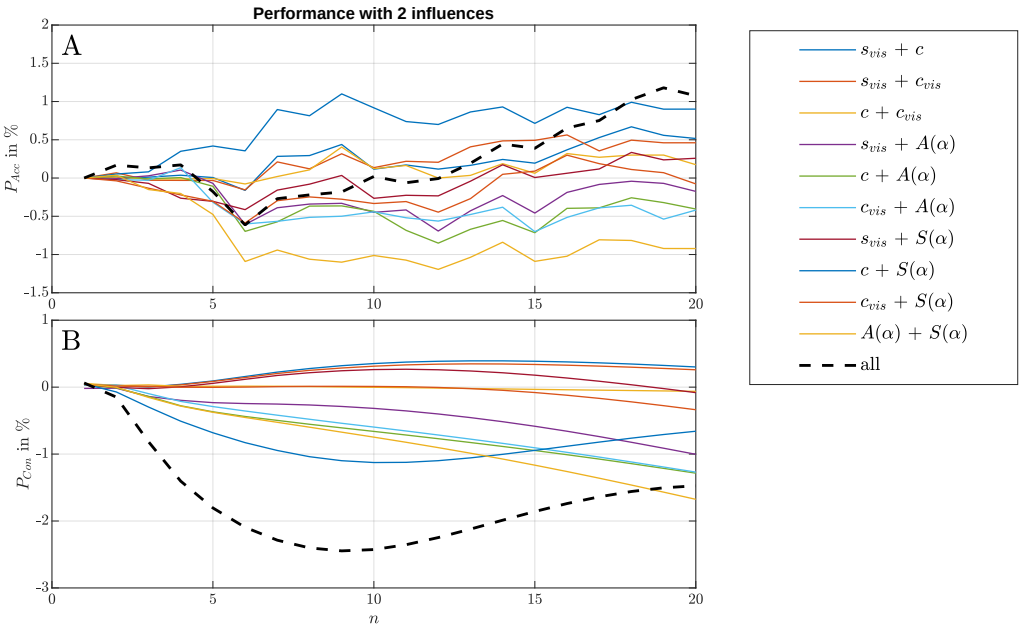

Figure 4.2: Performance measures of the approaches based on combinations of 2 topological attributes

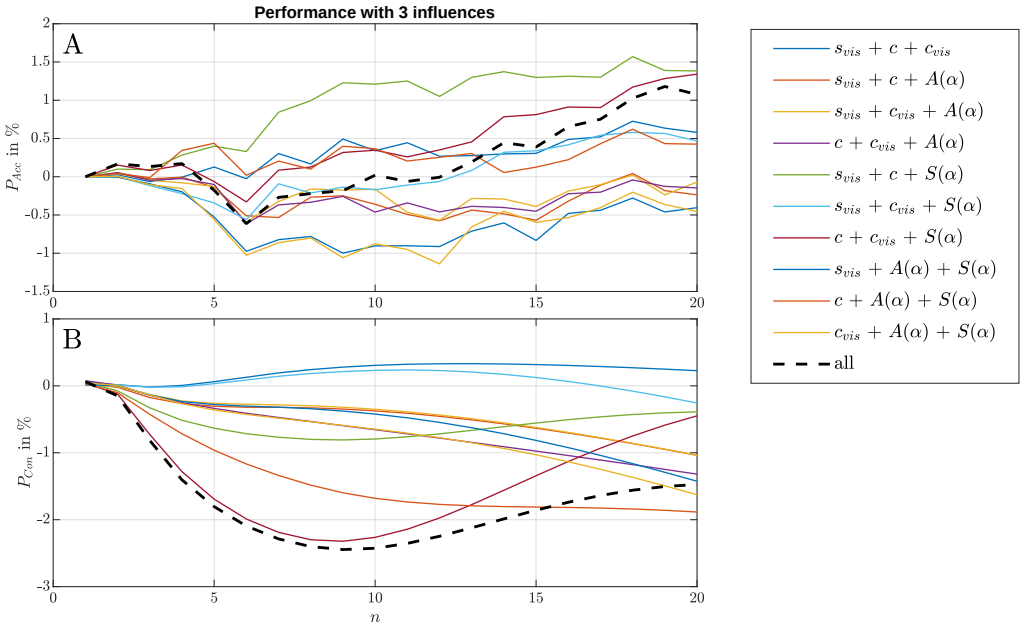

Figure 4.3: Performance measures of the approaches based on combinations of 3 topological attributes

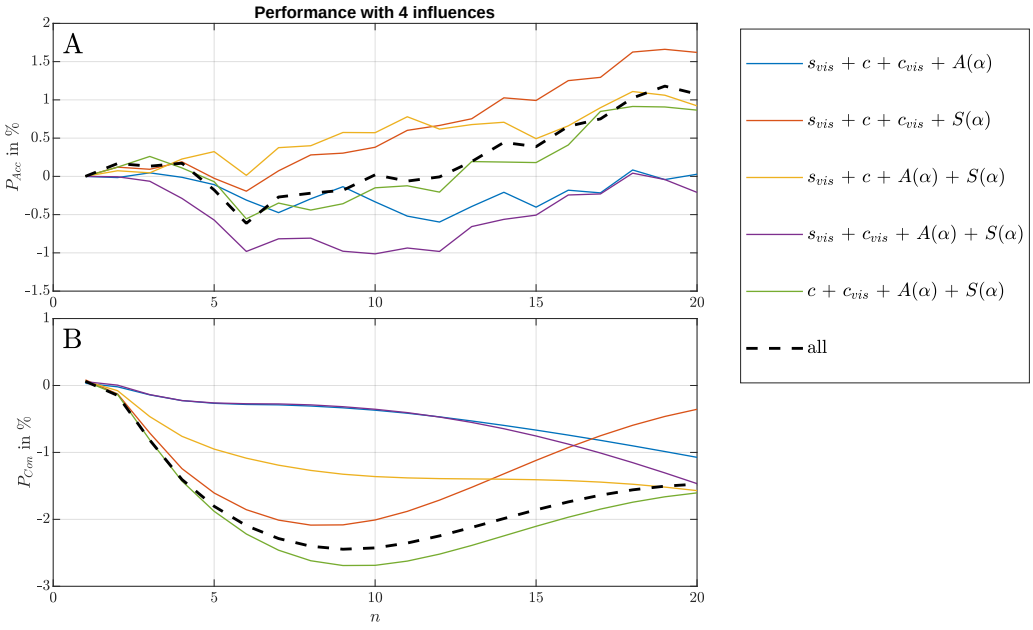

Figure 4.4: Performance measures of the approaches based on combinations of 4 topological attributes

4.4 Comparison based on the considered topological attributes

As yet, it was shown that the interval is tightened by adding more topological attributes into consideration. This could be seen in Figure 4.1 – 4.4 as only the change of the whole curve set was observed. But it is hardly possible to make out which of the 5 or 10 curves is the best or to compare two specific curves. In order to assess and compare the performance of the models, another representation of the results will be used. The idea is, that, in terms of accuracy, one model is better than the other if the curve of the prediction accuracy "lies above" the curve of the other model. If one curve lies above the other, the mean value $\overline{P}_{Acc}(n)$ over $n = 1 \dots 20$ of the first mentioned curve has to be higher than the other one. Therefore, the mean value of the prediction accuracy $\overline{P}_{Acc}$ and prediction conservativeness $\overline{P}_{Con}$ each will be used as measures for comparing the models.

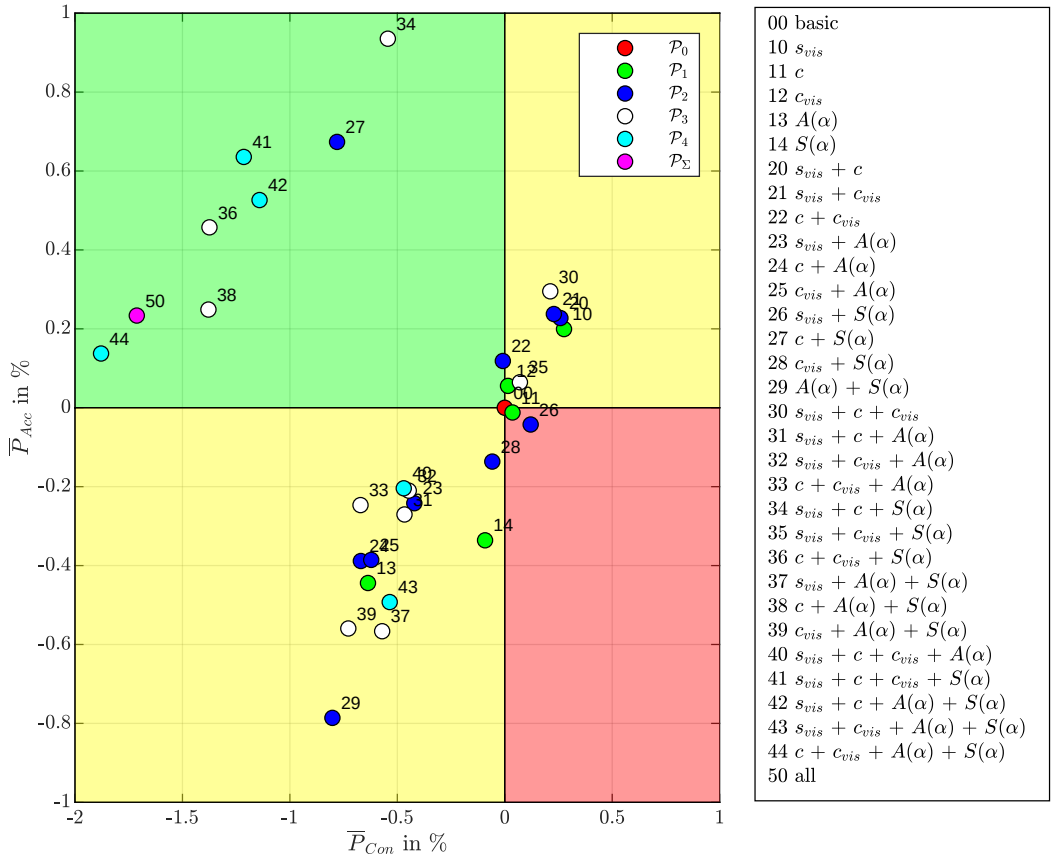

Figure 4.5: 2D-Plot of the performance measures of all possible combinations of the topological attributes.

In Figure 4.5 the results of these adjustments are shown. Every data point in this plot stands for one model or one combination of the topological attributes which were considered in the Bayesian net. The points are enumerated, with the first cypher

signalizing the number of topological attributes that were included in this case and the second cypher representing the enumeration within this group. All data points with the same number of included variables have the same color in the plot and the small legend in the right top corner shows which group has which color. With the large legend on the right side one can see which topological attributes were used at the respective enumerated data point. For example, at data point "37" 3 topological attributes were considered in the Bayesian net, namely visible distance s_{vis} and the magnitude $A(\alpha)$ and the sign $S(\alpha)$ of the slope. This data point is the 7^{th} data point in the group of those points, where 3 attributes were considered, and all of them are colored in white in the plot.

As the performance measures are calculated in relation to those of the basic method (only first principles in the Bayesian net), the data point of the basic method is the reference point in Figure 4.5, where the 4 colored rectangles meet. The colors of the rectangles signalize a assessment of the quality of the prediction performance in relation to the basic method. There are 3 possible situations:

1. Green: The prediction is more accurate and less conservative (tighter prediction interval) than the basic method, and therefore the model performs clearly better than the basic model

$$\overline{P}_{Acc} > 0 \qquad \text{and} \qquad \overline{P}_{Con} < 0$$

2. Yellow: The prediction is either more accurate and more conservative, or less accurate and less conservative. So, looking at one measure, the performance indeed is better, but looking at the other measure makes the prediction seem worse.

$$\overline{P}_{Acc} > 0 \qquad \text{and} \qquad \overline{P}_{Con} > 0$$

$$\text{or}$$

$$\overline{P}_{Acc} < 0 \qquad \text{and} \qquad \overline{P}_{Con} < 0$$

3. Red: The prediction is less accurate and more conservative, which signals clearly a worse behavior in comparison to the basic method

$$\overline{P}_{Acc} < 0 \qquad \text{and} \qquad \overline{P}_{Con} > 0$$

Looking at Figure 4.5, one can see, that, in a way, the most prediction approaches had a better performance than the basic method. The majority of those data points lies in the yellow areas and so the performance is not clearly better, as there are drawbacks. Approximately a quarter of the data points, so 8 out of 31, had a clearly better performance than the basic method and therefore lie in the green area. As one can see, there is no clear cluster regarding the colors, as points of every group lie in the green and in the yellow area. Only the green points, where only one topological attribute is considered, do not have a point in the green area. So it can be said, that considering only one topological attribute yields no advantage in comparison to the

basic method. But now it is to analyze, if other characteristics can be found, which are unique at the points in the green area and therefore guarantee a good prediction performance. Therefore, the topological attributes, which were considered at the data points in the green area in Table 4.1 and those considered at the data points in the other areas in Table 4.2 will be examined:

Table 4.1: Overview of the topological attributes which were included at data points with a good performance

Num	s_{vis}	c	c_{vis}	$A(\alpha)$	$S(\alpha)$
27	0	1	0	0	1
34	1	1	0	0	1
36	0	1	1	0	1
41	1	1	1	0	1
38	0	1	0	1	1
42	1	1	0	1	1
44	0	1	1	1	1
50	1	1	1	1	1
Absolute frequency	4	8	4	4	8
Relative share of subset	50%	**100%**	50%	50%	**100%**
Relative share of whole set	12.5%	25%	12.5%	12.5%	25%

By comparing Table 4.1 and Table 4.2, it can be seen, that regarding the visible distance s_{vis}, the maximum curvature in the visible distance c_{vis} and the magnitude of the slope $A(\alpha)$, no clear statement can be made about their influence on the performance, as all of them appear in approaches with good and bad performances. Also the curvature c and the sign of the slope $S(\alpha)$ do not seem to guarantee a good or bad performance by themselves. But, the main conclusion that can be drawn from Table 4.1 is, that every data point where c and $S(\alpha)$ are included together performs well. As no data point which includes c and $S(\alpha)$ together appears in the red or yellow zone, it can be said that a prediction method performs clearly better than the basic method, if and only if, it includes c and $S(\alpha)$ together in its Bayesian net.

4.5 Influence of the consideration of particular topological attributes

As a next step, the question arises, how the inclusion or extraction of particular topological attributes affects the performance. For example, looking at Table 4.1, the change in performance by switching from point 27 to point 34, which means including the visible distance s_{vis}, is to be determined. Of course, both points are in the green area and thus are approaches with a good performance, but nevertheless, through this inclusion the performance could be even better (nearer to the top left corner in Figure 4.5) or worse (nearer to the bottom right corner). The determination of the

Table 4.2: Overview of the topological attributes which were included at data points with a "bad" performance

Num	s_{vis}	c	c_{vis}	$A(\alpha)$	$S(\alpha)$
10	1	0	0	0	0
11	0	1	0	0	0
20	1	1	0	0	0
12	0	0	1	0	0
21	1	0	1	0	0
22	0	1	1	0	0
30	1	1	1	0	0
13	0	0	0	1	0
23	1	0	0	1	0
24	0	1	0	1	0
31	1	1	0	1	0
25	0	0	1	1	0
32	1	0	1	1	0
33	0	1	1	1	0
40	1	1	1	1	0
14	0	0	0	0	1
26	1	0	0	0	1
28	0	0	1	0	1
35	1	0	1	0	1
29	0	0	0	1	1
37	1	0	0	1	1
39	0	0	1	1	1
43	1	0	1	1	1
Absolute frequency	12	8	12	12	8
Relative share of subset	50%	33%	50%	50%	30%
Relative share of whole set	37.5%	25%	37.5%	37.5%	25%

change in performance will be done visually by adding arrows in Figure 4.5, where the arrows will signalize the transition from one point to another (e.g., arrow from point 27 to point 34).

There are three possibilities for the direction of the arrows:

- Towards top left corner: this transition or inclusion of an attribute causes a rise in performance as the new data point respectively prediction method achieves a higher accuracy and lower conservativeness. These arrows will be depicted in green.
- Towards top right or bottom left corner: this inclusion causes an improvement of one performance measure, but also at the same time a worsening of the other performance measure. For example, an arrow pointing towards the top right corner means a rise in prediction accuracy, but also a rise in conservativeness. These arrows are depicted in yellow.
- Towards bottom right corner: this inclusion causes a worsening of both performance measures and thus an overall decline in performance. These arrows are depicted in red.

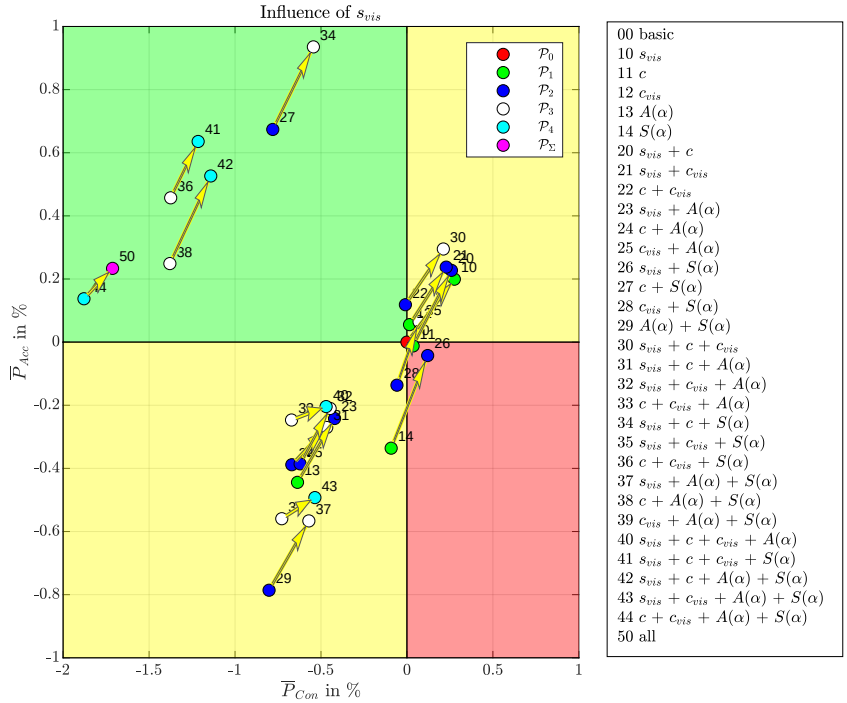

Figure 4.6: Influence of the inclusion of the visible distance s_{vis} on the prediction performance

.

In Figure 4.6 – 4.10 the transitions are depicted in form of arrows for every topological attribute. There already can be seen some tendencies on first sight, but to get an overall view over the influence of the respective variable, the cumulated result in Figure 4.11 is used. In this figure, all single arrows for one variable are summed up vectorially, in order to get an overall trend of the behavior. The resulting arrows are

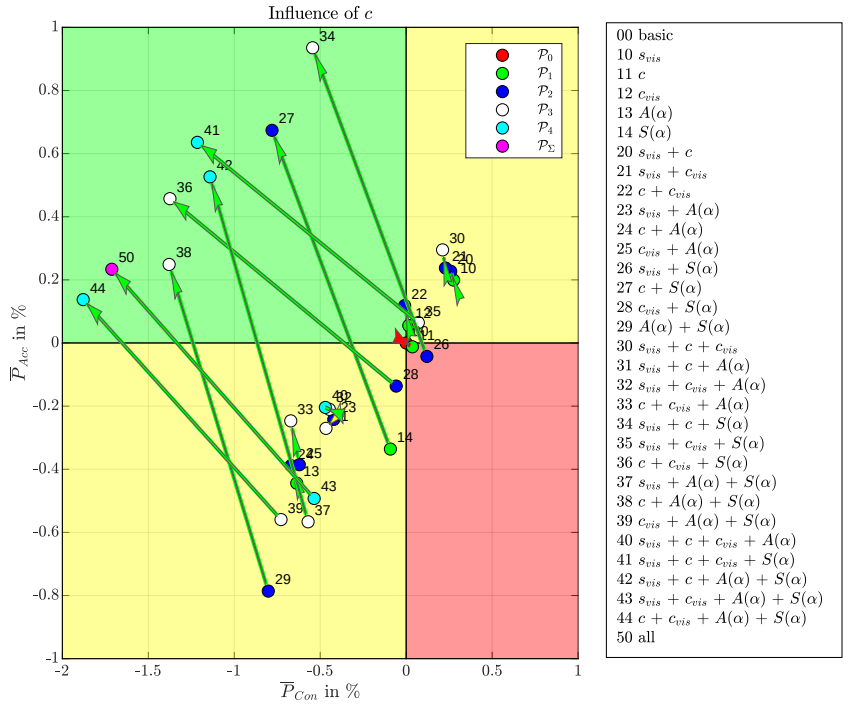

Figure 4.7: Influence of the inclusion of the curvature c on the prediction performance

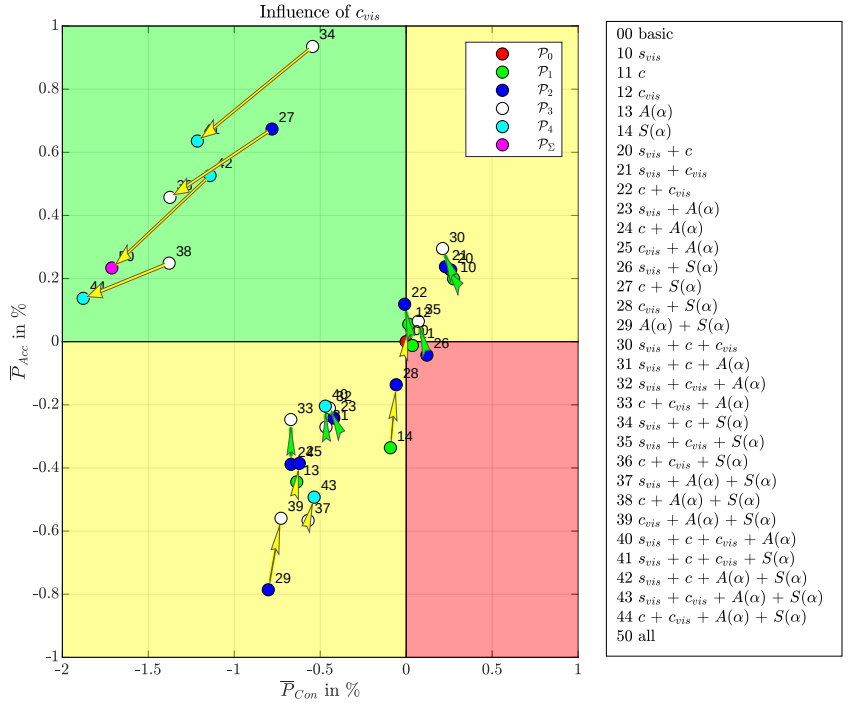

Figure 4.8: Influence of the inclusion of the maximum curvature c_{vis} on the prediction performance

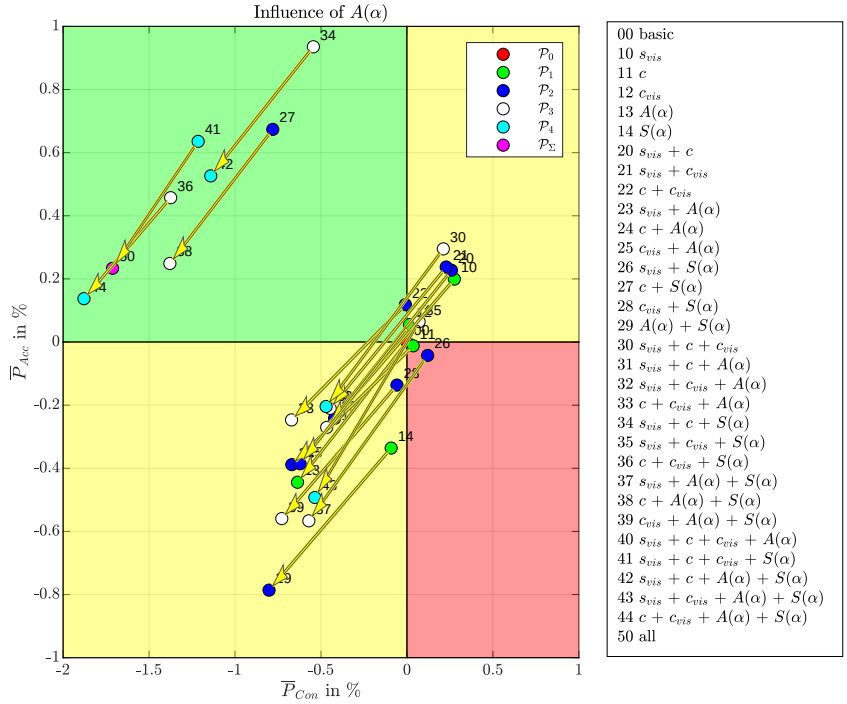

Figure 4.9: Influence of the inclusion of the magnitude of the slope $A(\alpha)$ on the prediction performance

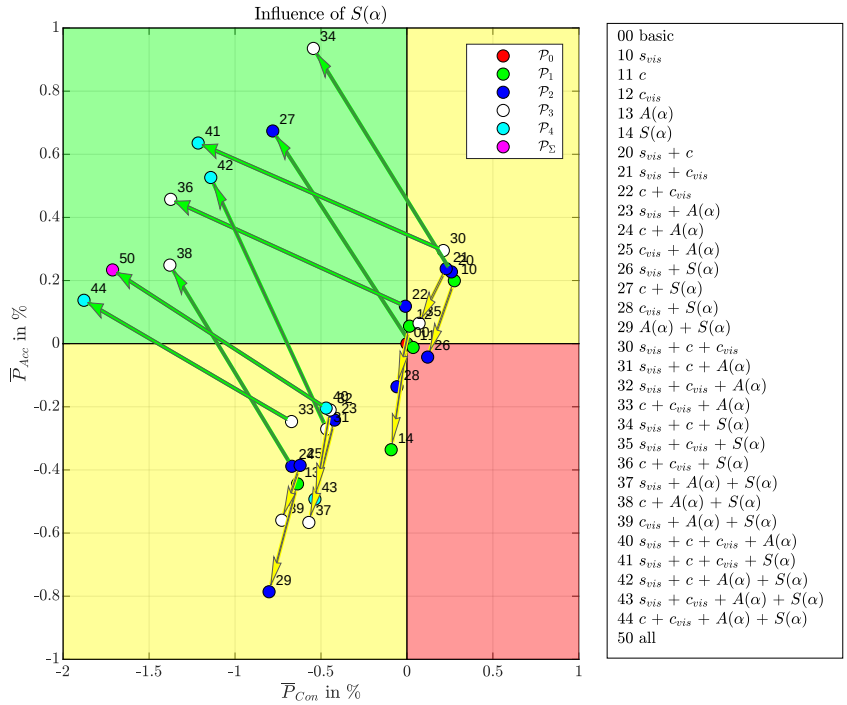

Figure 4.10: Influence of the inclusion of the sign of the slope $S(\alpha)$ on the prediction performance

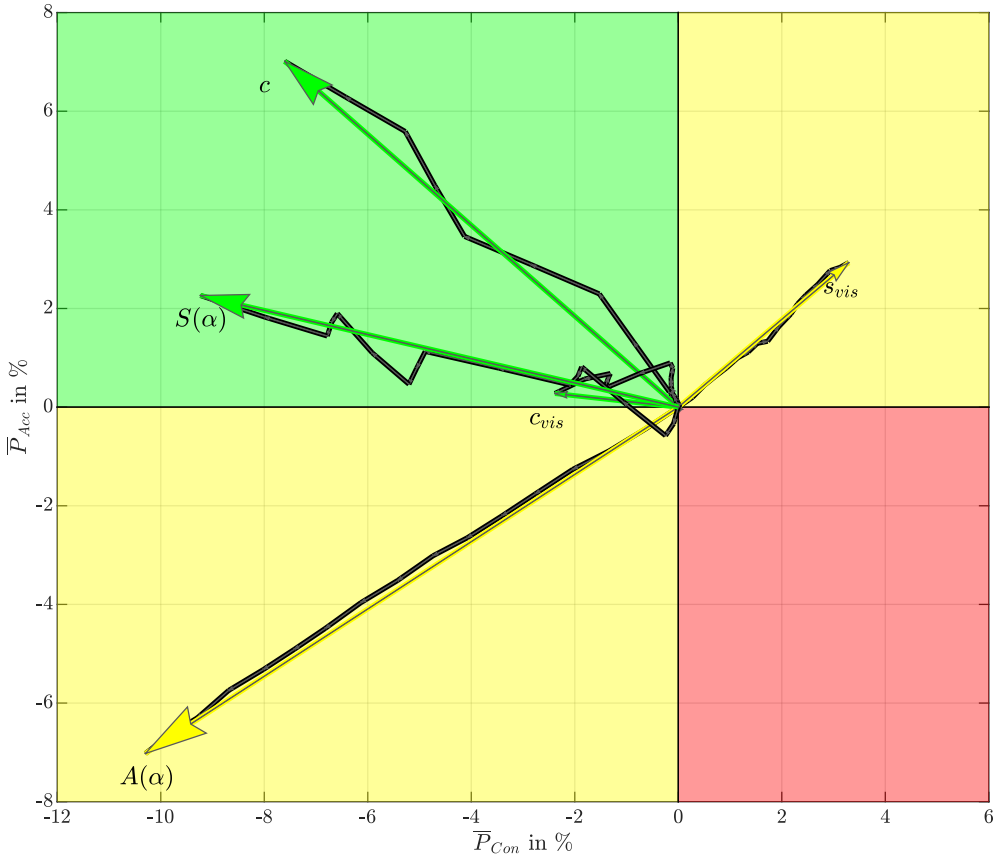

Figure 4.11: Cumulated influences of the respective topological attributes

colored equally as the "small" arrows, i.e. if the arrow points towards the top left corner it is colored green and so on.

The results of this analysis are, that the additional inclusion of the curvature c, the maximum curvature c_{vis} and the sign of the slope $S(\alpha)$ leads to a better performance in general. This was to expect, as it could be seen before that the consideration of the curvature and the sign of the slope guarantees a good performance. The maximum curvature in the visible distance c_{vis} is in some kind related to the curvature and therefore it is no surprise that it also has a good cumulated influence. The curvature has the best cumulated result of those 3 variables, as the direction of the resulting arrow is almost exactly towards the top left corner, which means that both performance measures are improved equally. The arrow of $S(\alpha)$ shows that the conservativeness is more reduced than the prediction accuracy is raised.

The additional inclusions of the magnitude of the slope $A(\alpha)$ and the visible distance s_{vis} lead to two different, but not categorically bad, results. On the one hand, the inclusion of $A(\alpha)$ in almost all data points leads to a drop of the prediction accuracy, but also to an improvement by lowering the conservativeness of the prediction. Therefore, the resulting arrow points towards the bottom left corner. On the other hand, the inclusion of s_{vis} leads to an opposite behavior, where the prediction accuracy is raised,

but also the prediction conservativeness. Hence, the resulting arrow points in the opposite direction. From an intuitive point of view, this is quite surprising, as one would expect that the visible distance is an attribute with a high positive influence on the driving behavior.

4.6 Comparison of the behavior for short and long prediction horizons

In the previous section all 2D-plots were created based on the mean values of the prediction performance measures, with the mean considering all prediction steps $n = 1 \dots 20$ from Figure 4.1 – 4.4. Now the question arises, if the before observed behavior is the same over the whole prediction horizon or if there are differences between, e.g., short and long prediction horizons. Therefore, the prediction horizon is divided in two parts: short prediction horizon $n_1 = 1 \dots 10$ and long prediction horizon $n_2 = 11 \dots 20$. The same analyses as in the previous chapter were executed for these two parts each and depicted in Figure 4.12 – 4.15.

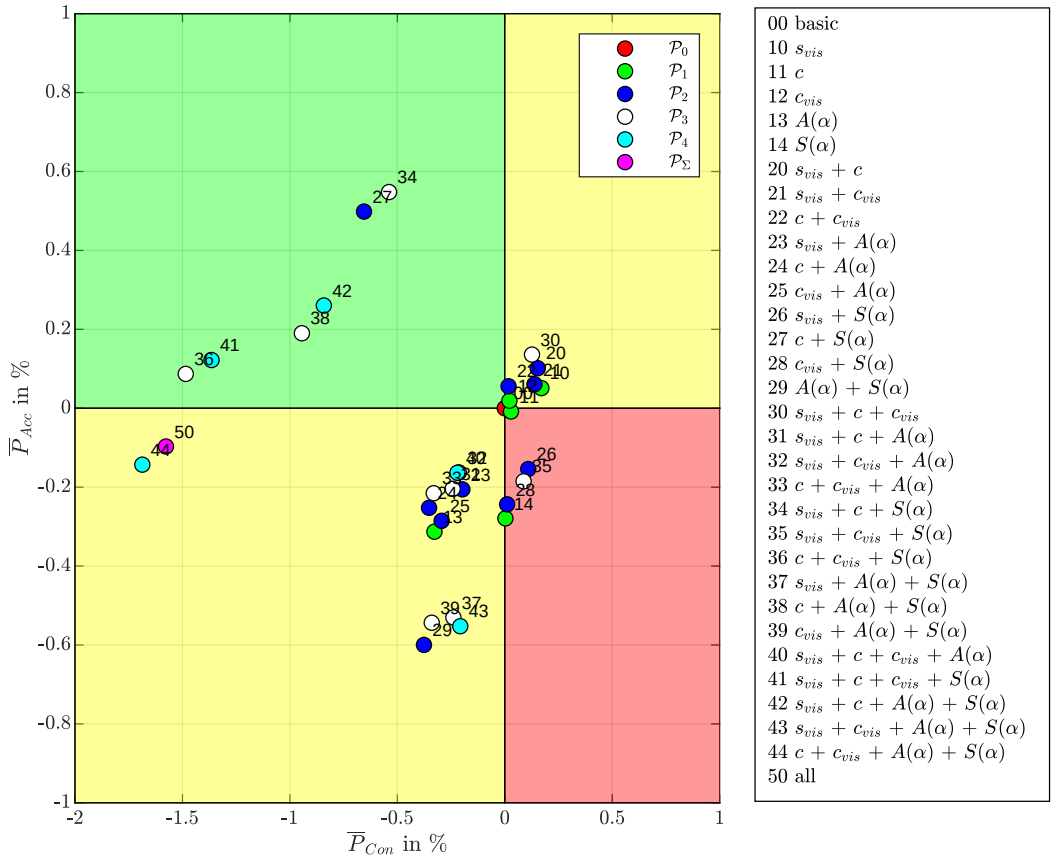

Figure 4.12: 2D-Plot of the performance measures of all possible combinations of the topological attributes for a short prediction horizon $n = 1 \dots 10$.

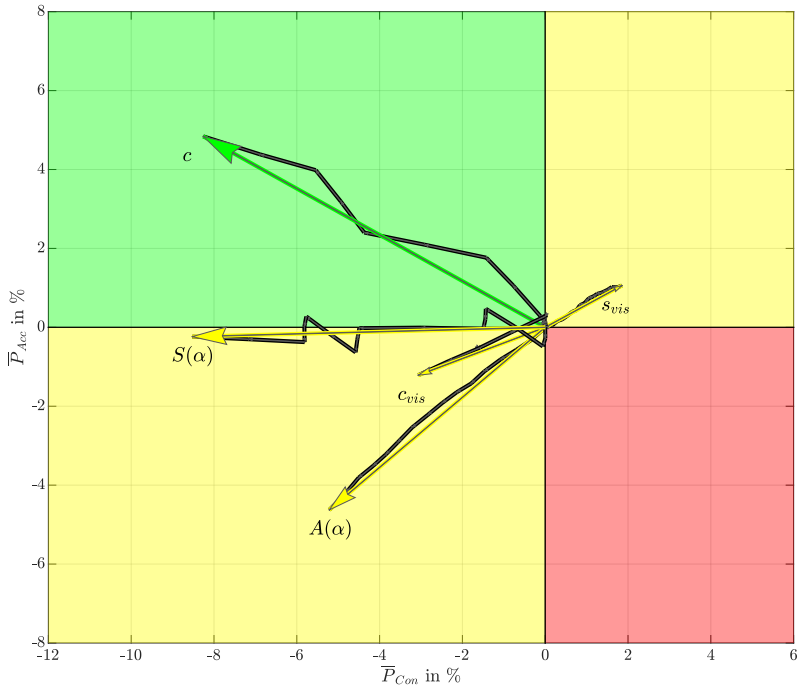

Figure 4.13: Cumulated influences of the respective topological attributes for a short prediction horizon $n = 1 \ldots 10$

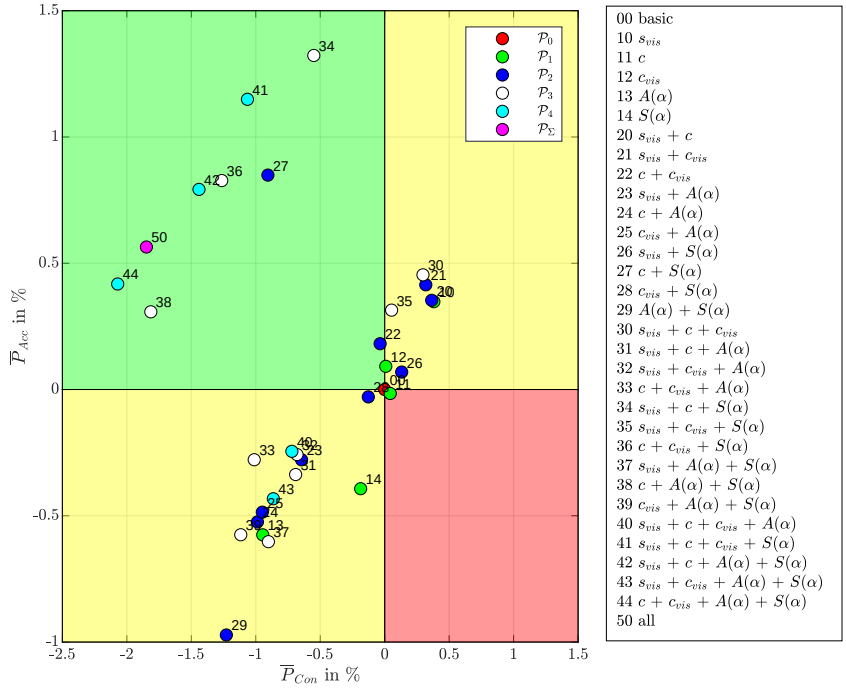

Figure 4.14: 2D-Plot of the performance measures of all possible combinations of the topological attributes for a long prediction horizon $n = 11 \ldots 20$.

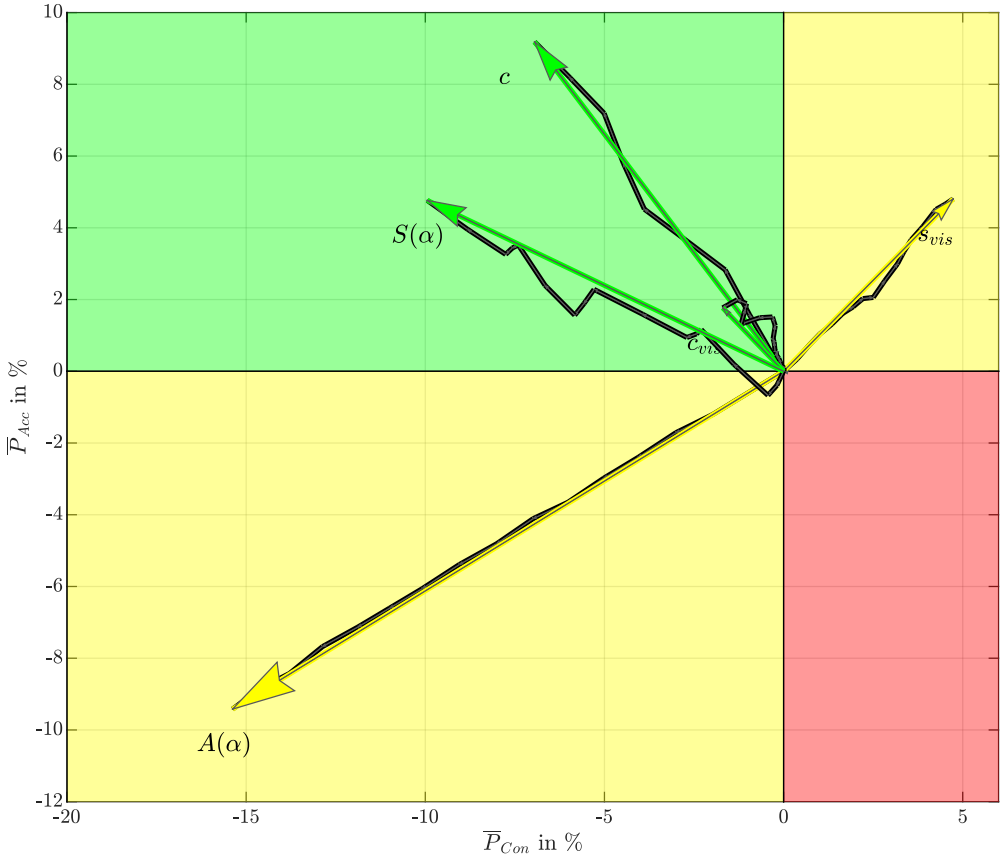

Figure 4.15: Cumulated influences of the respective topological attributes for a long prediction horizon $n = 11 \ldots 20$

Figure 4.12 and Figure 4.13 show the results for the short prediction horizon. In comparison to the overall behavior, it can be seen that the performance in general is worse, as the data points in Figure 4.12 are nearer to the reference point and less scattered in the performance measure space. Especially in the green area, which indicates a clearly better performance than the basic prediction method with first principles, contains only six data points, two less than in the approach over the entire prediction horizon. Furthermore, in Figure 4.13 can be seen, that the influences of the inclusion of the topological attributes is smaller and rather directed towards worse performance than with the overall approach. The resulting cumulated arrows are shorter and rotated anti-clockwise, which leads to the fact, that, e.g., the inclusion of the sign of the slope $S(\alpha)$ on average brings no improvement in prediction accuracy.

The results of the long prediction horizon show an opposing behavior. On first sight, Figure 4.14 looks similar to Figure 4.5, but in this case the axis limits were changed in order to show all data points. This means that, in comparison to the approach over the entire prediction horizon, the data points in the green area are spreaded wider and thus the performance of these approaches is better for longer prediction horizons. Also the influences of the inclusion of the topological attributes are higher, as the resulting

cumulated transition arrows in Figure 4.15 are longer, whereby the axis limits were adapted too. Regarding the direction of the arrows, a slight clockwise rotation can be detected, but the main difference compared to the overall approach is the length of the arrows and thus the impact of the inclusion of the respective attributes.

4.7 Comparison of different sampling rates

As described in Section 2.4, the validation of the models is done by drawing N_S random samples of the output nodes. Due to this process, every output node generates N_S predicted trajectories, resulting in a set of predicted trajectories, which are described by their mean trajectory and its standard deviation.

In order to determine the influence of the number of samples N_S on the prediction quality, three different sampling rates will be deployed at the 32 prediction approaches from this chapter. The result of this observation can be seen in Figure 4.16.

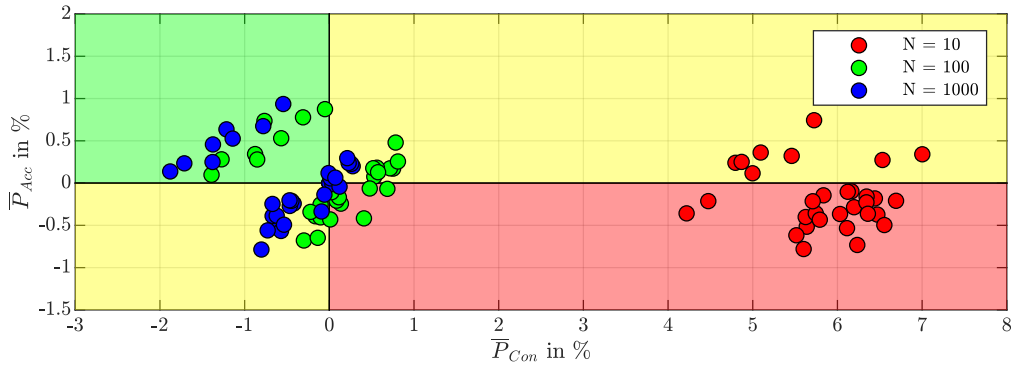

Figure 4.16: Comparison of the quality of prediction models based on the number of validation samples N_S

It becomes apparent, that the number of validation samples has an influence on the prediction results. While the influence on the prediction accuracy (vertical axis) is insignificant, the difference regarding the conservativeness is obvious. With a number of samples $N_S = 10$, the prediction conservativeness is up to 7 % worse than with $N_S = 100$ or respectively $N_S = 1000$. The difference between the two higher N_S is rather small with a maximum difference of 1 %.

To sum up, the prediction is the less conservative, the higher the number of validation samples N_S is, whereby the difference in conservativeness from $N_S = 100$ only changes slightly.

In this chapter it became apparent that the topological attributes have an impact on the quality of the prediction models. Especially the consideration of the curvature and the sign of the slope of the road in the Bayesian network yields good results.

Furthermore, it was shown that the prediction models perform differently for short and long prediction horizons and that especially the prediction conservativeness is dependent on the validation sample rate.

Chapter 5

Validation of different identification strategies

In the previous chapter, the influence of the choice of the considered topological attributes on the quality of the prediction model was determined. In this chapter, the influence of the choice of the training data will be analyzed. Therefore, a measure for the "richness" of a dataset will be introduced, in order to define three different identification strategies and to validate them subsequently.

5.1 Methodology for identification

5.1.1 Definition of the richness

In Chapter 3 and especially in Figure 3.3 it could be seen, that the used data covers a wide range in terms of the topological attributes and that the certain attributes are differently distributed along the route. The question arises, how the distribution of these attributes within the training dataset, or respectively, how the "richness" of a chosen training dataset affects the prediction performance. The richness of a consecutive segment of the route $\mathcal{R}_{[s(k'),\, s(k')+L_{\mathrm{sel}}]}$ – with a spatial length L_{sel} – is defined by comparing the histograms of the segment and the route $\mathcal{R}_{[0,\, L_{\mathrm{tot}}]}$ – with a total spatial length L_{tot}. It is assumed that the optimal or "richest" selection of the segment would be the one, whose histogram is the most "similar" to the histogram of the whole route. This means, that the histogram of the selection would cover exactly a fraction $L_{\mathrm{sel}}/L_{\mathrm{tot}}$ of each bin m. As richer data than the fraction $L_{\mathrm{sel}}/L_{\mathrm{tot}}$ in one bin comes at the cost of reduced data in another bin, it is assumed that this additional share does not increase the richness of the respective segment. It is assumed that the M bins cover equal ranges in terms of the respective data.

The concept of the richness is depicted in Figure 5.1. On the left side, an exemplary histogram of the whole route is given, whose bins are colored in white. Additionally, the histogram of a exemplary chosen road segment is depicted in blue. On the right side of the figure, the seventh bin is analysed regarding the richness of the chosen segment. There it can be seen, that the bin of the whole route (a) represents the

maximum achievable score. The optimal score for this certain bin is depicted in green (b), which is determined by multiplying the total score with $L_{\text{sel}}/L_{\text{tot}}$. In lightblue ($c$), the actual score of the chosen segment is depicted. As explained above, a higher score than the optimal score comes at the cost of another bin, and so it is neglected for the determination of the richness, which is shown in blue (d).

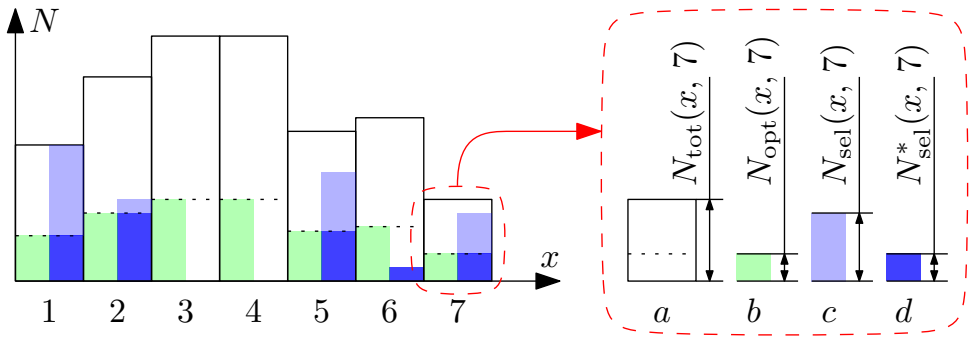

Figure 5.1: Concept of "richness" calculation using the $M = 7$ bins of a given histogram of the total dataset. Details are given for bin $m = 7$. a) Single bin with maximum achievable score (dotted). b) "Optimal" share per bin for this case. c) Actual share per bin for this case. d) Share that contributes to the "richness" score.

The above given visual explanation of the determination of the richness can be expressed in three or respectively four equations. The richness $R_x(k')$ of a route segment with a spatial length L_{sel}, $\mathcal{R}_{[s(k'),\, s(k')+L_{\text{sel}}]}$ connected to a generic variable x can be calculated as

$$R_x(k') = \frac{1}{M} \sum_{m=1}^{M} \frac{N^*_{\text{sel}}(k',\, m)}{N_{\text{tot}}(m)} \tag{5.1}$$

$$N^*_{\text{sel}}(k',\, m) = \min(N_{\text{opt}}(\, m),\, N_{\text{sel}}(k',\, m)) \tag{5.2}$$

$$N_{\text{opt}}(m) = N_{\text{tot}}(m) \cdot \frac{L_{\text{sel}}}{L_{\text{tot}}}. \tag{5.3}$$

Consequently, the total "richness" of a segment starting at $s(k')$ can be calculated as

$$R(k') = \frac{1}{3} \left(R_c(k') + R_{s_{\text{vis}}}(k') + R_\alpha(k') \right). \tag{5.4}$$

5.1.2 Definition of the identification strategies

Before defining the identification strategies, the available dataset has to be divided in an identification and validation part. As mentioned in Chapter 3, the available dataset consists of a total of 39 drives in both directions. For this approach, the dataset will be divided almost equally, with the first 20 drives used for identification and the last 19 drives used for validation.

In the previous section, the richness was introduced to describe the distribution of the topological attributes within a certain segment of the route. In the following, the richness is used as a basic criterion for the choice of the training data within the

identification part of the dataset. Furthermore, it is used to define three different strategies, which are compared for the total route with a length of $L_{\text{tot}} = 5\,\text{km}$:

A) The whole route (both directions) is utilized for identification.

B) The "richest" (consecutive) $L_{\text{sel}} = 2\,\text{km}$ of the route (both directions) are used for model identification.

$$k_{\text{B}} = \arg\max_{k'} R(k') \;\text{ s.t. }\; s(k') + L_{\text{sel}} < L_{\text{tot}} \tag{5.5}$$

C) The "least rich" (consecutive) $L_{\text{sel}} = 2\,\text{km}$ of the route (both directions) are used for model identification.

$$k_{\text{C}} = \arg\min_{k'} R(k') \;\text{ s.t. }\; s(k') + L_{\text{sel}} < L_{\text{tot}} \tag{5.6}$$

This methodology is summarized in Figure 5.2, where the identification part is shaded in blue and the validation part is shaded in green. Furthermore, within the identification part, the blue shade shows the difference in terms of the used data segments for identification based on the richness.

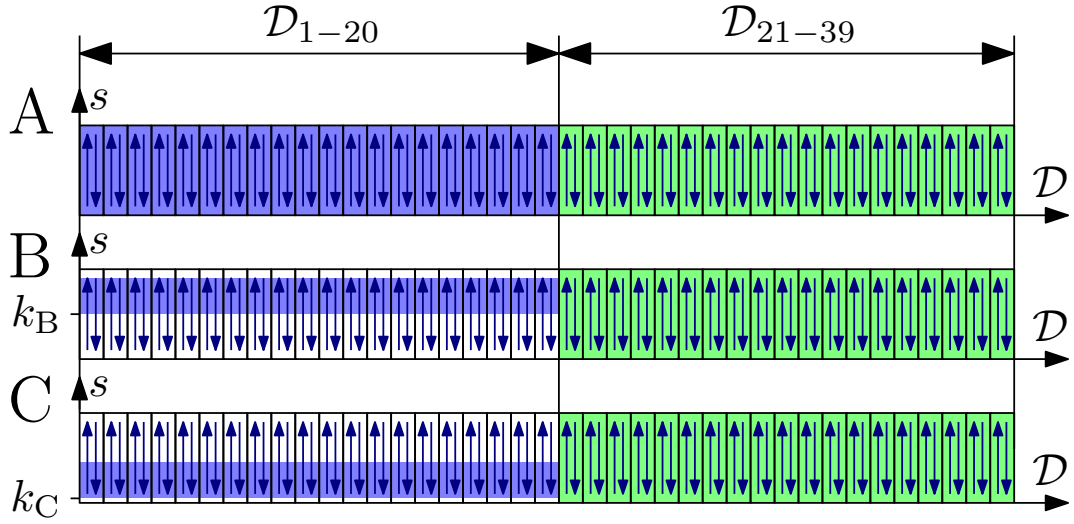

Figure 5.2: Concept of data selection for the identification. The first 20 drives $\mathcal{D}_{1-20}$ are used for identification (blue shaded parts) and the remaining 19 drives $\mathcal{D}_{21-39}$ (whole route) for validation.

The resulting richnesses according to (5.1)-(5.4) of the strategies are shown in Table 5.1, where the "richness" of the curvature appears as the main difference between B and C.

In contrast to the previous chapter, where 32 models were trained, for this approach only 6 models will be trained per identification strategy, as the focus lies on the comparison of the strategies, and not particularly on the comparison of the models. For this purpose, the following models will be trained:

Table 5.1: Calculated richnesses for B) the richest segment and C) the least rich segment

Strategy	$R_{s_{\text{vis}}}$	R_c	R_α	R
B	0.29	0.37	0.09	0.25
C	0.26	0.16	0.11	0.18

$\mathcal{P}_0$ ... first principles

$\mathcal{P}_1$ ... $\mathcal{P}_0$ + visible distance $\{s_{\text{vis}}(k')\}$

$\mathcal{P}_2$ ... $\mathcal{P}_0$ + curvature $\{c(k')\}$

$\mathcal{P}_3$ ... $\mathcal{P}_0$ + max. curvature within $s_{\text{vis}}(k')$ $\{\bar{c}_{\text{vis}}(k')\}$

$\mathcal{P}_4$ ... $\mathcal{P}_0$ + slope $\{|\alpha(k')|, \text{sign}(\alpha(k'))\}$

$\mathcal{P}_\Sigma$ ... all previous

Although the models with only one considered topological attribute did not perform best in the previous chapter, they are used for this observation, as they may be best suited for showing differences between the identification strategies. For example, one could expect, that prediction approach $\mathcal{P}_2$ performs better with the "rich" strategy B than with C, as the training data is "richer" regarding the curvature. Furthermore, the basic method with only first principles and the method with all topological attributes are deployed for better comparison.

5.2 Methodology for validation

The validation is done in accordance to the visualization from Figure 5.2, where the last 19 drives of the dataset are used for validation. In contrast to the identification, there is no choice of certain road segments based on the richness, but the whole route will be used in both directions for validation.

The performance of the prediction methods $\{\mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4, \mathcal{P}_\Sigma\}$ will be assessed based on the measures defined in Section 4.2. The assessment will mainly be done in relation to the basic method $\mathcal{P}_0$. These six methods are compared for all identification strategies {A, B, C}, which are defined above.

5.3 Results and discussion

Three different training datasets for the three identification strategies {A, B, C} are deployed. The Bayesian nets are structured based on the topological attributes, which yields six different prediction methods or structures of the net $\{\mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4, \mathcal{P}_\Sigma\}$. All in all, 18 different models are identified and validated and the results of these validations will be compared and discussed in the following.

Looking at Figure 5.3, it can be stated that the richness (B versus C) as well as the volume (A vs. B | C) of the identification dataset influences the quality of the

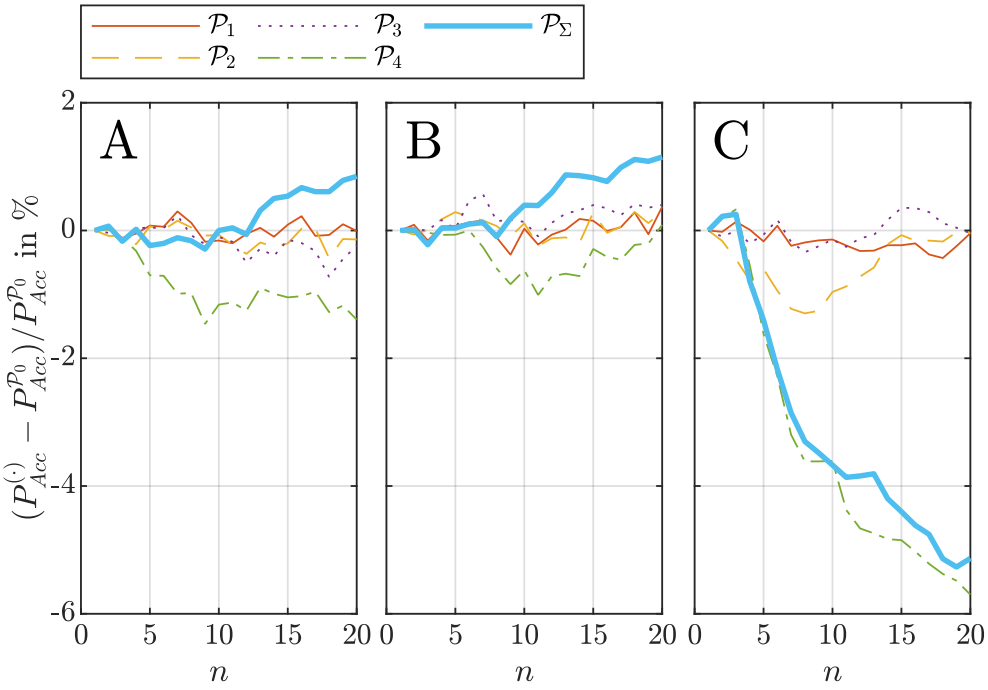

Figure 5.3: Comparison of the Bayesian network-based prediction methods by relating them to the basic method $\mathcal{P}_0$. Identification was done based on A) the whole route, B) the richest segment and C) the least rich segment of the route.

prediction models. While the accuracies of the different prediction methods are very similar for the identification strategy B with a rich/variable dataset, the accuracies for identification strategy C differ significantly. Especially, the model $\mathcal{P}_\Sigma$ considering all influences – together with $\mathcal{P}_4$ – has the worst performance at identification strategy C. One reason could be, that $\mathcal{P}_\Sigma$ tries to find a good generic result and therefore has to follow the weakest link (in this case $\mathcal{P}_4$), while other models only have to consider one additional influence.

On the other hand, $\mathcal{P}_\Sigma$ has the best performance at identification strategy A, which used the whole identification dataset for identification – especially for longer prediction horizons. This is an important result, as it has the highest practical relevance (as the presence of sufficient identification data exceeding our limited dataset can be assumed). This advantage at longer prediction horizons is also present if the identification dataset is rich enough, as for strategy B.

Figure 5.4 displays the evaluated values of (4.4). It can be observed, that $\mathcal{P}_\Sigma$ is the least conservative prediction model for almost all predicted steps and identification strategies. For strategy C, it gets – compared to other methods – even less conservative for a longer prediction horizon, which might also be one of the reasons for the drop in accuracy observable in Figure 5.3-C. Looking at the general prediction performance, it can be concluded that – with an identification dataset which is rich enough regarding volume and "richness" – the combined prediction approach $\mathcal{P}_\Sigma$ performs best.

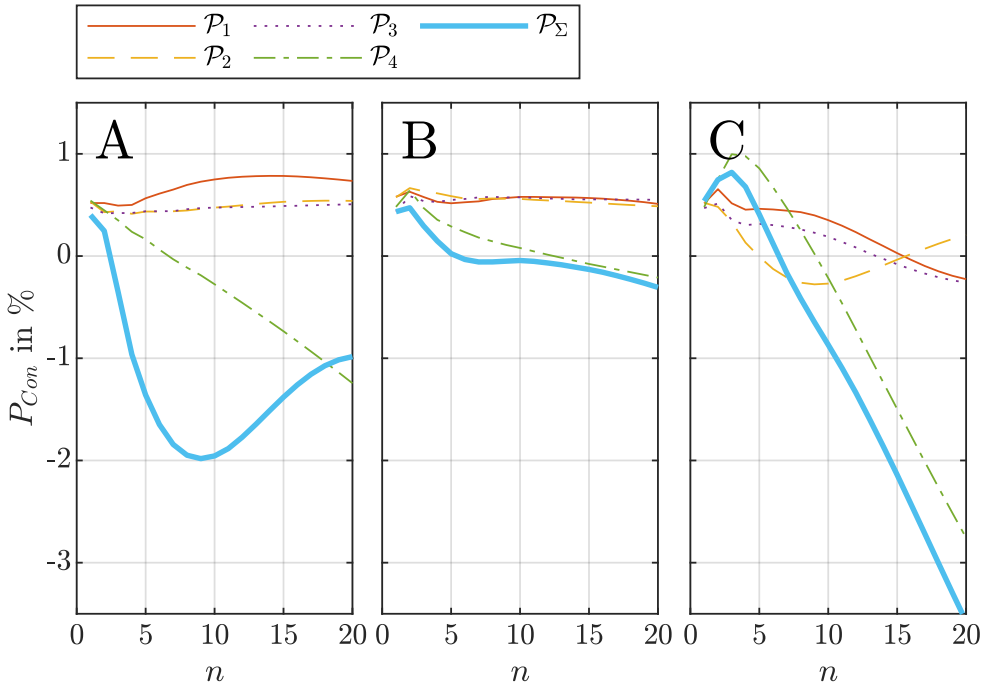

Figure 5.4: Calculated prediction conservativeness resulting for prediction models identified with A) the whole route, B) the richest segment and C) the least rich segment of the route.

These observations can also be made by looking at Figure 5.5, where the comparison is done based on a two-dimensional representation, as introduced in Chapter 4. It can be seen, that $\mathcal{P}_\Sigma$ has the best performance when identifying with the full dataset, as the prediction is the least conservative and the accuracy is relatively high. The approach $\mathcal{P}_\Sigma$ performs also well for the other identification strategies, with the difference that the accuracy is rather low for the identification strategy using the poor dataset.

The question, which arises in this course, is how the volume of the training dataset influences the quality of the prediction models. Therefore, five different road segments with increasing length were used for the identification of 32 different prediction methods, as known from Chapter 4, whereby the start of every of these five segments is the start of the route. These five approaches are compared in a two-dimensional representation in Figure 5.6.

It becomes apparent, that the volume of the identification dataset has a significant influence on the quality of the prediction models. For short road segment lengths (1 km and 2 km), the sensitivity of the prediction performance concerning the considered topological attributes is relatively high, which is obvious because of the wide spread of the data points. But, the longer the dataset is, the smaller is the spread of the data points, as it can be seen for the higher lengths. Another interesting fact is that the prediction approaches considering the topological attributes perform only better than the basic method $\mathcal{P}_0$ for the models identified with the highest length of 5 km, which is equal to the whole dataset. At all other road segment lengths, the models perform worse than the basic method, as they are placed bottom left in relation to the basic method, which is depicted as a black data point. One shortcoming of the

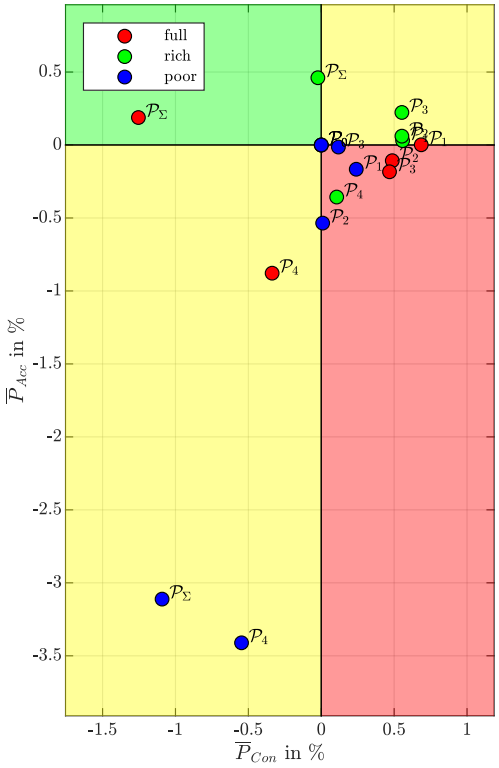

Figure 5.5: Two-dimensional comparison of the results from Figure 5.3 and Figure 5.4 for three different identification strategies

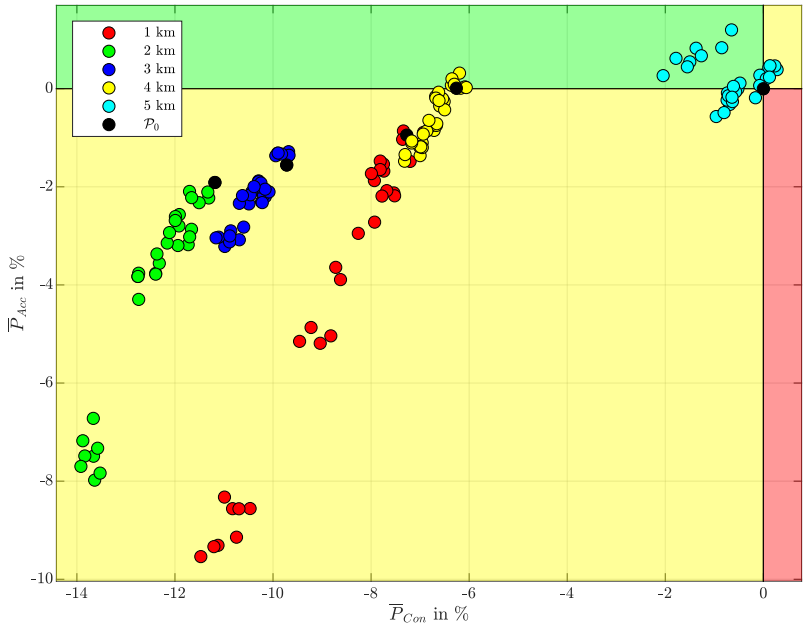

Figure 5.6: Comparison of the prediction quality of models identified with datasets with different length

models identified with the whole dataset is that they are up to 14 % more conservative than the models identified with smaller datasets, while therefore the accuracy is lower. This can be seen as a trade-off between accuracy and conservativeness, whereby the improvement in accuracy is paid with a higher worsening in conservativeness.

Chapter 6

Practical application – ACC

To demonstrate the performance of the combined prediction method $\mathcal{P}_\Sigma$ on a practical application, it is – as well as comparative methods – evaluated using a simple ACC where the region of the vehicle's position is assessed using the respective prediction methods to be able to guarantee a required safe inter-vehicle distance.

6.1 Introduction to ACC and MPC

In the following, the basic concepts of ACC and MPC will be explained, in order to comprehend the deployment of the prediction method in this chapter.

6.1.1 Adaptive Cruise Control

As already mentioned in Chapter 1, ACC systems belong to the group of ADAS (Advanced Driver Assistance Systems). Therefore, its goal is to relive the driver of simple or monotone tasks in road traffic and to perform these tasks in a way, which achieves an increase of driving comfort, fuel efficiency or safety. The task, which the ACC takes over from the driver, is to follow the preceding vehicle with an appropriate distance at a certain velocity. Therefore, it is an extension or enhancement of the traditional cruise control, where the controller tracks only the reference velocity without consideration of the distance to the preceding vehicle. So, the ACC tracks a reference velocity v_{ref} that is set by the driver while maintaining a safe inter-vehicle distance d_{s} to the preceding vehicle.

The determination of this safe inter-vehicle distance is done according to a so called spacing policy, which defines the desired steady state spacing between two consecutive vehicles during vehicle following. The choice of the spacing policy is done at the beginning of the ACC system design and plays a significant role in various aspects such as traffic capacity, fuel/energy consumption, driver's subjective acceptance, and safety.

Criteria for the choice of spacing policies

Important criteria for the choice and evaluation of spacing policies are summarized in [34] and [35]:

1. Individual vehicle stability:
   The spacing policy and its associated control law have to ensure individual vehicle stability, which means that the spacing error

   $$\delta_i = d_i - d_{des}$$

   of the ego vehicle should converge to zero if the preceding vehicle is driving at constant speed ($\dot{v}_{i-1} = 0$). d_i denotes the current distance between the vehicles and d_{des} stands for the desired distance.

2. String stability:
   The string stability is a group property of a queue of ACC controlled vehicles. As the spacing error propagates towards the end of the queue, the spacing policy has to ensure string stability, in order to avoid a diverging spacing error.

3. Traffic flow stability:
   The selected spacing policy should guarantee traffic flow stability, which describes the variations of traffic flow in response to small disturbances in traffic density.

4. Collision avoidance:
   The spacing policy should ensure safety for the ego vehicle, even when unpredictable actions of the preceding vehicle occur.

5. Comfort of the driver:
   The spacing policy should provide similar driving patterns to human driving behaviors, in order to avoid possible discomfort for the driver and passengers.

Types of spacing policies

Besides the different criteria for spacing policies, also different types of spacing policies exist, which are summarized in [35]. In general, there are two major groups of spacing policies: Constant spacing policies and variable spacing policies.

As the name says, constant spacing policies are based on a constant inter-vehicle spacing, independent of the driving environment. On the one hand, the advantages of this approach are a low computational load and a high traffic capacity when choosing a small distance. On the other hand, it has been proven that this policy can not guarantee string stability if ACC systems with linear controllers are used. This leads to a bad drive quality and potentially also to collisions. One solution to this problem would be inter-vehicle communication, which is not available in road traffic so far.

In contrast to this approach, variable spacing policies define the desired inter-vehicle spacing as a function of the vehicle's velocity. Within this group of policies, four different types from existing works are summed up in [35]:

1. Time headway policy:
   The time headway policy is the most common spacing policy. Time headway describes the period during which the front bumper of the preceding vehicle and the front bumper of the ego vehicle pass a fixed position on the road. The desired inter-vehicle distance d_s consists of a minimum distance d_0 and a velocity-dependent part, where the velocity of the ego-vehicle $v^{\mathcal{E}}(k)$ is multiplied with the time headway.
   $$d_\mathrm{s}(k) = v^{\mathcal{E}}(k)\, t_\mathrm{h} + d_0$$
   Furthermore, two types exist for the time headway: Constant time headway (CTH) and variable time headway (VTH). As the name says, at CTH the time headway t_h is constant. At VTH, the time headway is a variable, which is also dependent on the velocity. This approach has the advantage, that one can implement an upper limit for the time headway as the velocity gets higher, in order to maintain the traffic throughput and to not reduce it further due to large distances between the vehicles.

2. Traffic flow stability policy:
   CTH policies cannot ensure traffic flow stability. Therefore, traffic flow stability policies were invented as an alternative in order to overcome this problem, as they incorporate the traffic density in the determination of the desired distance.

3. Constant safety factor policy:
   Constant safety factor (CSF) spacing polices were proposed to improve safety and minimize the possibility of collisions. The determination of the desired distance is based on a safe stopping distance D_stop and a safety factor K:
   $$d_\mathrm{s}(k) = D_\mathrm{stop}(k) \cdot K$$

4. Human driving behavior policy:
   The idea of human driving behavior policies is that the ACC should act similar to a human driver in terms of spacing. For this purpose, real human driving data is recorded and utilized to develop a spacing policy
   $$d_\mathrm{s}(k) = A + T \cdot v^{\mathcal{E}}(k) + G \cdot v^{\mathcal{E}}(k)^2$$
   where A represents the inter-vehicle spacing at rest, and T and G are the coefficients of the first and second order terms, which are determined from the real world data.

After the choice of the spacing policy, the realization of the ACC is pending. This is usually done by formulating the ACC system as an optimal control problem and by solving it within a MPC (Model Predictive Control) framework. The basic ideas and concepts of MPCs will be expounded in the following section.

6.1.2 Model Predictive Control

This section is mainly based on the books [36] and [37], which describe the concepts of model predictive control comprehensively.

Working scheme

MPC belongs to the group of predictive controllers and is one of the most common type of this group. The working concept of MPC will be explained based on Figure 6.1:

1. In a MPC, a physical model of the plant to be controlled is used, in order to predict the controlled variable $\hat{y}$ for $k = 1 \dots n_{\mathrm{PH}}$ time steps. These predicted values are dependent on past output values $y(t)$ and input values $u(t)$, as well as future input values $\hat{u}(t+k|t)$ for $k = 0 \dots n_{\mathrm{CH}} - 1$ time steps.
2. These future input values $\hat{u}(t|t) \dots \hat{u}(t+n_{\mathrm{CH}}-1|t)$ are computed by solving an optimal control problem, whereby the future output values $\hat{y}(t+1|t) \dots \hat{y}(t+n_{\mathrm{PH}}|t)$ should track a reference $r(t+1|t) \dots r(t+n_{\mathrm{PH}}|t)$. The cost function is usually a quadratic function and incorporates the error between the predicted output values and the reference, as well as the input values.
3. After determining the inputs $\hat{u}(t|t) \dots \hat{u}(t+n_{\mathrm{CH}}-1|t)$, only the first value $\hat{u}(t|t)$ is used as an input for the plant. Subsequently, the process starts again from the next step.

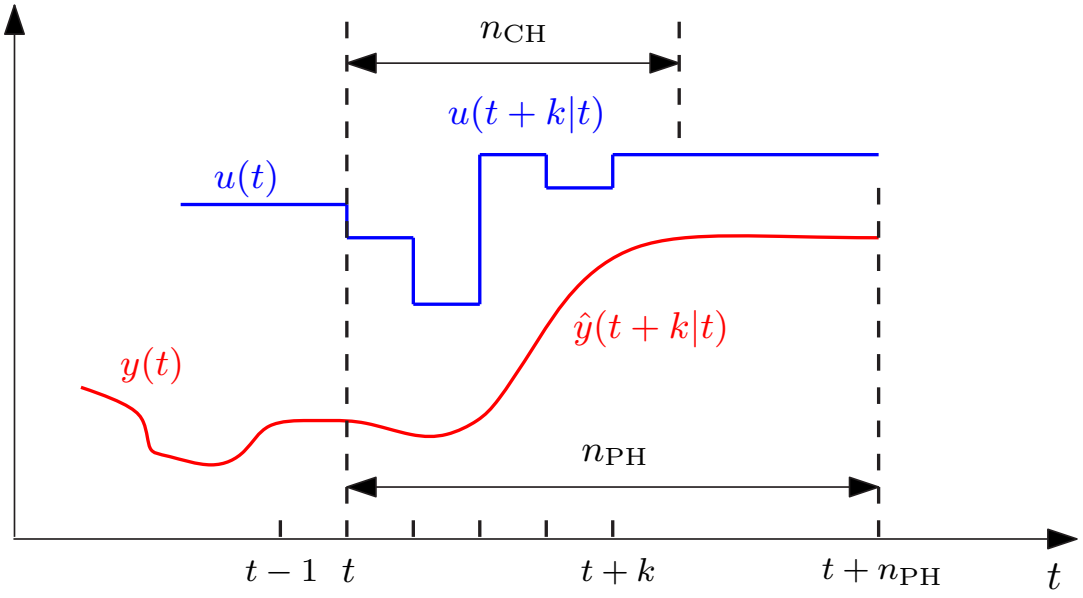

Figure 6.1: Concept of Model Predictive Control

Advantages and Disadvantages

One major advantage of MPC is the simple consideration of constraints. Due to physical limitations, many systems underlie constraints on its inputs, outputs or states.

Generally, this leads to non-linear behavior, which makes the treatment of constraints in classic linear control applications quite difficult. At MPC, the constraints can be considered easily in the minimization problem, which means that at every time step a new controller is designed. Further advantages of MPC are:

- The case of multiple inputs and multiple outputs (MIMO) can be handled easily.
- They are suitable for a large number of systems.
- A feed-forward extension allows the compensation of measurable disturbances, as well as a predictive compensation, if the disturbance is predictable.
- The resulting controller can be implemented by a linear control law, whereby this law can change in every time step.

One of those advantages contains a major drawback, namely that in every time step the control law changes, as in every time step a optimization problem is solved. This leads to a high computational burden, which makes the usage at fast changing processes difficult.

Cost Function

For simplicity, we will consider a linear MPC, which we will also use later in the chapter. Therefore, the model of the plant has the form

$$x(k+1) = \mathbf{A}\,x(k) + \mathbf{B}\,u(k) \tag{6.1}$$

$$y(k) = \mathbf{C}\,x(k) \tag{6.2}$$

In order to find the optimal input sequence, the cost function has to be defined. A commonly used cost function has the form

$$\mathcal{V}_k = \sum_{i=n_0}^{n_{\mathrm{PH}}} \left(\hat{y}_{k+i|k} - r_{k+i}\right)^\top Q_i \left(\hat{y}_{k+i|k} - r_{k+i}\right) + \sum_{i=0}^{n_{\mathrm{CH}}-1} \left(\Delta\hat{u}_{k+i|k}\right)^\top R_i \left(\Delta\hat{u}_{k+i|k}\right) \tag{6.3}$$

The cost function is minimized regarding the degrees of freedom $\Delta\hat{u}_{k+i|k}$, which consist of the sequence $\Delta\hat{u}_{k|k} \ldots \Delta\hat{u}_{k+n_{\mathrm{CH}}-1|k}$. The variable n_{CH} is called control horizon and describes from where on the input signal is constant, meaning $\Delta u_{k+n_{\mathrm{CH}}} = \Delta u_{k+n_{\mathrm{CH}}+1} = \cdots = \Delta u_{k+n_{\mathrm{PH}}} = 0$. The prediction horizon n_{PH} specifies the length of the prediction. The matrices $Q_i > 0$ and $R_i > 0$ are tuning parameters, which can be used to weigh certain inputs or outputs more or less. As it can be seen, besides the error between the reference and the output, also the change of the input signal is weighed with R_i. Alternatively, the input signal itself can be considered in the cost function, as it will be done in the following.

6.2 Design of the ACC

6.2.1 Choice of spacing policy

The inter-vehicle distance is determined according to the most common spacing policy, the constant time headway (CTH) policy

$$d_{\mathrm{s}}(k) = v^{\mathcal{E}}(k)\, t_{\mathrm{h}} + d_0, \tag{6.4}$$

where $v^{\mathcal{E}}(k)$ is the actual speed driven, $t_{\mathrm{h}} = 2.6\,\mathrm{s}$ denotes the time headway and $d_0 = 3\,\mathrm{m}$ represents a safe distance for low velocities and during standstill, as deployed in [38].

6.2.2 Formulation of the optimal control problem

As explained in the previous section, the ACC is formulated as an optimal control problem and solved within an MPC framework. The MPC calculates an optimal (input) control sequence $\mathbf{a}^{\mathcal{E}}_{\mathrm{opt}} = [a^{\mathcal{E}}(k'), a^{\mathcal{E}}(k'+1), \ldots, a^{\mathcal{E}}(k'+N)]$ through minimizing a cost function $\mathcal{V}(k', N)$ during a finite time horizon N (starting from the current time step k') by using a model of the vehicle to predict its future states. In this specific example, the advantage of an MPC is that constraints like, for example, the safe inter-vehicle distance (6.4) that must not be exceeded, can be considered easily. The first element of $\mathbf{a}^{\mathcal{E}}_{\mathrm{opt}}$ is applied to the vehicle, and the optimization begins anew from the new vehicle state at time step $k'+1$, see also [5]. The vehicle model within the MPC is formulated as a double integrator of the vehicle's position $s^{\mathcal{E}}(k)$ and discretization leads to

$$x^{\mathcal{E}}(k+1) = \mathbf{A}\, x^{\mathcal{E}}(k) + \mathbf{B}\, a^{\mathcal{E}}(k), \tag{6.5}$$

with

$$x^{\mathcal{E}}(k) = \begin{bmatrix} s^{\mathcal{E}}(k) \\ v^{\mathcal{E}}(k) \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 1 & T_{\mathrm{s}} \\ 0 & 1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0.5\, T_{\mathrm{s}}^2 \\ T_{\mathrm{s}} \end{bmatrix}, \tag{6.6}$$

where T_{s} denotes the sample time and $x^{\mathcal{E}}(k)$ the state. The distance $d(k)$ to the preceding vehicle is defined as

$$d(k, \cdot\,) = s^{\mathcal{E}}(k) - \hat{s}^{(\cdot)}_{\mathrm{lb}}(k) \tag{6.7}$$

utilizing the predicted lower boundary of the spatial distribution $\hat{s}^{(\cdot)}_{\mathrm{lb}}(k)$ within the prediction horizon of the MPC. Both, the method $\mathcal{P}_{\Sigma}$ and $\mathcal{C}$, as well as a frozen time

prediction approach $\mathcal{F}$ [39] are compared in this setup. The resulting optimization problem for the ACC is a quadratic program with linear constraints and follows to

$$\mathbf{a}^{\mathcal{E}}_{\text{opt}} = \arg\min_{\mathbf{a}^{\mathcal{E}}} \; \mathcal{V}(k', N) \tag{6.8}$$

$$\mathcal{V}(k', N) = \sum_{k=k'}^{k'+N} Q\,(v^{\mathcal{E}}(k) - v_{\text{ref}}(k'))^2 + R\,a^{\mathcal{E}}(k)^2 \tag{6.9}$$

$$\text{s.t.} \quad x^{\mathcal{E}}(k+1) = \mathbf{A}\,x^{\mathcal{E}}(k) + \mathbf{B}\,a^{\mathcal{E}}(k) \tag{6.10}$$

$$0\,\text{m/s} \leq v^{\mathcal{E}}(k) \leq 35\,\text{m/s} \tag{6.11}$$

$$d(k, \cdot\,) < d_{\text{s}}(k). \tag{6.12}$$

$v_{\text{ref}}(k')$ is set to the respective maximum allowed speed by law at step k' and $v^{\mathcal{E}}(k)$ is restricted to a realistic range, while $a^{\mathcal{E}}(k)$ is not explicitly restricted as high values are avoided based on the second term (weighted with R) in the cost function. Q and R, which represent the "aggressiveness" of the ACC – where higher ratios of R/Q lead to a less aggressive controller – were manually tuned to $Q = 1$ and $R = 10$. The optimization problem was solved using Yalmip [40] and quadprog [41].

6.2.3 Validation of the ACC

Again, the validation dataset is utilized for simulation, but this time the measurements serve as preceding vehicles which are followed by the controlled vehicle $\mathcal{E}$ with a head start of 10 s for the respective preceding vehicles in the initial state. To compare the influence of the prediction methods, the validation cost function

$$\mathcal{V}_{\text{v}}^{(\cdot)}(k', N) = \sum_{k=k'}^{k'+N} a^{\mathcal{E}}_{\text{opt}}(k)^2 \tag{6.13}$$

was evaluated for all simulation results over the whole simulation time, as the squared acceleration is a measure of both fuel economy [24] and comfort [42] which qualifies it as a good measure for the quality of the results.

6.2.4 Comparative prediction methods

Stochastical approach $\mathcal{C}$

In addition to the prediction method based on a Bayesian network structure, the approach from [4] (from now on with superscript $\mathcal{C}$) is utilized. As a base of the calculation, the time dependent spatial mean $\hat{s}^{\mathcal{C}}_{\mu}(k'+n)$ as well as the spatial standard deviation $\hat{s}^{\mathcal{C}}_{\sigma}(k'+n)$ need to be calculated. $s(k')$ and $v(k')$ represent the initial position and velocity. Based on the assumptions from [4], the stochastic distribution over all steps results in

$$\hat{s}^{\mathcal{C}}_{\mu}(k'+n) = s(k') + v(k')\,(n \cdot T_s) + \frac{a^{\mathcal{C}}_{\mu}}{2} \cdot (n \cdot T_s)^2 \tag{6.14}$$

for the spatial mean. Through simulation the spatial standard deviation can be approximated to

$$\hat{s}_{\sigma}^{\mathcal{C}}(k'+n) \approx \sqrt{\frac{1}{3}} \cdot a_{\sigma}^{\mathcal{C}} \cdot (n \cdot T_s)^{\frac{3}{2}}. \tag{6.15}$$

where $\mu_a^{\mathcal{C}}$ and $\sigma_a^{\mathcal{C}}$ represent the mean and standard deviation of an assumed distribution of acceleration. Based on [4], $a_{\mu}^{\mathcal{C}} = 0.2\,\mathrm{m/s^2}$ and $a_{\sigma}^{\mathcal{C}} = 1\,\mathrm{m/s^2}$ are used.

As it can be seen, this prediction method only bases on first principles, like the position and the velocity, but no topological informations are used.

Frozen time approach $\mathcal{F}$

The frozen time prediction approach from [39] bases on the actual position and velocity and can be calculated with the following equation:

$$\hat{s}_{lb}^{\mathcal{F}}(k+n) = s(k) + v(k) \cdot n \cdot T_S \tag{6.16}$$

6.3 Outcome of the example

The validation costs (6.13) for the respective prediction methods are visualized with box plots in Figure 6.2 and further described in Table 6.1.

It becomes apparent that the developed prediction method $\mathcal{P}_{\Sigma}$ is superior to the method $\mathcal{C}$ which does not take into account the environment and also $\mathcal{F}$ which is relatively simple and only based on first principles. The mean value of $\mathcal{P}_{\Sigma}$ lies approximately 20% lower than the ones of the other approaches, which means that a significant increase in fuel economy and driving comfort can be expected by considering environmental attributes in the prediction of driving behavior.

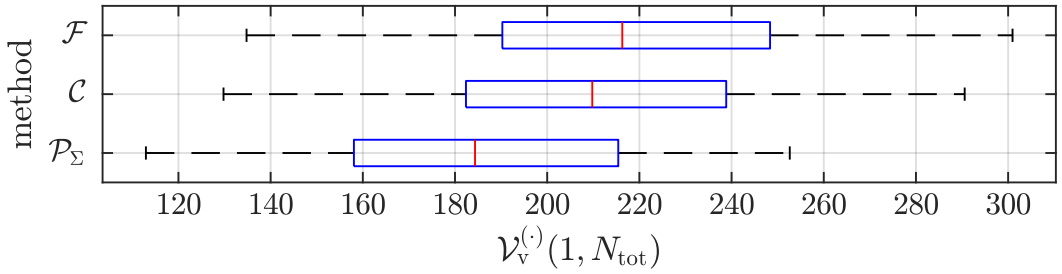

Figure 6.2: Visualization of the evaluated cost function for the respective prediction methods.

In Table 6.1 the boundaries of the boxplot are given explicitly. It can be seen, that for $\mathcal{P}_{\Sigma}$ the range of the cost function is smaller compared to the two other prediction approaches.

Table 6.1: Statistical features of the validation costs for the prediction methods.

$\mathcal{V}_{\mathrm{v}}^{(\cdot)}(1, N_{\mathrm{tot}})$	**min**	$\boldsymbol{Q_{25}}$	$\boldsymbol{Q_{50}}$	$\boldsymbol{Q_{75}}$	**max**
$\mathcal{F}$	134.78	190.27	216.29	248.35	300.96
$\mathcal{C}$	129.78	182.38	209.78	238.84	290.57
$\mathcal{P}_{\Sigma}$	112.94	158.07	184.34	215.42	252.62

Chapter 7

Conclusion and Outlook

In this thesis, probabilistic prediction method for estimating the future range – in terms of spatial position – of vehicles on country roads which incorporates topological factors such as visible range, slope, curvature and maximum curvature within the visible range was developed.

The prediction itself was implemented using Bayesian networks to determine the influence factors of the single topological attributes. It was shown that especially the consideration of the curvature of the road in combination with the sign of the slope has a positive influence on the prediction results. This holds especially for longer prediction horizons.

Furthermore, various identification strategies were followed to gain insight into how the "richness" of identification data – which is separately defined within the paper – influences the prediction performance. In this course, it could be seen that the "richness" and also the volume of the identification dataset influences the quality of the prediction models. Models incorporating all measurable topological attributes achieve the least conservative prediction, and the highest accuracy for longer prediction horizons.

Finally, the Bayesian network model was used for an Adaptive Cruise Controller (ACC) in simulation, where it – compared to other prediction approaches – improved both comfort and fuel economy.

Summing up, it can be said that the consideration of topological attributes of country roads for the prediction of driving behavior yields in more accurate and less conservative predictions compared to approaches considering only first principles.

In future works, it is planned to use the method for a safety application as presented in [5] which additionally considers possible lane changes of a driver, and therefore requires a realistic prediction of surroundings for maneuver validation.

Bibliography

[1] STATISTIK AUSTRIA. *Strassenverkehrsunfälle mit Personenschaden*. 2020. URL: https://www.statistik.at/web_de/statistiken/energie_umwelt_innovation_mobilitaet/verkehr/strasse/unfaelle_mit_personenschaden/index.html.

[2] Ekim Yurtsever et al. "A survey of autonomous driving: Common practices and emerging technologies". In: *IEEE access* 8 (2020), pp. 58443–58469.

[3] Hans Andersen et al. "Trajectory optimization for autonomous overtaking with visibility maximization". In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2017, pp. 1–8.

[4] Daniel Adelberger, Meng Wang, and Luigi del Re. "Decision making through stochastic maneuver validation for overtaking on country roads". In: *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE. 2020, pp. 3487–3493.

[5] Daniel Adelberger, Gunda Singer, and Luigi del Re. "A Topology Based Virtual Co-Driver for Country Roads". In: *2022 American Control Conference (ACC)*. Accepted. IEEE. 2022.

[6] Haibo Zhou et al. "Evolutionary V2X technologies toward the Internet of vehicles: Challenges and opportunities". In: *Proceedings of the IEEE* 108.2 (2020), pp. 308–323.

[7] Jianwei Gong et al. "Decision-making model of overtaking behavior for automated driving on freeways". In: *2016 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. IEEE. 2016, pp. 1–6.

[8] S. Söntges and M. Althoff. "Computing the Drivable Area of Autonomous Road Vehicles in Dynamic Road Scenes". In: *IEEE Transactions on Intelligent Transportation Systems* 19.6 (2018), pp. 1855–1866.

[9] Christina Miller, Christian Pek, and Matthias Althoff. "Efficient Mixed-Integer Programming for Longitudinal and Lateral Motion Planning of Autonomous Vehicles". In: *Proc. of the IEEE Intelligent Vehicles Symposium*. 2018, pp. 1954–1961.

[10] S. Manzinger and M. Althoff. "Tactical Decision Making for Cooperative Vehicles Using Reachable Sets". In: *Proc. of the 21st IEEE International Conference on Intelligent Transportation Systems*. 2018.

[11] Ngoc Anh Nguyen et al. “Autonomous overtaking using stochastic model predictive control”. In: *2017 11th Asian Control Conference (ASCC)*. IEEE. 2017, pp. 1005–1010.

[12] Dominik Moser et al. “Risk functions oriented autonomous overtaking”. In: *2017 11th Asian Control Conference (ASCC)*. IEEE. 2017, pp. 1017–1022.

[13] Guotao Xie et al. “Vehicle trajectory prediction by integrating physics-and maneuver-based approaches using interactive multiple models”. In: *IEEE Transactions on Industrial Electronics* 65.7 (2017), pp. 5999–6008.

[14] Stefan Reisinger, Daniel Adelberger, and Luigi del Re. “A two-layer switching based trajectory prediction method”. In: *European Journal of Control* (2021).

[15] Stephanie Lefevre, Dizan Vasquez, and Christian Laugier. “A survey on motion prediction and risk assessment for intelligent vehicles”. In: *Robomech Journal* 1 (July 2014).

[16] Jason Kong et al. “Kinematic and dynamic vehicle models for autonomous driving control design”. In: June 2015, pp. 1094–1099.

[17] Daniel Adelberger and Luigi del Re. “A two-layer predictive emergency steering and escape assistant”. In: *2020 American Control Conference (ACC)*. 2020, pp. 4849–4855.

[18] Samer Ammoun and Fawzi Nashashibi. “Real time trajectory prediction for collision risk estimation between vehicles”. In: *2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing*. 2009, pp. 417–422.

[19] Quan Tran and Jonas Firl. “Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression”. In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. 2014, pp. 918–923.

[20] Stephanie Lefevre et al. “Lane Keeping Assistance with Learning-Based Driver Model and Model Predictive Control”. In: 2014.

[21] Georges S. Aoude et al. “Threat assessment design for driver assistance system at intersections”. In: *13th International IEEE Conference on Intelligent Transportation Systems*. 2010, pp. 1855–1862.

[22] M. Brand, N. Oliver, and A. Pentland. “Coupled hidden Markov models for complex action recognition”. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1997, pp. 994–999.

[23] Dominik Moser et al. “Flexible spacing adaptive cruise control using stochastic model predictive control”. In: *IEEE Transactions on Control Systems Technology* 26.1 (2017), pp. 114–127.

[24] J Holzinger et al. “Context Aware Control of ADAS”. In: *2020 American Control Conference (ACC)*. IEEE. 2020, pp. 2288–2293.

[25] Guotao Xie et al. “A driving behavior awareness model based on a dynamic Bayesian network and distributed genetic algorithm”. In: *International Journal of Computational Intelligence Systems* 11.1 (2018), pp. 469–482.

[26] Fisnik Sulejmani et al. “Autonomous Overtaking Assistant for Country Road Scenarios”. In: *2020 American Control Conference (ACC)*. 2020, pp. 1217–1222.

[27] Kevin Murphy et al. “The bayes net toolbox for matlab”. In: *Computing science and statistics* 33.2 (2001), pp. 1024–1034.

[28] Admir Sovtic, Daniel Adelberger, and Meng Wang. “Probabilistic Prediction of Driving Behavior on Country Roads”. In: *2022 European Control Conference (ECC)*. Accepted. IEEE. 2022.

[29] Daniel Adelberger and Luigi del Re. “Robust Handling of Diffuse Hazard in Country Road Traffic”. In: *2021 American Control Conference (ACC)*. IEEE. 2021, pp. 418–423.

[30] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. Adaptive Computation and Machine Learning series. MIT Press, 2009. ISBN: 9780262013192.

[31] Philipp Polterauer and Luigi Del Re. “Emission constrained fuel optimal vehicle control with route lookahead”. In: *2018 Annual American Control Conference (ACC)*. IEEE. 2018, pp. 5510–5515.

[32] OpenStreetMap contributors. *Planet dump retrieved from https://planet.osm.org*. `https://www.openstreetmap.org`. 2017.

[33] D. Adelberger et al. “Prediction of postprandial glucose excursions in type 1 diabetes using control-oriented process models”. In: *Proceedings of the 11th IFAC Symposium on Biological and Medical Systems (BMS)*. 2021.

[34] D. Swaroop and K.R. Rajagopal. “A review of constant time headway policy for automatic vehicle following”. In: *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*. 2001, pp. 65–69.

[35] Cunxue Wu et al. “Spacing Policies for Adaptive Cruise Control: A Survey”. In: *IEEE Access* 8 (2020), pp. 50149–50162.

[36] E.F. Camacho, C. Bordons, and C.B. Alba. *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing. Springer London, 2004. ISBN: 9781852336943. URL: `https://books.google.at/books?id=Sc1H3f3E8CQC`.

[37] J.B. Rawlings, D.Q. Mayne, and M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2017. ISBN: 9780975937730.

[38] Dominik Moser. “Predictive vehicle control for stochastic risk constrained advanced driver assistance systems”. PhD thesis. Johannes Kepler Universität Linz, 2017.

[39] Gunda Obereigner, Philipp Polterauer, and Luigi del Re. “A Two-Layer Approach for Ecodriving under Traffic”. In: *2020 American Control Conference (ACC)*. IEEE. 2020, pp. 2282–2287.

[40] Johan Lofberg. “YALMIP: A toolbox for modeling and optimization in MATLAB”. In: *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*. IEEE. 2004, pp. 284–289.

[41] Berwin A Turlach and Andreas Weingessel. “quadprog: Functions to solve quadratic programming problems”. In: *CRAN-Package quadprog* (2007).

[42] Gunda Singer et al. “Tire Particle Control with Comfort Bounds for Electric Vehicles”. In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 2046–2052.

List of Figures

List of Tables