Driver Behavior Analysis and Decision-Making for Autonomous Driving at Non-Signalized Inner City Intersections

Zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für

Elektrotechnik und Informationstechnik

des Karlsruher Instituts für Technologie (KIT)

angenommene

DISSERTATION

von

Hannes Weinreuter, M.Sc.

geb. in Karlsruhe

Tag der mündlichen Prüfung:	20. September 2024
Hauptreferent:	Prof. Dr.-Ing. Michael Heizmann, KIT
Korreferentin:	Prof. Dr.-Ing. Dipl.-Psych. Barbara Deml, KIT

Preface

The present thesis originates from my time at the Institute of Industrial Information Technology (IIIT) at the Karlsruhe Institute of Technology (KIT) from June 2017 to June 2023. I owe a debt of gratitude to several people who supported me in various ways.

First, I would like to thank Prof. Dr.-Ing. Fernando Puente León for giving me the chance to work on my dissertation under his supervision during the first three years. I will remember him as a good mentor and a dedicated researcher. I am further grateful to Prof. Dr.-Ing. Michael Heizmann who took over the supervision for the remainder of my time at the institute. Despite the challenging and labor-intensive period for the entire institute, he always had time for good advice and helpful feedback. I would also like to thank Prof. Dr.-Ing. Dipl.-Psych. Barbara Deml for acting as the co-supervisor of my dissertation and her interest in my work.

I thank the permanent staff at the institute for their support in all administrative and technical aspects. My colleagues at the institute made sure there was always a good atmosphere. I remember the many fruitful discussions about our research topics but also the good times we had outside of work. I thank them for that as well as their willingness to proof-read the manuscript of this thesis. The students who wrote their Bachelor's and Master's theses under my supervision contributed to this work with their analyses and discussions.

Finally, I would like to thank my parents for proof-reading this thesis and most importantly, for their continued support to this point.

Karlsruhe, October 2024 Hannes Weinreuter

Zusammenfassung

Der Fokus dieser Arbeit liegt auf dem menschliches Fahrverhalten im Straßenverkehr. Davon werden zwei Aspekte abgedeckt: Dessen Vorhersage, einschließlich der Identifikation relevanter Einflussfaktoren, sowie die Verhaltensgenerierung für autonome Fahrzeuge.

Die Verhaltensvorhersage basiert auf einer Feldstudie, bei der Probanden mit einem Messfahrzeug innerstädtische Kreuzungen durchfahren haben. Die gefahrene Trajektorie und Lidar-Aufnahmen werden dazu genutzt, um Komplexitätseigenschaften zu definieren, die die Umgebung an der Kreuzung, den Verkehr dort sowie den Fahrweg beschreiben. Das Fahrverhalten wird über weitere Eigenschaften charakterisiert. Basierend auf den Komplexitätseigenschaften werden Regressionsmodelle trainiert, um die Verhaltenseigenschaften vorherzusagen. Dazu werden lineare Regression, Random Forest und Gradient Boosting Machine genutzt. Verschiedene Mengen der Komplexitätseigenschaften, inklusive einiger, die durch einen Autoencoder reduziert werden, werden für die Vorhersage verwendet. Die Ergebnisse zeigen, dass das Fahrverhalten zuverlässig vorhergesagt werden kann. Werden jedoch Komplexitätseigenschaftsmengen mit nur wenigen Eigenschaften verwendet, ist die Vorhersageleistung reduziert.

Um einen Komplexitätswert zu erhalten, der sich mit der menschlichen Wahrnehmung von Komplexität deckt, wurde eine Onlinestudie durchgeführt, bei der Videos von Anfahrten an Kreuzungen genutzt wurden. In Paarvergleichen waren die Teilnehmer aufgefordert, die komplexere Situation zu identifizieren. Anhand dieser Daten werden Komplexitätswerte für die in der Studie beinhalteten Kreuzungsdurchfahrten berechnet. Mehrere Methoden kommen zum Einsatz, um diese Werte den Durchfahrten der ursprünglichen Feldstudie zuzuweisen. Diese zugeordneten Komplexitätswerte werden benutzt, um Verhaltens-Regressionsmodelle zu trainieren. Die Ergebnisse zeigen, dass die Verhaltensvorhersage mit den Komplexitätswerten möglich ist, jedoch erfordern es die meisten

Varianten, dass zusätzlich die Abbiegerichtung als zweite Eigenschaft berücksichtigt wird.

Die Verhaltensgenerierung zur Entscheidungsfindung an T-Kreuzungen basiert auf einem ereignisdiskreten System. Für dieses werden mehrere Eigenschaften genutzt, um Ereignisse zu definieren, welche den Status des Entscheidungsfindungsprozesses an der Kreuzung beschreiben. Die Ereignisse lösen die Übergänge zwischen den Zuständen des ereignisdiskreten Modells aus. Allen Zuständen ist entweder offensives oder defensives Fahrverhalten zugewiesen, welches über das Intelligent Driver Model implementiert wird. Der Algorithmus wird über eine Simulationsumgebung validiert. Unter Nutzung einer generischen Karte und mehrerer realer Karten wird das Entscheidungsfindungsmodell 14 400 mal simuliert, während es mit weiteren Kooperationsfahrzeugen interagiert. Keiner dieser Durchläufe führte zu einer Kollision des Fahrzeugs, das den Algorithmus ausführte, und die Zeiten zur Durchquerung der Kreuzung können mit der Anzahl von Kooperationsfahrzeugen sowie den Kreuzungsformen erklärt werden. Weitere Simulationen dienen dazu, den Einfluss eingeschränkter Sicht an den Kreuzungen auf das Modell zu untersuchen.

Abstract

The focus of this work is on human driving behavior in road traffic. Two aspects of it are covered, the prediction of it, including the identification of relevant influencing factors, as well as the behavior generation for autonomous vehicles.

The behavior prediction is based on a field study during which participants drove a measurement vehicle through inner-city traffic. Using the driven trajectories and lidar recordings complexity features to describe the surroundings at the intersection, the traffic there and the driving path are defined. The driving behavior is characterized by further features. Based on the complexity features regression models are trained to predict the behavior features. For that, linear regression, random forest and gradient boosting machine are utilized. Different complexity feature sets, including ones that are reduced with the help of an autoencoder, are used for prediction. The results show that the driving behavior can be predicted reliably. However, when using complexity feature sets with only few features the prediction performance is reduced.

In order to obtain a complexity score that is in line with human perception of complexity, an online study using videos of approaches to intersections was conducted. In pairwise comparisons participants were asked to identify the more complex situation. From that data complexity scores for the intersection passes included in the study are calculated. Several methods are used to assign these scores to the runs of the original field study. Behavior regression models are trained using these assigned complexity scores. The results show that behavior prediction with the complexity scores is possible, however, most variants require to also consider the turning direction as a second feature.

The behavior generation for decision-making at T-intersections is based on a discrete event system (DES). For it, several features are used to define events that describe the status of the decision-making process at the intersection. The events trigger the transitions between the states of the

DES. All states are associated with either offensive or defensive driving behavior, which is implemented using the intelligent driver model. The algorithm is validated with a simulation framework. Using a generic map and several real maps, the decision-making model is simulated 14 400 times while interacting with further cooperation vehicles. None of these runs resulted in a collision involving the vehicle running the algorithm and the times to pass the intersection can be explained by the numbers of cooperation vehicles and the intersection layouts. Further simulations are used to investigate the influence of limited visibility at the intersections on the model.

Contents

Nomenclature

Common Abbreviations

Abbreviation	Description
AE	autoencoder
AV	(general) autonomous vehicle
A-V	vehicle running the decision-making algorithm
A-V_C	vehicle running the simplified algorithm
BT	Bradley-Terry model
B-V	vehicle that can block the exit lane
B-V_C	vehicle that can block the exit lane (of the A-V_C)
C-V	cooperation vehicle
CZ	collision zone
DES	discrete event system
DL	deadlock
GNSS	global navigation satellite system
IDM	intelligent driver model
IMU	inertial measurement unit
lidar	light detection and ranging
LR	linear regression
LSP	last stop point
L-V	leading vehicle (driving in front of the A-V)
L-V_C	leading vehicle (driving in front of the A-V_C)
PN	Petri net
P-V	vehicle with priority
P-V_C	vehicle with priority (seen from the A-V_C)
RBL	*right before left* rule for priority at intersections
RF	random forest
RMSE	root mean squared error

Abbreviation	Description
ROS	robotic operating system
SLAM	simultaneous localization and mapping
StVO	Straßenverkehrsordnung (German traffic laws)
TB	tree-based boosting
TCT	traffic conflict technique
V2I	vehicle to infrastructure communication
V2V	vehicle to vehicle communication
V2x	vehicle to anything communication
Y-V	vehicle that has to yield
Y-V_C	vehicle that has to yield (seen from the A-V_C)

Symbols

Latin Letters

Symbol	Description
a	acceleration
a_{a}^{x}	assumed acceleration of vehicle x
$a_{\mathrm{IDM}}^{\mathrm{a}}$	acceleration calculated using IDM
$\mathcal{D}$	data set
$\mathcal{D}^{\mathrm{F}}$	field study data set (all runs at the T-intersections)
$\mathcal{D}^{\mathrm{V}}$	video study data set
$\mathcal{D}_{\mathrm{iv}}^{\mathrm{F}}$	runs from $\mathcal{D}^{\mathrm{F}}$ with a corresponding intersection/p_{e} combination in $\mathcal{D}^{\mathrm{V}}$
$\mathcal{D}_{\mathrm{ov}}^{\mathrm{F}}$	runs from $\mathcal{D}^{\mathrm{F}}$ without a corresponding intersection/p_{e} combination in $\mathcal{D}^{\mathrm{V}}$
d	distance
d_{b}^{x}	distance to stop for vehicle x
d_{c}	commit distance
$d_{\mathrm{c},x_{\mathrm{c}},\mathrm{b}}^{x}$	distance to CZ start (of the A-V and C-V x_{c}) for vehicle x
$d_{\mathrm{c},x_{\mathrm{c}},\mathrm{e}}^{x}$	distance to CZ end (of the A-V and C-V x_{c}) for vehicle x
$d_{\mathrm{f}}^{\mathrm{b}}$	free distance behind the B-V
$d_{\mathrm{i}}^{\mathrm{b}}$	distance of the B-V from intersection end

Symbol	Description
d_{l}^{x}	distance to last stopping point for vehicle x
d_{s}^{x}	distance to scenario of vehicle x
d_{v}	visibility distance
$d_{\mathrm{v,c}}$	visibility distance (point clouds; all directions)
$d_{\mathrm{v,c}}^{\mathrm{y}}$	visibility distance (point clouds; yield directions)
$d_{\mathrm{v,p}}$	visibility distance (polygons; all directions)
$d_{\mathrm{v,p}}^{\mathrm{y}}$	visibility distance (polygons; yield directions)
d_{vs}	distance of visibility edge to curb (decision-making)
Δd	distance to the L-V
E_i	Elo score of object i
e_x	event x of the DES for decision-making of the A-V; all events are listed in table 4.2
$e_{x,\mathrm{c}}$	event x of the DES for decision-making of the A-$\mathrm{V_C}$; all events are listed in table 4.5
$e_{\mathrm{b}x}$	base event x of the DES for decision-making of the A-V; all base events are listed in table 4.1
$e_{\mathrm{b}x,\mathrm{c}}$	base event x of the DES for decision-making of the A-$\mathrm{V_C}$; all base events are listed in table 4.4
$\mathcal{F}$	set of intersection features
$\mathcal{F}^{\mathrm{V}}$	set of intersection features for with video study data set
$\mathcal{F}^{\mathrm{F}}$	set of intersection features for field study data set
$\hat{f}^{b}$	resulting regression function
$\hat{f}_{\mathrm{boost}}$	regression function of boosting method
N_{cv}	number of C-Vs
N_{d}	number of elements in the data set
N_{s}	number of simulation runs
N_{sc}	number of simulation runs per combination
N_{t}	number of decision trees
N_{tr}	number of elements in the training set
n_{gw}	number of vehicles that have to yield
n_{p}	number of visible pedestrians
n_{rw}	number of vehicles that have the right of way
n_{t}	number of trees around the intersection
n_{v}	number of visible vehicles
n_{vi}	number of interaction vehicles

Symbol	Description
$\mathcal{P}_\mathrm{b}$	hand-drawn polygon set of buildings
$\mathcal{P}_\mathrm{t}$	hand-drawn polygon set of trees
p_e	entry position
p_t	turning direction
$\mathbf{p}_{\mathrm{ref},i}$	reference point i for visibility calculation
$\mathbf{r}$	residual used for boosting method
s_{10}	DES state: A-V drives freely in zone 1
s_{21}	DES state: A-V drives offensively in zone 2
s_{22}	DES state: A-V drives defensively in zone 2
s_{31}	DES state: A-V drives offensively in zone 3
s_{32}	DES state: A-V drives defensively in zone 3
s_{41}	DES state: A-V drives offensively in zone 4
s_{42}	DES state: A-V drives defensively in zone 4
s_{51}	DES state: A-V drives offensively in zone 5
s_{52}	DES state: A-V drives defensively in zone 5
s_{53}	DES state: A-V drives offensively after stopping in zone 5
s_{60}	DES state: A-V drives freely in zone 6
$s_{10,\mathrm{c}}$	DES state: A-$\mathrm{V_C}$ drives freely in zone 1
$s_{21,\mathrm{c}}$	DES state: A-$\mathrm{V_C}$ drives offensively in zone 2
$s_{22,\mathrm{c}}$	DES state: A-$\mathrm{V_C}$ drives defensively in zone 2
$s_{31,\mathrm{c}}$	DES state: A-$\mathrm{V_C}$ drives offensively in zone 3
$s_{32,\mathrm{c}}$	DES state: A-$\mathrm{V_C}$ drives defensively in zone 3
$s_{41,\mathrm{c}}$	DES state: A-$\mathrm{V_C}$ drives offensively in zone 4
$s_{42,\mathrm{c}}$	DES state: A-$\mathrm{V_C}$ drives defensively in zone 4
$s_{43,\mathrm{c}}$	DES state: A-$\mathrm{V_C}$ drives offensively after stopping in zone 4
$s_{50,\mathrm{c}}$	DES state: A-V drives freely in zone 5
$\mathcal{T}$	training set
t	time
$t^x_{\mathrm{c},\mathrm{x_c},\mathrm{b}}$	time to CZ start (of the A-V and the C-V x_c) for vehicle x
$t^x_{\mathrm{c},\mathrm{x_c},\mathrm{e}}$	time to CZ end (of the A-V and the C-V x_c) for vehicle x
t_d	time to drive through an intersection
$\bar{t}_\mathrm{d}$	average time to drive through an intersection
$\bar{t}^0_\mathrm{d}$	average time to drive through an intersection without C-Vs
t^a_w	time the A-V has waited at the intersection
t^p_w	time the P-V has waited at the intersection

Symbol	Description
V_f	free space at the intersection
V_int	considered space around intersection
v	velocity
v_d	velocity drop
v_m	minimum velocity
Δv	velocity difference to the L-V
$\mathbf{w}_\mathrm{a}$	available width vector
$w_\mathrm{a,b}$	available width before intersection
$w_\mathrm{a,e}$	available width after intersection
$\mathbf{w}_\mathrm{s}$	street width vector
$w_\mathrm{s,b}$	street width before intersection
$w_\mathrm{s,e}$	street width after intersection
$\mathbf{w}_\mathrm{v}$	visible width vector
$w_\mathrm{v,b}$	visible width before intersection
$w_\mathrm{v,e}$	visible width after intersection
$\mathbf{X}$	matrix of features
$\mathbf{x_j}$	vector of feature j
x_{ij}	element i of feature j
$\mathbf{y}$	label vector
y_i	label of element i

Greek Letters

Symbol	Description
α_s	minimum angle between two streets
$\boldsymbol{\beta}$	parameter vector for LR
$\hat{\boldsymbol{\beta}}$	estimated parameter vector for LR
λ	weight parameter for lasso/ridge variants of LR
λ_b	factor for boosting trees
π_i	Bradley-Terry score of object i
π_{ij}	probability that object i is preferred to object j (based on their Bradley-Terry scores)
$\tilde{\pi}_i$	estimated Bradley-Terry score of object i

Superscripts

Symbol	Description
$(\bullet)^{\mathrm{a}}$	features of the A-V
$(\bullet)^{\mathrm{b}}$	features of the B-V
$(\bullet)^{\mathrm{l}}$	features of the L-V
$(\bullet)^{\mathrm{p}}$	features of the P-V
$(\bullet)^{\mathrm{y}}$	features of the Y-V

Subscripts

Symbol	Description
$(\bullet)_{(\bullet),\mathrm{c}}$	features of the cooperation vehicle algorithm
$(\bullet)_{\mathrm{In}}$	BT/Elo scores from direct assignment
$(\bullet)_{\mathrm{RF}}$	BT/Elo scores from RF model
$(\bullet)_{\mathrm{LR}}$	BT/Elo scores from LR model
$(\bullet)_{\mathrm{LR,r}}$	BT/Elo scores from model with ridge variant of LR
$(\bullet)_{\mathrm{LR,l}}$	BT/Elo scores from model with lasso variant of LR

Parameters

Symbol	Description	Value	Unit
a_{c}	comfort deceleration	–2.5	$\mathrm{m\,s^{-2}}$
a_{e}	emergency deceleration	–7.5	$\mathrm{m\,s^{-2}}$
a_{h}	hard deceleration	–4.5	$\mathrm{m\,s^{-2}}$
a_{m}	max. acceleration	2.5	$\mathrm{m\,s^{-2}}$
d_{E}	divisor for Elo score calculation	400	
$d_{\mathrm{l,o}}$	stop distance before an LSP	1	m
d_{min}	min. distance for a following vehicle	1.5	m
d_{n}	near the intersection (threshold in d_{s}^{x})	12	m
d_{o}	distance offset to avoid toggling	0.2	m

Symbol	Description	Value	Unit
Δd^{P}	min. lead dist. to drive before the P-V	10	m
$\Delta d_{\mathrm{c}}^{\mathrm{P}}$	lead distance threshold for the A-V_C to drive rel. to the P-V_C	–15	m
$\Delta d_{\mathrm{s,c}}^{\mathrm{P}}$	lead dist. threshold for the A-V_C to drive rel. to the P-V_C (special behavior)	7	m
E_0	initial Elo score	1600	
k_{E}	factor for Elo score calculation	10	
l_{v}	vehicle length	4.4	m
t_{min}	min. lead time for a following vehicle	1.2	s
t_{r}	reaction time	1	s
t_{y}	waiting time threshold for the P-V	2	s
Δt^{P}	minimum lead time to drive rel. to the P-V	2.5	s
$\Delta t_{\mathrm{c}}^{\mathrm{P}}$	lead time threshold for the A-V_C to drive rel. to the P-V_C	–2.5	s
$\Delta t_{\mathrm{c}}^{\mathrm{Y}}$	lead time threshold for the A-V_C to drive rel. to the Y-V_C	3	s
$\Delta t_{\mathrm{s,c}}^{\mathrm{P}}$	lead time threshold for the A-V_C to drive rel. to the P-V_C (special behavior)	3	s
$\Delta t_{\mathrm{s,c}}^{\mathrm{Y}}$	lead time threshold for the A-V_C to drive rel. to the Y-V_C (special behavior)	–3	s
v_{a}	assumed velocity of the A-V (straight)	6.5	$\mathrm{m\,s^{-1}}$
v_{a}	assumed velocity of the A-V (turning)	4.0	$\mathrm{m\,s^{-1}}$
v_{max}	speed limit	30	$\mathrm{km\,h^{-1}}$
v_{s}	velocity threshold for a stopped vehicle	0.15	$\mathrm{m\,s^{-1}}$
$v_{\mathrm{s,c}}$	v_{s} for the A-V_C algorithm	0.3	$\mathrm{m\,s^{-1}}$
v_{sl}	velocity threshold for a slow vehicle	2	$\mathrm{m\,s^{-1}}$
$v_{\mathrm{sl,c}}$	v_{sl} for the A-V_C algorithm	1	$\mathrm{m\,s^{-1}}$
v_{t}	target velocity of the A-V (dependent on state, see table 4.3)		
$v_{\mathrm{t,c}}$	target velocity of the A-V_C (dependent on state, see table 4.6)		
w_{v}	vehicle width	1.8	m

1 Introduction

Automatic driving is a promising technical development for a multitude of reasons. Shifting the driving task from humans to machines offers several advantages, the most important of which certainly is the potential for higher safety on the roads and therefore fewer accidents, especially including those with fatal outcomes or severe injuries. Automated vehicles (AV) will very likely only be admitted onto the roads if they are significantly safer than human drivers. From that one can conclude that traffic will become safer with the introduction of such vehicles. The urgent need for safer streets is highlighted by the accident statistics. In 2022 there were 2788 [3] people killed in traffic in Germany alone. Many more were severely injured (57 727 people in 2022 [2]). This number has been decreasing over the last decades from its peak of 19 193 people who died in traffic in West Germany alone in 1970 [3]. Schulze and Koßmann [142] attribute this decrease to the introduction of many safety measures like the introduction of a speed limit on rural roads, the requirement to use seat belts, the introduction of maximum blood alcohol levels and many safety features for cars and infrastructure alike. The authors, however, also note that a clear causation cannot be proven due to the lack of studies. Given that an autonomous vehicle does not get tired or bored and is not affected by its current mood, its introduction has the potential to be another safety feature to further reduce the number of people affected by traffic accidents.

But there are aspects other than improving safety on the roads that make AVs attractive. Vehicles that are fully autonomous and do not require any input by their passengers[1] will likely give more people access to individual mobility than is currently the case. This includes people who do not know how to drive, elderly people, people with disabili-

[1] Other than the input of the desired destination.

ties, young people who have not reached the minimum age to acquire a driver's license or people who choose not to drive themselves. This is especially relevant for rural regions with poor coverage of public transport. An ecological aspect of the introduction of AVs is their potential to reduce the energy consumption needed for driving due to more efficient accelerating and braking. This technology is also very attractive from an economical viewpoint. Transport services might become cheaper as fewer personnal is needed for their operation and e.g. trucks could be run for longer parts of a day as currently drivers need rest periods in which the vehicle is stationary. This aspect could also benefit public transport as routes with smaller demand could also be served and routes could be run more regularly due to the lower costs of operation. However, there are also some problematic aspects of autonomous vehicles. The cost of AVs is likely going to be higher than a conventional vehicle due to the need to include a large number of sensors and computing hardware. Further, it may also make public transport less attractive because the time in one's own vehicle can be spent more productively. Gruel and Stanford [60] give a systematic overview on potential consequences of the introduction of autonomous vehicles.

Development of and research into autonomous vehicles has been an active field for some time now. In 2004 and 2005 the *Defense Advanced Research Projects Agency* (DARPA) hosted the *DARPA Grand Challenge* where teams were tasked to drive autonomous vehicles through a desert course [150]. In 2007 a similar competition was held, the *DARPA Urban Challenge* [113]. In it autonomous vehicles had to travel through an urban scenario in a test environment. They had to interact with the given street layout and other traffic participants. More recently, many universities and companies have created vehicles that are able to drive in real traffic, although still with a safety driver for emergencies. One prominent example is the Bertha vehicle [177]. In 2013 it autonomously completed the 103 km long route that Bertha Benz had completed 125 years before. Levinson et al. [89] present the autonomous test vehicle of Stanford University. Manufacturers increasingly equip their models with systems that support the driver but do not allow autonomous driving yet, i.e. the driver is still responsible. A prominent example is the *Autopilot* function of the company *Tesla*. This system, however, is also faced with some criti-

cism [110, 149]. A step closer to full autonomy is the *Drive Pilot* function of the company *Mercedes-Benz*, which allows drivers to fully delegate the driving task to the vehicle in some scenarios on highways where the velocities are not too large [54, 56]. In San Francisco the company *Waymo* is commercially operating a fleet of fully autonomous vehicles as taxis [111].

1.1 Problem Statement

Among the many challenges that come with the development and introduction of automatic driving, this thesis focuses on non-signalized inner-city intersections. At intersections many driving paths overlap each other. Therefore vehicles driving along these intersecting paths are in conflict with each other and cannot drive simultaneously. Instead, a driving order has to be established. In the year 2022 there were 342 852 accidents in Germany in which at least one person was injured or even killed. Of these, 45 857 were caused by non-compliance to the right of way [1]. This highlights the importance of improving safety in situations like intersections where the right of way is crucial for safe driving.

Even after automated vehicles are introduced, it is reasonable to assume that traffic for the foreseeable future will be made up of a mixture of automated vehicles and vehicles driven by humans. For that reason, throughout this thesis the current state of all traffic regulations is assumed to be in force, i.e. the regulations are not adopted to enable more efficient driving by automated vehicles. This assumption is in contrast to many works from literature that assume new and more efficient rules; especially those which deal with decision-making for autonomous vehicles. Section 4.1 contains a detailed literature review on that topic.

If the priority at intersections is regulated by traffic lights, the decision-making for automated vehicles is straight forward, i.e. it can enter the intersection if it has green light and the traffic allows the vehicle to not get stuck within the intersection. One can further reasonably expect human drivers to behave similarly and an automated vehicle will, in the absence of emergencies, be able to drive through the intersection with these simple rules. In Germany intersections can further be regulated by traffic signs that give priority to drivers on certain roads while

drivers on the remaining roads are signalized to yield. In case there are no traffic lights or signs at an intersection, the German traffic regulations (Straßenverkehrsordnung, StVO[2]) give priority to vehicles entering the intersection on the street to the right of one's own street[3]. This is commonly referred to as the *right before left* (RBL) rule. Also, oncoming traffic has priority over turning left[4].

The reasons for not using traffic lights at intersections with low or medium traffic volumes are obvious: First, the installation and maintenance of traffic lights are certainly costly and second, if there is a traffic light at an intersection with low traffic volumes, it is likely that vehicles will be waiting at a red light even though there is no conflicting traffic. However, regulating traffic at intersections without traffic lights also has some drawbacks. Drivers are forced to make the decision on when to drive through the intersection on their own. This can lead to unclear and potentially dangerous situations especially in the case when a driver (or an AV) has to decide whether one can drive before another vehicle (that has priority over oneself) in order to cross a traffic stream or to merge into it. The traffic regulations do not regulate this explicitly – which would probably not be feasible for human drivers anyway – and instead only state that drivers who have to wait have to adapt their driving behavior in time, especially by braking. It further instructs drivers to not endanger or significantly obstruct others who have the right of way[5].

This may become risky if a driver misjudges a gap or tries to force a driver who has priority to wait. Such situations can occur if priority signs are used as well as if priority is regulated by the RBL rule. These critical gaps, i.e. the minimum size of a gap before a vehicle with priority arrives so that a driver who enters the intersection from the yielding street can still drive first, have been the focus of research for some time, see e.g. Ashton [15] or McGowen and Stanley [107]. Toledo [151] provides an overview on that topic. A further aspect that complicates driving at signalized and non-signalized (and thus RBL) intersections are drivers who

[2] https://www.gesetze-im-internet.de/stvo_2013/BJNR036710013.html (in German, accessed 29 Dec. 2023)
[3] § 8 Abs. 1 StVO
[4] § 9 Abs. 3 StVO
[5] § 8 Abs. 2 StVO

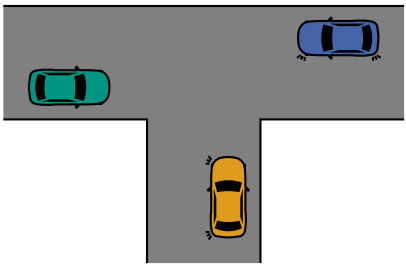


Figure 1.1 Deadlock situation at a T-intersection.

do not follow the traffic regulations. This further increases the demands on humans and AVs alike who have to interact with such drivers.

Finally, deadlock scenarios can occur at non-signalized intersections; these are situations in which no driver has priority over all other drivers and therefore all drivers have to wait according to the RBL rule and the rule that gives priority to driving straight over turning left. An example of that scenario at a T-intersection is shown in figure 1.1. There, the yellow vehicle has to yield to the blue vehicle entering from the right; the blue vehicle has to give way to the green vehicle on the street from the left. Finally, the green vehicle has to yield to the yellow vehicle. As this situation cannot be resolved using only the RBL rule, one of the drivers has to give up his/her right of way. The German traffic regulations[6] state that one has to give up one's right of way if the situation requires it. Also, one may only accept the waiver of the right of way by a cooperation partner after this decision has been communicated between the drivers.

Non-signalized intersections are common in Germany, especially in residential areas with low or medium traffic density. They are therefore an important aspect of daily driving in Germany. Given these aspects, the understanding of human driving behavior at such intersections is very important for the eventual introduction of automatic vehicles onto the roads. Also, automated vehicles must be able to safely interact with human drivers in these situations, i.e. their decision-making policy has to be able to interpret human driving behavior and to communicate

[6] § 11 Abs. 3 StVO

their own intentions to their human counterparts. Therefore, an analysis of human driving behavior at non-signalized inner-city intersections is conducted in this work. This is followed by the introduction of a decision-making model at intersections of this type.

1.2 Contributions

The contributions presented in this thesis have in part been published before [189–192]. The first aspect covered in this work is an **analysis of human driving behavior** at inner-city intersections. The main contributions on that subject are as follows:

- The behavior of human drivers is investigated in a field study that was performed in Karlsruhe, Germany. In that study participants drove through non-signalized inner-city intersections and the driven trajectory as well as the surroundings at the intersections were recorded.
- Based on the data from the study features are defined to describe the surrounding environment at an intersection (visibility conditions, street width, etc.). In this work this is interpreted as the *complexity* of an intersection. Additionally, features to describe the driving behavior are derived from the driven trajectories.
- Using the complexity features, regression models are trained to predict the behavior features. That way the influence of the complexity of an intersection on the driving behavior is investigated. Several variants to reduce the dimensionality of the complexity feature set are evaluated.
- To improve the ability to interpret the concept of intersection complexity, ideally a one-dimensional complexity measure can be defined. To that end, a second study was conducted, in which participants were shown videos of approaches to intersections and were asked to pick the more complex intersection in pairwise comparisons.

- Using established methods for evaluating pairwise comparison data, complexity measures are defined. These are then also used to train regression models to predict the driving behavior.

The second aspect that is covered in detail in this thesis proposes a method for **behavior generation** for autonomous vehicles. This decision-making algorithm does not require any communication between the autonomous vehicle and other vehicles or the infrastructure. Instead, only the observable state of its cooperation partners, i.e. their positions, velocities and accelerations, are needed. The most important contributions of this thesis to that aspect are:

- Within the algorithm the other vehicles at an intersection are categorized (e.g. vehicles that have priority, vehicles that drive in front) and only the closest vehicle to the ego vehicle is taken into consideration, as the remaining vehicles are currently unable to interact with the ego vehicle anyway. With that approach the algorithm scales well with the number of vehicles at an intersection.

- A decision-making model for non-signalized T-intersections is introduced that is based on a discrete event system. The events that trigger the state of the model to change and the features the events are based upon are presented in detail. The state machine, its states and the driving behavior that is associated with these states are introduced.

- A simulation framework is developed to test and evaluate the proposed model. The framework includes a model to simulate cooperation partners and it takes realistic visibility conditions at the intersection into consideration. Also, different maps can be used for the simulations and cooperation vehicles can show driving behavior that deviates from the expected behavior or even the traffic regulations.

- An in-depth evaluation of the decision-making algorithm regarding its safety and efficiency is performed. Also, important properties like the influence of the number of cooperation vehicles and the visibility conditions and different maps are investigated.

1.3 Structure

As is evident from the previous section, this thesis deals with two topics that are somewhat distinct from one another. The first topic is the analysis of human driving behavior, which is presented in chapters 2 and 3. The second topic, the decision-making algorithm for an autonomous vehicle at non-signalized intersections, is described in chapter 4. Given that the thesis is made up of two distinct parts, the literature review is not presented at once but rather the relevant literature for each chapter is given separately at the beginning of that chapter (sections 2.1, 3.1 and 4.1, respectively).

Chapter 2 is on the behavior analysis based on the field study. In it, first the field study is introduced (section 2.2), then the complexity features (section 2.3) and the behavior features (section 2.4) used in this work are presented. The results of the behavior predictions using the complexity features are presented afterwards (section 2.5).

Chapter 3 focuses on the second aspect of the behavior analysis in that it contains the derivation of the complexity measure for intersections. For that purpose, first the study based on the videos is introduced (section 3.2), followed by the derivation of the complexity scores (section 3.3). Next, the results of predicting the driving behavior with the complexity scores are shown (section 3.4).

The second topic of this thesis is the decision-making algorithm for autonomous vehicles; it is introduced in chapter 4. After describing the algorithm itself and its features, events and states (section 4.2), the simulation framework is given (section 4.3). Finally, the results of the simulation runs are presented (section 4.4).

Each of these chapters concludes with a summary and an outlook for potential future work (sections 2.6, 3.5 and 4.5, respectively). In chapter 5 the conclusion of the entire thesis is discussed, again with a summary (section 5.1) and an outlook (section 5.2).

2 Using Intersection Complexity for Behavior Prediction

Understanding why drivers behave the way they do is an important question. It is relevant in general as a better understanding of the reasons for human driving behavior could lead to implications for better road and vehicle design to improve safety. It is also important in the context of autonomous driving because AVs need to be able to interpret human driving behavior in order to react accordingly. A better understanding of typical human driving behavior might further contribute to AVs that behave similar to a human driver if this knowledge is incorporated into the decision-making pipeline. It might make both, passengers of the AV and cooperation partners of an AV, more comfortable if an AV behaves as a human driver would. It would therefore be especially beneficial during the early phases of the deployment of AVs while interactions with them are rare and human traffic participants are not yet used to them. Also, as mixed traffic between AVs and human-driven vehicles will probably be prevalent in the foreseeable future, good cooperation between AVs and conventional vehicles is desirable.

These considerations are true for a wide range of traffic scenarios, but in this thesis the focus is on the driving behavior at non-signalized intersections. As elaborated in the previous chapter, these are challenging to human drivers for several reasons: First, it might be difficult to identify a safe and suitable gap in the traffic stream, especially if the traffic flow on the priority lane is relatively dense. Second, not fully regulated situations can occur and drivers might behave in an unexpected manner or even disregard the traffic regulations. Therefore, the underlying question in this chapter is about the factors that influence a driver's behavioral decisions. These certainly include a driver's personality and the current mood, but these aspects are not considered here. Instead, only the influence of the surroundings at an intersection is investigated.

To this end the concept of intersection complexity is introduced. In this work intersection complexity describes all those aspects of the surroundings at an intersection that influence the driving behavior. One aspect of intersection complexity are features that describe stationary properties of the intersection. This includes the distance at which a driver can see far enough into an intersection, the width of the road, the available, i.e. drivable, width of the road and the visible width that describes the lateral distance to the nearest objects. Also, the free volume at an intersection, the angle at which the streets intersect each other and the number of trees are included. The hypothesis is that a narrow, poorly visible intersection with little free space, many trees and where streets meet at an angle that deviates considerably from 90° is more complex than the opposite. Additionally, it is assumed that denser traffic contributes to increased complexity. Thus the number of vehicles, pedestrians and vehicles that one has to interact with – including priority relations – are added as features. Finally, the desired trajectory through an intersection is considered to influence intersection complexity. The measure by which the influence of intersection complexity is judged in this thesis is its influence on the driving behavior.

This work is based on data gathered during a field study conducted in Karlsruhe, Germany (see section 2.2). In that study participants drove through several non-signalized inner-city intersections. Using a roof-mounted lidar, the surroundings at the intersection were recorded. From that data features are extracted that describe the above-mentioned properties. The combination of these features is regarded as the intersection complexity; they are introduced in detail in section 2.3. The traffic at the intersection is also extracted from the lidar data. The driving behavior is obtained from the driven trajectory, see section 2.4. Using the complexity features, regression models are trained to predict the behavior features. To identify the most relevant of the complexity features, their influence on the prediction is investigated and several methods to train models with a smaller complexity feature set are examined. All of these analyses are presented in section 2.5. A summary is finally given in section 2.6.

The contents of this chapter have been published before [190, 192]. In the present work however, more detailed evaluations are presented.

2.1 Related Work

Many aspects of this chapter have been the focus of previous work. Therefore, a literature review is given in this section. First, an overview on environmental aspects previous works have identified as relevant for the driving behavior is given; also complexity definitions by other authors are introduced. The second aspect that is discussed concerns the methods used to predict driving behavior.

2.1.1 Factors Influencing Driving Behavior

The question of what influences driving behavior is an important one, as elaborated above. External factors are the focus of several previous works. Generally, these factors can be classified into two groups, stationary aspects and dynamic aspects. Stationary features include the infrastructure, vegetation, buildings or parked vehicles, among others; the dynamic environment is mostly caused by other traffic participants. The term *complexity* is not used consistently, authors attribute different aspects to contribute to complexity.

There are several static features of the driving environment that are the focus of scientific studies. A relatively generic definition is used by Imbsweiler et al. [181] who determine T-intersections as more complex than narrow passages because participants in a study showed more defensive behavior in the former case. Similarly, Faure et al. [49] classify highway driving as least complex followed by rural roads and inner-city driving as most complex, because the more complex scenarios coincide with increased levels of mental workload. One can alternatively take parked vehicles at the side of the road into consideration like Edquist et al. [46] or compare intersections that are signalized with those that are not [91]. The driving trajectory also influences the driving task. For that purpose Hancock et al. [61] investigate the difference between driving straight at an intersection and turning left or right there. Image based features can also be used for complexity assessment of intersections; Wijnands et al. [163] automatically identify intersections from satellite images. In their work complex intersections have at least one street with multiple lanes, traffic islands, slip lanes or more than four roads leading into it. Further, visual clutter can be used to assess traffic situations [66].

As an alternative to these static features, others investigate the influence of dynamic properties of the driving task. Patten et al. [123] define three complexity classes; they assign the highest complexity to scenarios with high demand on both information processing and vehicle control. The lowest complexity is assigned to scenarios where both aspects require low demand and a medium class is assigned to scenarios where one of the aspects has high demand while the other only requires low demand. Jahn et al. [74] use the same complexity classes but without the medium complexity case. Teh et al. [148] define complexity by traffic density and the occurrence of lane changes. The density of both pedestrians and vehicles at an intersection can also be used to define complexity [103]. Another traffic feature that has shown an effect on driving behavior is the occurrence of traffic congestion [92]. Werneke and Vollrath [161] define intersection complexity as a combination of traffic density, the presence of a zebra crossing and the number of vehicles coming from the left direction.

Some works consider a combination of static and dynamic features of the surroundings and investigate their influence on the driving behavior. Horberry et al. [67] define complexity by the number of advertisement signs, buildings, oncoming vehicles and further infrastructure while driving on a highway. Cantin et al. [28] consider a straight road as least complex followed by an intersection where a stop is required and assume an overtaking maneuver as most complex within the scope of their study. Oviedo-Trespalacios et al. [122] define complexity by the grade of urbanization, the presence of oncoming traffic, leading traffic and the street geometry (straight road or driving around corners) and investigate the influence of these aspects on the driving behavior of distracted drivers. A similar problem is considered by Zhao et al. [171]; they examine road complexity of off-road environments. Road complexity in their work is determined by features describing the road conditions like the slope, street width or road roughness.

In summary one can conclude that the definition of complexity varies considerably among authors but in general still follows a similar pattern. Aspects that can distract drivers from the driving task or increase its difficulty because the driving task is augmented by additional tasks like interacting with other drivers are considered to make up or at least

contribute to complexity. It is noteworthy, however, that most authors only consider a small number of all potential influences. The usage of the different aspects is certainly justified by the influence they evidently have on the driving behavior; the present work intends to contribute to these findings by considering a large set of features to make up complexity.

Even though the definition of complexity is not consistent among the different authors, the metrics by which the influence of complexity on the driving behavior is assessed are often the same. Mental workload is used in many works [28, 46, 49, 61, 74, 122, 123, 148]; all find that increased complexity corresponds with higher levels of workload. The combination of complexity and a driver's age and their influence on driving behavior is also investigated [28, 66, 67]. In more complex scenarios older drivers drive slower [67]. In comparison to younger drivers their workload is increased in complex environments [28]. Increased age, combined with larger visual clutter, decreases search efficiency of traffic signs [66]. Further, driving is found to be more aggressive after a congestion [92]. The visual scanning behavior differs between signalized and unsignalized intersections [91]. The complexity of intersections further influences the driving behavior and the attention allocation [161]; also drivers feel less safe while driving through T-intersections compared to narrow passages, which are assumed to be less complex [181].

Features to describe the driving behavior of individual drivers in the above works include the lateral positioning on the driving lane [46, 92, 148], the velocity of the drivers [46, 67, 92, 148, 161] or the amount they deviate from the posted speed limit [67, 122]. Further, variants such as maximum or minimum values, mean and standard deviation are used.

A related approach to assessing driving behavior is the traffic conflict technique (TCT) [31, 39, 77]. It is used to assess traffic situations by considering not only data on accidents that occur but also data on conflict situations. For that analysis several measures are commonly used: The time to collision (TTC) [31, 39, 77, 112] is the remaining time until two vehicles collide under the assumption that both continue driving as they currently are. The post encroachment time (PET) [77] describes the time between a first vehicle leaving a conflict zone and a second vehicle entering it. The deceleration to safety (DTS) [39, 77] describes the minimum necessary deceleration to avoid a collision. There exist several extensions

and applications of this concept, e.g. Minderhoud and Bovy [112] extend the definition of the TTC, while Domeyer et al. [39] use several of these measures to investigate the interaction between drivers and pedestrians. As the TCT focuses on conflict scenarios among drivers it is not applicable in this work as the focus is on the influence of external factors on the driving behavior, which is not limited to interactive scenarios or even conflicts with other traffic participants.

2.1.2 Methods for Behavior Prediction

In this chapter the focus is on predicting driving behavior based on features of complexity. Predicting driving behavior is not a new task in literature, there are several works that focus on certain aspects of it. Being able to reliably predict the turning direction of other vehicles at an intersection is important for e.g. decision-making. Zyner et al. [179] do so at T-intersections by using separate Long Short Term Memory (LSTM) networks for each approach direction, while Streubel and Hoffmann [146] use hidden Markov models (HMM) for the prediction. Phillips et al. [126] predict the turning direction at both T- and X-intersections with LSTMs, Multi Layer Perceptrons and Conditional Probability Tables based on a large set of features. Klingelschmitt et al. [80] deal with the similar problem of selecting situation hypotheses including a risk assessment for the observing vehicle. Also, being able to predict if a vehicle drives or yields at an intersection is relevant for many applications. Ward and Folkesson [158] predict whether drivers who have to yield at intersections do so or if they drive before the vehicle with priority. They use k nearest neighbors, a support vector machine and a random forest based on trajectory features.

In this thesis the driving behavior is described by features that assume continuous values (see section 2.4), thus regression needs to be performed. This prediction is based on features that describe intersection complexity, they are introduced in section 2.3. The basis for regression is a data set $\mathcal{D}$ consisting of features $\mathbf{X}$ and labels $\mathbf{y}$:

$$(\mathbf{X}, \mathbf{y}) = (\mathbf{x}_1, \dots \mathbf{x}_m, \mathbf{y}), \quad \mathbf{x}_1, \dots \mathbf{x}_m, \mathbf{y} \in \mathbb{R}^{N_\mathrm{d}}. \tag{2.1}$$

This data set has m features for N_d elements in the data set. In this work the features and labels of a single element are scalar. Typically, only a sub

set $\mathcal{T}$ of a data set $\mathcal{D}$ is used to train a model; the remaining data points are used for validation and testing.

Three methods for regression are used, *linear regression*, *random forests* and *boosting*. All are introduced briefly in the following sections.

2.1.2.1 Linear Regression

Linear regression (LR) finds, as the name suggests, linear dependencies between the features $\mathbf{X}$ and the labels $\mathbf{y}$. The label is the result of a weighted sum of all m features; the general model of LR is as follows [176]:

$$\mathbf{y} = \beta_0 + \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2 + \cdots + \beta_\mathrm{m} \mathbf{x}_m = [\mathbf{1}\,\mathbf{X}]\,\boldsymbol{\beta}. \tag{2.2}$$

In the above equation $\boldsymbol{\beta} = [\beta_0\, \beta_1 \ldots \beta_\mathrm{m}]^\mathrm{T}$ are the parameters of the model and $\mathbf{1}$ is a vector containing the value 1. An estimate $\hat{\boldsymbol{\beta}}$ of the coefficients $\boldsymbol{\beta}$ needs to be found based on the training set $\mathcal{T}$; a least squares approach can be used for that purpose [75]:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N_\mathrm{tr}} \left(y_i - \beta_0 - \sum_{j=1}^{m} \beta_j x_{ij} \right)^2. \tag{2.3}$$

Here, x_{ij} describes the jth feature of the ith data point from the training set $\mathcal{T}$ and N_tr is the number of elements in the training set. The regression model is described by the set of parameters $\boldsymbol{\beta}$ that minimize this equation.

Many extensions of this model exist; e.g. polynomial regression allows to consider powers of features [75]. Two extensions are used in chapter 3; the *lasso* and *ridge* variants of LR. Both add a term to the minimization problem above. In case of the ridge variant of LR this results in [75]:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N_\mathrm{tr}} \left(y_i - \beta_0 - \sum_{j=1}^{m} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{m} \beta_j^2. \tag{2.4}$$

This adds the tuning parameter $\lambda \geq 0$ that controls the impact of the newly added factor $\sum_{j=1}^{m} \beta_j^2$. This factor forces the $\boldsymbol{\beta}$ to become small. Different values of λ result in different models; therefore this value has to be optimized as well.

The lasso version of LR is defined similarly. A term, weighted by λ, is added as well [75]:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N_{\mathrm{tr}}} \left(y_i - \beta_0 - \sum_{j=1}^{m} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{m} \left| \beta_j \right|. \tag{2.5}$$

The lasso variant forces, in contrast to the ridge version of LR, some of the β_j to be zero (for large enough λ [75]), thus excluding the corresponding features from the model. For both, ridge and lasso, cross-validation is used to determine λ.

To perform LR in this work the *MATLAB* function `regress` is utilized. Ridge regression is performed with the function `ridge`, for the lasso variant the function `lasso` is used.

2.1.2.2 Methods Based on Decision Trees

Decision trees recursively split the feature space into smaller subsets. Depending on their feature values, the elements of the training set $\mathcal{T}$ are assigned one of these subsets $\mathcal{T}_{\mathrm{c}}$. In case of a regression tree, each subset is assigned a regression value, typically the average value of the elements of $\mathcal{T}$ that fall into it [62]. At each split, the vector of all values of the feature $\mathbf{x_j}$ and a value x_j^{s}, where the current split is to occur, have to be determined. All elements of the currently considered set $\mathcal{T}_{\mathrm{c}}$ for which $\mathbf{x_j} \leq x_j^{\mathrm{s}}$ holds are sorted into the first subset; the remaining elements, i.e. $\mathbf{x_j} > x_j^{\mathrm{s}}$, are sorted into the second subset of the split [62]. This selection process can be done by minimizing the squared prediction error [62]. Similarly, a classification tree can be generated as well. As classification is not needed here, this is not discussed further.

The advantage of decision trees include their simplicity and interpretability [62], but they have a high variance, i.e. small changes in the data set may result in very different trees [62]. The concept of *bagging* (*bootstrap aggregation*) is used to counteract that property by combining several of these "weak learners" [75]. Using the bagging approach, N_{t} decision trees are trained using equally many subsets of the learning set. These subsets are generated by taking repeated samples from the training set T. The bth decision tree learns the function $\hat{f}^b(\mathbf{x})$ which outputs a

regression value for given feature values $\mathbf{x}$. For the bagging process these N_t regression models are then averaged [75]:

$$\hat{f}_\mathrm{bag}(\mathbf{x}) = \frac{1}{N_\mathrm{t}} \sum_{b=1}^{N_\mathrm{t}} \hat{f}^b(\mathbf{x}). \tag{2.6}$$

To get a random forest (RF) [26] model from that, one additional step has to be taken. When using RF, the training of the trees in the bagging procedure is adapted in that only a randomly selected subset of all p features is considered at each split in the trees [75]. This helps to decorrelate the trees by ensuring that the current subset is not always split along a very strong feature in the first division [75]. In this work the *MATLAB* function `TreeBagger` is used to train RF models. It is set up to additionally output the importance of the features on the regression result. This is done by assessing the amount by which a given feature reduces the residual sum of squares [75], i.e. the squared prediction error. All behavior predictions with RF use $N_\mathrm{t} = 300$ trees and the minimum leaf size (the minimum number of training examples that have to remain in the terminal split of a tree) is set to $N_\mathrm{L} = 5$.

A different approach is used for *boosting* (abbreviated here as *TB*, short for *tree-based boosting* [75]), the second tree-based regression method used here. It is also an ensemble of decision trees, but new trees are trained using the current residuals $\mathbf{r}$ of the training examples instead of the labels $\mathbf{y}$. The training process is described according to James et al. [75]: The regression model is initialized as zero, i.e. $\hat{f}_\mathrm{boost}(\mathbf{x}) = 0$, and the residuals are initialized with the label values ($\mathbf{r} = \mathbf{y}$). The bth tree $\hat{f}^b(\mathbf{x})$ is then added based on the modified training set $(\mathbf{x}, \mathbf{r})$ for $b = 1 \dots N_\mathrm{t}$. After training a new tree, the regression function $\hat{f}_\mathrm{boost}(\mathbf{x})$ is updated to $\hat{f}_\mathrm{boost}(\mathbf{x}) + \lambda_\mathrm{b}\hat{f}^b(\mathbf{x})$ and the residuals $\mathbf{r}$ are set to the new value of $\mathbf{r} - \lambda_\mathrm{b}\hat{f}^b(\mathbf{x})$. λ_b is a parameter that determines the influence of newly added trees on the regression result. After all trees are trained, the final model results in:

$$\hat{f}_\mathrm{boost}(\mathbf{x}) = \sum_{b=1}^{N_\mathrm{t}} \lambda_\mathrm{b}\hat{f}^b(\mathbf{x}). \tag{2.7}$$

The factor λ_b, the maximum number of nodes per tree N_n and the number of trees N_t are the tuning parameters of the model. In this thesis the *MATLAB* function `fitrensemble` is used. The three parameters of

Figure 2.1 Measurement vehicle that was used for the field study. The lidar is mounted on top of the plate on the roof, the IMU is located inside the box below the plate. The two GNSS antennas are placed on the outer ends of the roof railing. The image is a cropped version of the original.

the TB are optimized using the functionality for that implemented in `fitrensemble` based on the data set of the T-intersections with the full feature set and the commit distance as the label; see section 2.5.1 for more details on the features and the training process. These values are used for all models: $\lambda_\mathrm{b} = 0.046\,416$, $N_\mathrm{n} = 11$ and $N_\mathrm{t} = 136$.

2.2 Field Study and Data Processing

The data for this chapter originates from a field study that was conducted in the inner city of Karlsruhe, Germany in the summer of 2020. In the following, first the study itself is introduced, then the steps to process the measurement data into a data set are presented.

In the study participants drove a measurement vehicle through several different non-signalized intersections along a predefined course in Karlsruhe. The measurement vehicle is a 2011 *Volkswagen Passat 2.0 TDI*, which is equipped with an eye-tracking system, that was not used during

the study, and the option to record internal vehicle data. The latter was recorded for some participants but is not evaluated here. The data of the study comes from a 16 channel lidar (*Velodyne VLP-16*), two global navigation satellite system (GNSS) receivers (*u-blox EVK-M8T*) and an inertial measurement unit (IMU, *Xsens MTi-610*). The GNSS receivers, the IMU and the lidar sensor are mounted on a roof rack. The data of these four sensors is recorded using the robot operating system (ROS, [136]), which runs on a notebook. In the following evaluation only the data gathered from these four sensors is used. An image of the vehicle is shown in figure 2.1.

Before the study the participants were informed about the study and signed a consent form. After that followed a short test drive to get the participants acquainted with the vehicle. This was followed by the measurement drive itself; finally, the participants were asked to fill out a questionnaire. During the measurement drive the participants were guided along the course by an instructor who was seated in the rear of the vehicle. All participants followed the same course; in case of a missed turn or obstacles like temporary construction zones or blockages due to e.g. loading the instructor led them back to the planned course as quickly as possible. The course was approximately 22 km long and the measurement drive had a mean duration of 73.0 min ($\sigma = 6.4$ min).

Along the course there were 14 T-shaped intersections, i.e. intersections with three streets leading into it, and 4 X-shaped intersections (four streets per intersection). Of these intersections five T-intersections were selected explicitly, the remaining intersections were included as they lay along the way between the selected intersections. At one of these five intersections a deadlock situation was created by two instructed drivers. This intersection was passed three times from all three directions, resulting in nine passes. After each pass the participants were asked to rate the perceived complexity of the situation and to state their intention of driving first or last during the approach. After the deadlock situations the participants were guided to the four remaining selected intersections, which they traversed ten times in total from different directions. After each pass they were again asked to rate the perceived complexity. They drove through the remaining intersections without being asked any further questions. All 18 intersections considered here lay within zones

with a reduced speed limit of 30 km h^{-1}. The participants were on average 27.9 years (σ = 8.18 years) old and of the 34 participants 25 identified as male, 8 as female and one participant did not answer. The study was approved by the ethics commission of the Karlsruhe Institute of Technology.

For this work the data recorded at the intersection with the deadlock is not used as it does not provide any conclusive results. Similarly, the complexity ratings of the remaining four intersections are not used for the same reason. The runs through these four intersections are used, as the participants interacted with regular traffic and the questions were asked after the drivers passed the intersection and thus the questions did not have an influence on the driving behavior. Also, the runs through the remaining nine T-intersections and the 4 X-intersections are used for evaluation. In total there are 1818 runs through 13 T-intersections and 565 runs through 4 X-intersections. To distinguish this study from the one presented in chapter 3 it is in the following referred to as the *field study*.

To extract the individual runs from the raw sensor measurements of the field study, the data is further processed to first extract the driven trajectories. For that the GNSS, IMU and lidar data is fed into Google Cartographer [64], a simultaneous localization and mapping (SLAM) algorithm. This algorithm provides poses for each lidar frame. These poses position the individual lidar frames relative to the start point and, because of the GNSS receivers, also globally. The poses further describe the driven trajectory. Using the poses, the lidar frames can also be combined into larger but still consistent point clouds, which are needed for some of the features. In the next step the individual runs are extracted from the trajectories of the entire drives. For each run through one of the selected intersections all lidar frames are extracted for which the pose, and thus the vehicle, is within approximately 35 m from the intersection center. With that method a sequence of geo-referenced point clouds and the driven trajectory of the measurement vehicle are available for each run.

For further evaluation the maps of the intersections are needed as well. The maps are created using mapping data provided by the City of Karlsruhe. Each of these maps contain the intersection area itself and a

sufficiently long segment of each of the roads leading to the intersection. The maps are stored in a format that is a simplification of the lanelet concept [22, 127]. Each lanelet describes a lane segment. Lanelets begin where several lanes merge into one and end where a lane diverges into multiple lanes. An illustration of a lanelet map of an intersection is given in figure 4.2 of chapter 4. Each lanelet contains its left and right border and its predecessor and successor lanelets, thus creating a graphical representation of the driving possibilities. The lanelet sequence the measurement vehicle drove along is extracted next. For that purpose the driven trajectory is compared to all possible driving paths through the intersection[1] and the most similar path, and that way the lanelet sequence, is selected.

Some of the features introduced below require the detection of moving objects from the lidar point clouds, i.e. other vehicles, cyclists and pedestrians. As the resolution of the lidar sensor used here is low, neural network based approaches like *PointNet* [132] or *PointPillars* [86] are not used. Instead, an approach based on clustering is employed. First, the ground plane is removed using the MATLAB function `pcfitplane`, the remaining point clouds are further processed to only include reflections caused by objects on the street, i.e. other traffic participants. This is achieved by only keeping those points that are above the plane defined by the lanelets of the intersection. Also, points higher than 3 m are removed as they cannot belong to any vehicle and are probably caused by overhanging trees. Then, clustering is performed where points that belong to different clusters are at least 1.8 m apart. The bounding boxes of these clusters are fitted by using an L-shape fitting method [169]. Bounding boxes which are too large are removed but there is no minimum size for a cluster as many objects, due to the low resolution, only contain very few points.

The `trackerGNN` function in MATLAB is used, which implements a multi-object tracker with a global nearest neighbor assignment algorithm. In combination with an interacting motion model (IMM) tracking filter (`initekfimm` in MATLAB) tracks are generated from the center

[1] There are 6 possible paths through a T-intersection and 12 possible paths through an X-intersection. All of them are defined by a sequence of lanelets.

points of the clusters. This tracking filter uses three models of the tracked objects: one with constant velocity, one with constant acceleration and one with a constant turn rate. This is necessary because the traffic participants show changing behavior, i.e. they turn at the intersection, they accelerate, etc. Some of the tracked road users are detectable for long periods of time or may be occluded for some time. The tracker is parameterized for these cases. Occluded tracks are interpolated linearly, while stationary tracks are removed entirely. The velocity and the dimensions of the bounding boxes are filtered with a median filter; the class of a tracked object (pedestrian, cyclist, car, truck) is determined by the size and velocity of the bounding box. However, only two classes are used in this thesis: cyclists, cars and trucks are combined into one class as they are all traffic participants that are typically expected to be on the street while pedestrians are mostly only on the street if they are crossing it.

2.3 Intersection Complexity Features

As stated above, the underlying assumption in this chapter is that the intersection has an influence on the driving behavior of human drivers. Therefore, an intersection is described by a set of intersection features. For that, several properties of an intersection can be considered for their influence on the driving behavior. The first aspect are the properties of the trajectory driven by the participant. The second kind of features deals with the static environment of the intersection, i.e. its visibility, the area available or the perceived narrowness of the street and the intersection as a whole. The final aspect of the intersection that is assumed to have an influence on the driving behavior is the traffic at the intersection.

The property of the intersection that causes the hypothesized influence on the driving behavior is the complexity of an intersection. This property cannot be measured directly but is perceived by human drivers who are faced with a scenario at an intersection. In this chapter the complexity is thus viewed as the set (or relevant subset) of the intersection features that are presented in the following as a proxy for the perceived complexity. The terms *complexity*, *intersection complexity* and *intersection features* are used interchangeably in this chapter.

2.3.1 Driving Path Features

It is assumed that the driving path through the intersection is important for the driving behavior because for example turning through oncoming traffic is presumably more challenging and thus associated with a higher complexity than driving straight through an intersection. The driving path in the context of this work is the combination of the street the driver enters the intersection from and the direction he/she takes there, both of which are used as features. Both features can easily be obtained from the lanelet sequence the measurement vehicle drove along during its run through the intersection.

The **entry position** p_e is the direction from which a vehicle enters the intersection. In the case of T-intersections the intersection is turned so that it resembles the letter *T*. Then the three labels *left*, *bottom* and *right* are assigned to the three streets that make up the intersection. In case of the X-intersections the additional *top* label could be assigned to the northernmost street and the remaining labels could be distributed accordingly. However, since X-intersections are symmetrical in that from each entry position every turning direction is possible, this feature is only used for the T-intersections.

The **turning direction** p_t describes the direction of travel through the intersection of a vehicle. Three directions are possible: *left*, *right* or *straight*. Note that at T-intersections not all turning directions are available depending on the entry position, e.g. driving straight is not possible when entering a T-intersection from the bottom street.

It is important to distinguish between these features that merely describe the path a vehicle takes through an intersection and the actual driving behavior (e.g. the velocity) along that path, which are dependent on the driven trajectory. The latter does not describe the intersection but the driving behavior itself. Some features of that type are used to label the runs through the intersections, see section 2.4.

2.3.2 Stationary Features

Besides the driving path also the surroundings of an intersection are likely to have an influence on the driving behavior. This is plausible as for example a wider street with good visibility into the other streets is

probably less demanding and thus less complex than the opposite. The calculation of some of these features is visualized in figure 2.2.

The first feature of that kind is the **visibility distance**. It describes the distance of the measurement vehicle's position from the intersection center at which reference points in the remaining streets become visible for the first time. The positions of the reference points need to be defined in a consistent manner for all intersections. For that purpose the intersection center is projected onto the street center line of the streets entering the intersection; the center lines of these streets are extended by a straight segment to ensure the projection is possible. The reference points are placed on the center of the streets at the constant distance of

$$d_{\mathrm{ref}} = v_{\mathrm{max}} t_{\mathrm{r}} + \frac{v_{\mathrm{max}}^2}{2\,|a_{\mathrm{b}}|} \tag{2.8}$$

from the projected intersection center. This distance is measured along the extended center lines of the streets. With the speed limit of $v_{\mathrm{max}} = 30\,\mathrm{km\,h^{-1}}$, the reaction time $t_{\mathrm{r}} = 1\,\mathrm{s}$ and the assumed braking deceleration of $a_{\mathrm{b}} = -6\,\mathrm{m\,s^{-2}}$ the reference distance results in $d_{\mathrm{ref}} = 14.12\,\mathrm{m}$. This is the distance a human driver traveling at the speed limit needs to come to a complete stop when braking hard.

Unless otherwise specified, in the following the distance d of a vehicle is measured along the center of the lane the vehicle in question drives along. To this end the two points between which the distance is to be measured are projected onto the center line. If the vehicle is before the intersection center, d is positive, otherwise it is negative. The visibility distance $d_{\mathrm{v},i}^{\{\cdot\}}$ for each reference point i is the furthest distance d of the measurement vehicle at which the reference point in question is visible for the first time, i.e. it can be seen from the measurement vehicle's position.

Two methods of determining the overall visibility distance $d_{\mathrm{v}}^{\{\cdot\}}$ of an intersection are used as features in this work. The first version considers the reference points in all streets but the one straight ahead. This reference point is always visible at all considered intersections. For example a vehicle entering a T-intersection from the left would only have a reference point placed on the bottom street but not on the right street. Alternatively, only those streets can be considered from where a vehicle which the measurement vehicle would have to yield to would arrive from. If the

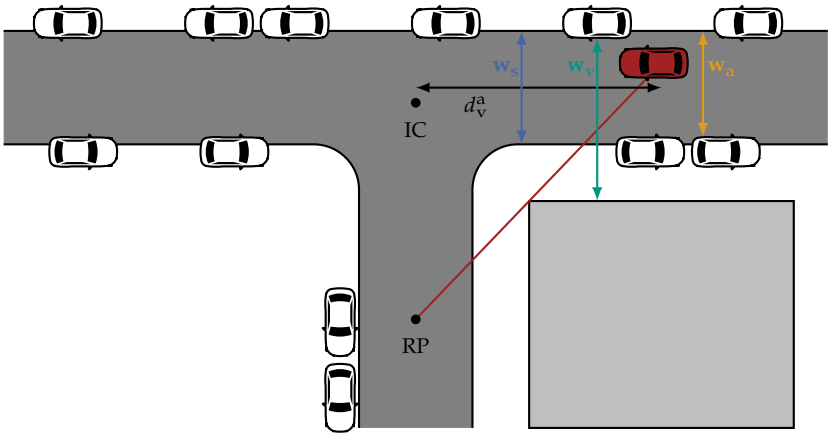


Figure 2.2 Visualization of the calculation of some of the features. The visible width $\mathbf{w}_v$, the street width $\mathbf{w}_s$, the available width $\mathbf{w}_a$ and the visibility distance based on the hand-drawn polygons $d_{v,p}$ is showcased here. For the latter case the intersection center (IC) and the reference point (RP) are required. The visualization is not to scale.

measurement vehicle e.g. arrives from the left road, a vehicle entering on the bottom street would have priority at a T-intersection. At an X-intersection also the vehicles from the right road would have priority; the visibility distance of that reference point is thus included as well. In both variants the visibility distance is the distance at which the last considered reference point is visible for the first time:

$$d_v^a = \min_i \left(d_{v,i}^a \right), \tag{2.9}$$

$$d_v^y = \min_j \left(d_{v,j}^y \right). \tag{2.10}$$

Where i includes all intersections that are not straight ahead, j contains the streets from which vehicles with priority could potentially arrive; d_v^a is the visibility distance for the method that considers all streets not straight ahead and d_v^y is the visibility distance for those streets one has potentially to yield to.

The visibility distances of the individual reference points $d_{v,i}^{\{\cdot\}}$ are measured by two different methods. The first is based on hand-drawn polygons of obstacles at the intersection and the second is based on the point

clouds of the lidar sensor. For the polygon-based approach a set of building polygons $\mathcal{P}_\mathrm{b}$ and a set containing polygons of the tree trunks at the intersection $\mathcal{P}_\mathrm{t}$ are used. Each of the points that make up the driven trajectory is checked if reference point i is visible from that position. The furthest point for which this is the case determines the distance d that is regarded as the visibility distance of reference point i. The point is visible if the sight line $s_i\,(d)$ does not intersect with any of the polygons in the two sets:

$$d_{\mathrm{v,p},i} = \arg\max_d \left((\mathcal{P}_\mathrm{b} \cup \mathcal{P}_\mathrm{t}) \cap s_i\,(d) = \emptyset \right). \tag{2.11}$$

The second variant of the visibility distance uses the point clouds of the lidar sensor. For that purpose the point clouds associated with the current distance d and the two before and after are merged into a combined point cloud $P_\mathrm{m}\,(d) = [\mathbf{x}, \mathbf{y}, \mathbf{z}]$ to get a denser representation of the environment. From the current vehicle position at distance d again a sight line s_i is drawn to reference point i. As this evaluation is done in three-dimensional space, a height for this line is also required. Both the reference point and the origin of the sight line within the vehicle are set to a height of 1 m. A cylinder[2] $\mathbf{C}_{\mathrm{s},i}\,(d)$ with a radius of 0.6 m is drawn around $s_i\,(d)$. Reference point i is considered visible if there is no point of P_m within $\mathbf{C}_{\mathrm{s},i}$:

$$d_{\mathrm{v,c},i} = \arg\max_d \left(\mathbf{C}_{\mathrm{s},i}\,(d) \cap P_\mathrm{m}\,(d) = \emptyset \right). \tag{2.12}$$

For both types of calculating the visibility distance, the approach based on the polygons and the one based on the point clouds, both variants of which reference points are considered are used as features for the behavior prediction. Thus there are in total four features available that describe the visibility conditions at an intersection.

To account for the narrowness of the streets (both the actual narrowness and the perceived narrowness are considered) several widths are defined and used as features. Less space to the sides during the approach might increase the perceived complexity of an intersection scenario. All widths are defined along the normal to the driven trajectory, which is parallel

[2] The function `point_to_line_distance` is used here (https://github.com/thrynae/point_to_line_distance, accessed 1 June 2024).

to the ground plane; $\mathbf{n}_\mathrm{s}(d)$ describes this normal vector at distance d from the intersection center. Using this normal, the **street width** $\mathbf{w}_\mathrm{s}$ is defined. It describes the width of the entire street, i.e. one's own lane and the opposite one, at the position of the measurement vehicle:

$$\mathbf{w}_\mathrm{s}(d) = \left|\mathbf{p}_{\mathrm{s,l}}(d) - \mathbf{p}_{\mathrm{s,r}}(d)\right|_2 . \tag{2.13}$$

In the above equation $\mathbf{p}_{\mathrm{s,l}}(d)$ describes the point where the normal at distance d intersects with the street curb to the left, and $\mathbf{p}_{\mathrm{s,r}}(d)$ is the equivalent with the right curb.

The width of the street is only one aspect, the perception of an intersection might also be influenced by the distance a driver can see to the sides. This hypothesis is covered with the second width feature, the **visible width** $\mathbf{w}_\mathrm{v}$. To implement it, the occurrence of the first lidar reflection to the left and right, i.e. along the normal to the driven trajectory, is calculated at sensor height $\mathbf{n}_\mathrm{v}$ for each trajectory point. As a lidar sensor only has a sparse coverage of its surroundings, points within a larger angular range to the side are considered: the first point within $\pm 10°$ in horizontal direction and within $\pm 5°$ in vertical direction determines the visible width. The position of the first object on the left side at distance d is given by $\mathbf{p}_{\mathrm{v,l}}(d)$; the first object to the right is located at $\mathbf{p}_{\mathrm{v,r}}(d)$. Consequently, the visible width at distance d is thus

$$\mathbf{w}_\mathrm{v}(d) = \left|\mathbf{p}_{\mathrm{v,l}}(d) - \mathbf{p}_{\mathrm{v,r}}(d)\right|_2 . \tag{2.14}$$

The final feature describing a width in this thesis is the **available width**; it describes the width on the street that is available for driving. It can thus be seen as a combination of the other two features as it reduces the street width in case there is an obstacle on the street:

$$\mathbf{w}_\mathrm{a}(d) = \min\left\{\mathbf{w}_\mathrm{s}(d), \mathbf{w}_{\mathrm{v,mod}}(d)\right\} . \tag{2.15}$$

The visual width $\mathbf{w}_{\mathrm{v,mod}}$ is modified here; a reflection is considered as the closest point if it is within the vertical range of $\pm 15°$. This is done to more reliably include vehicles that are parked on the street.

So far these three width features are calculated for each trajectory point, identified by their distance d, but in this work the features are scalar. For that reason the means of the widths within two segments are

used as features in the following. The first segment covers the part of the trajectory just before the intersection, the second one is calculated with the widths just after the intersection. These segments are specified depending on the distance d. All widths of trajectory points within that segment in question are averaged. For the widths before the intersection, the trajectory points within the distance range of $d_\mathrm{s} = 25\,\mathrm{m}$ and $d_\mathrm{e} = 7\,\mathrm{m}$ are included in the final features:

$$w_{\mathrm{s,b}} = \underset{d_\mathrm{s} > d > d_\mathrm{e}}{\mathrm{mean}} \left(\mathbf{w}_\mathrm{s}(d)\right), \tag{2.16}$$

$$w_{\mathrm{v,b}} = \underset{d_\mathrm{s} > d > d_\mathrm{e}}{\mathrm{mean}} \left(\mathbf{w}_\mathrm{v}(d)\right), \tag{2.17}$$

$$w_{\mathrm{a,b}} = \underset{d_\mathrm{s} > d > d_\mathrm{e}}{\mathrm{mean}} \left(\mathbf{w}_\mathrm{a}(d)\right). \tag{2.18}$$

The width features after the intersection include the widths associated with trajectory points within the range of $d_\mathrm{s} = -7\,\mathrm{m}$ and $d_\mathrm{e} = -15\,\mathrm{m}$:

$$w_{\mathrm{s,e}} = \underset{d_\mathrm{s} > d > d_\mathrm{e}}{\mathrm{mean}} \left(\mathbf{w}_\mathrm{s}(d)\right), \tag{2.19}$$

$$w_{\mathrm{v,e}} = \underset{d_\mathrm{s} > d > d_\mathrm{e}}{\mathrm{mean}} \left(\mathbf{w}_\mathrm{v}(d)\right), \tag{2.20}$$

$$w_{\mathrm{a,e}} = \underset{d_\mathrm{s} > d > d_\mathrm{e}}{\mathrm{mean}} \left(\mathbf{w}_\mathrm{a}(d)\right). \tag{2.21}$$

Another feature that describes the space at an intersection is the **free volume** V_f at an intersection. It is included as a more compact intersection (where e.g. objects are closer to the street and trees overhang the intersection area) might increase complexity and thus increase the perceived difficulty of driving there. For the calculation of this feature the lidar frames of a run through an intersection are combined into the global point cloud P_g. The space that is considered for this feature V_int is a cylinder with a radius of 25 m and a height of 6 m; its center coincides with the intersection center. The cylinder covers a height of 1 m below the ground to 5 m above the ground. The volume below ground is added to account for a possibly tilted or uneven driving surface. The center is excluded from evaluation by adding another cylinder of a radius of 2 m. This is done to avoid unreasonable results caused by reflections of the ego vehicle that are not properly removed or other vehicles that

also passed through the intersection. Removing points in the middle of the intersection is not problematic as this area has to be free anyway as driving would otherwise not be possible.

To determine the volume of this blocked evaluation space, the entire volume is filled with cubes with a side length of 0.5 m. For evaluation an entire cuboid with 50 m by 50 m by 6 m filled with these cubes is used, resulting in 120 000 cubes. The cuboid is placed so that the outer cylinder is fully contained in it. The k-th cube of the cuboid surrounding V_{int} is described by V_{int}^{k}; these are referred to as voxels in the following. The evaluation is based on a mask variable m_{v}, initially, $m_{\mathrm{v}} = 0, \forall k$. In a first step all voxels are checked whether they lie within the evaluation volume, i.e. within the outer cylinder and outside the inner one. For each voxel k for that this is not the case the mask is set to $m_{\mathrm{v}}^{k} = 1$. Each voxel inside the evaluation volume is next checked whether it is occupied, this is the case if there are at least 10 reflections within it; $m_{\mathrm{v}}^{k} = 2$ for all occupied voxels k.

It is not sufficient to only deal with these voxels that are directly occupied. All voxels behind occupied ones have to be considered as blocked ($m_{\mathrm{v}}^{k} = 3$) because they are not visible from the street and thus do not contribute to the free volume at the intersection. The determination whether a non-occupied voxel within the evaluation volume is free or blocked is dependent on whether there is an occupied voxel between the center of V_{int} and the currently considered voxel[3]. After this evaluation is performed for all voxels within V_{int} the mask m_{v} now contains the relevant information:

$$m_{\mathrm{v}}^{k} = \begin{cases} 0, & \text{if voxel } k \text{ is free,} \\ 1, & \text{if voxel } k \text{ is outside the evaluation space,} \\ 2, & \text{if voxel } k \text{ is occupied,} \\ 3, & \text{if voxel } k \text{ is blocked by an occupied voxel.} \end{cases} \tag{2.22}$$

Based on these results the free volume at the intersection feature is calculated as follows:

$$V_{\mathrm{f}} = \frac{N_{\mathrm{f}}}{N_{\mathrm{f}} + N_{\mathrm{o}} + N_{\mathrm{b}}}. \tag{2.23}$$

[3] The function `wooRaytrace` is used here (https://de.mathworks.com/matlabcentral/fileexchange/56527-fast-raytracing-through-a-3d-grid, accessed: 1 June 2024).

The number of free voxels is given in N_f, N_o contains the number of occupied voxels and the number of blocked voxels is given in N_b.

Additionally, the **minimum street angle** α_s is used to describe an intersection. The minimum street angle is defined as the minimum of all angles between pairs of the streets leading into the intersection. It is included in the feature set because it is assumed that a smaller angle between two streets makes it more demanding to see from one of the streets into the other and to interact with drivers in the other street, thus increasing the complexity of an intersection. To calculate the angles, direction vectors are defined for all the streets that make up an intersection. This is based on the center line of the street: $\mathbf{s}_\mathrm{i} = \mathbf{s}_{\mathrm{i},2} - \mathbf{s}_{\mathrm{i},1}$. Point $\mathbf{s}_{\mathrm{i},1}$ is the last point of the center line of street i, i.e. the point where the street diverges and thus the intersection begins. $\mathbf{s}_{\mathrm{i},2}$ is the intersection of the street center line with a circle with a 5 m radius around $\mathbf{s}_{\mathrm{i},1}$. The angle between two streets i and j is then:

$$\alpha_{\mathrm{s},ij} = \cos^{-1}\left(\frac{\mathbf{s}_\mathrm{i} \cdot \mathbf{s}_\mathrm{j}}{|\mathbf{s}_\mathrm{i}||\mathbf{s}_\mathrm{j}|}\right), \quad 1 \leq i \leq N_\mathrm{s}, 1 \leq j \leq N_\mathrm{s}. \tag{2.24}$$

N_s is the number of streets at the intersection, thus $N_\mathrm{s} = 3$ for T-intersections and $N_\mathrm{s} = 4$ for X-intersections. The minimum street angle α_s for an intersection is the minimum of all pairwise angles:

$$\alpha_\mathrm{s} = \min_{i,j;\ i \neq j} \alpha_{\mathrm{s},ij}. \tag{2.25}$$

The inclusion of this feature is motivated by previous findings: higher crash frequencies are observed at non-orthogonal intersections [41]. Also, accidents at such intersections lead to more severe accidents between motor vehicles and cyclists [14].

The final stationary feature is the **number of trees** n_t. For that purpose all those trees are counted whose trunk is positioned around the intersection area or besides the street the measurement vehicle enters the intersection from. Trees behind the intersection and those that are too far in front of it are not considered. For this feature the hand-drawn tree polygon set $\mathcal{P}_\mathrm{t}$ is used again.

2.3.3 Traffic Features

The final class of features describes the traffic during the approach to an intersection. These are included as well as it is assumed that more traffic at an intersection increases its perceived complexity. All features of this type depend on the pedestrians and vehicles that are detected within the lidar point clouds. This is also one of the two limitations that is inherent with this type of features. Clusters that are not detected or that are falsely classified directly affect the feature and thus their predictive power. More importantly, most runs within the data set were without any cooperation vehicles. The influence of the traffic features on the prediction of the driving behavior is therefore probably limited.

The first feature of this type is the **number of visible vehicles** n_{v} during the approach to the intersection. For this feature all vehicles are counted that are detected in the lidar data while the measurement vehicle approaches the intersection, i.e. $d \geq 0$. The **number of pedestrians** is defined accordingly, in that all pedestrians that are visible anywhere during the approach are counted. It is important to note, however, that, since only objects on the street polygon are tracked and are thus considered for classification, exclusively pedestrians walking on the street are included in this feature.

The **number of vehicles that interact** with the measurement vehicle n_{vi} is also used as a feature. Vehicles are interacting with the measurement vehicle if they are observed to pass the intersection center and if their distance along their trajectory is within 10 m to the intersection center at the same time as the measurement vehicle, i.e. $|d| < 10\,\mathrm{m}$ is true for both vehicles at the same time. The final two features are the **number of interacting vehicles with the right of way** n_{rw} and the **number of interacting vehicles that have to give way** n_{gw}. For that purpose each vehicle that is considered to be an interacting vehicle is classified if it has the right of way over the measurement vehicle or if it has to yield to the measurement vehicle. There are also interacting vehicles that do not fall in either category, e.g. a vehicle that drives on the same path through the intersection as the measurement vehicle. All complexity features, including those of the previous two subsections, are listed in table 2.2.

2.4 Behavior Features

In order to quantify the driving behavior of the participants at the intersections three features are defined. These features need to describe important aspects of the approach to an intersection and of the decision-making process there. Additionally, the features have to be defined in accordance with the prevalent conditions at the intersections. These include a speed limit of $30\,\mathrm{km\,h^{-1}}$and a varying number of interaction partners, including the case of no additional vehicle. As discussed above, the latter aspect does not allow the usage of features that are commonly used for the TCT [31, 39, 77] like the time to encounter or others that rely on the interaction between a pair of vehicles. Instead, features that only consider the driving behavior of the participants are used here.

The first behavior feature is the **commit distance** d_c. It is the distance at which, given the current velocity and taking the reaction time into consideration, stopping before the intersection center is no longer possible:

$$d_\mathrm{c} = \max_d \left(d < t_\mathrm{r} v(d) + \frac{v(d)^2}{2|a_\mathrm{b}|} \right). \tag{2.26}$$

It can be seen as the distance at which the decision to drive has been made. The decision is presumably made farther from the intersection because of the relatively high braking acceleration of $a_\mathrm{b} = -6\,\mathrm{m\,s^{-2}}$ and the fact that usually one would stop before the intersection center. At a distance closer than the commit distance the decision can no longer be reverted, a driver is thus committed to pass the intersection center and by that enter the intersection in any case. This feature is included as a descriptor of the driving behavior as a larger commit distance can be seen as more offensive driving behavior. This is plausible because driving at a relatively high velocity close to the intersection results in a larger commit distance than e.g. cautiously and slowly approaching the intersection.

A further feature for describing the human driving behavior is the **minimum velocity** v_m during the approach to the intersection. It is the minimum velocity the participant drives at while being within the approach interval of $d_\mathrm{s} = 25\,\mathrm{m}$ to $d_\mathrm{e} = 0\,\mathrm{m}$ from the intersection center:

$$v_\mathrm{m} = \min\left(v\left(d\right)\right), \quad d_\mathrm{s} > d > d_\mathrm{e}. \tag{2.27}$$

It is included under the assumption that the behavior is more defensive the lower the minimum velocity. However, slow driving or even stopping can also be caused by other circumstances, e.g. if a driver has to give way to another vehicle.

The final behavior feature considered in this work is the **velocity drop** v_d. It is the quotient of the velocity during the initial approach v_a and the minimum velocity v_m:

$$v_\mathrm{d} = \frac{v_\mathrm{m}}{v_\mathrm{a}}. \tag{2.28}$$

Where v_a is the average velocity within the interval from $d_\mathrm{s} = 25\,\mathrm{m}$ to $d_\mathrm{e} = 20\,\mathrm{m}$. In contrast to the minimum velocity the velocity drop also considers the amount by which a driver brakes at an intersection. It is assumed that the more defensive the driving behavior, the more pronounced the drop in velocity.

It is important to note that the driving behavior could also be described by further features; many more aspects of the approach to the intersection and the driving within it could be used. These features to describe the driving behavior could e.g. include the velocity itself or features based on acceleration. The behavior features used here are listed in the bottom part of table 2.2.

2.5 Behavior Prediction Results

Based on the features of intersection complexity the driving behavior is predicted. For that purpose the runs are labeled with the driving behavior features of the previous section and regression models are trained. First the entire set of complexity features is used for prediction, followed by subsets consisting of the most important features. Finally, an autoencoder is used to generate a lower dimensional representation of the feature set, which is then used for prediction. Separate models are trained for the T-intersections and the X-intersections.

The runs are split into a training set which contains 70 % of all runs and a test set which contains the remaining 30 %. The runs are randomly assigned to these sets. For each variant 10 models with random set assignments are trained; the reported regression performance is averaged

Table 2.1 Performance of different regression algorithms for behavior prediction using the full feature set. All three behavior features are used as labels: The commit distance d_c in m, the minimum velocity v_m in m s^{-1} and the velocity drop v_d. For each combination the prediction is run 10 times with different training set assignments and the results are averaged over these predictions. The prediction error is reported using the RMSE in the first row of each algorithm; the standard deviation is given in the second rows.

		T-intersections			X-intersections		
algorithm		d_c	v_m	v_d	d_c	v_m	v_d
LR	RMSE	1.624	1.089	0.157	1.850	1.179	0.169
	σ	0.069	0.028	0.005	0.112	0.091	0.010
RF	RMSE	1.437	0.990	0.147	1.623	1.103	0.161
	σ	0.062	0.033	0.006	0.114	0.068	0.012
TB	RMSE	1.458	0.981	0.148	1.653	1.091	0.156
	σ	0.059	0.021	0.006	0.082	0.046	0.009
Reference	RMSE	3.131	2.002	0.276	3.230	2.004	0.257
	σ	0.091	0.039	0.005	0.166	0.082	0.011

over the results of all 10 models. The prediction models are evaluated by the root mean squared error

$$\mathrm{RMSE} = \sqrt{\frac{1}{N_{te}} \sum_{i=1}^{N_{te}} (\hat{y}_i - y_i)^2}. \tag{2.29}$$

The number of runs in the test set is given by N_{te}, $\hat{y}_i$ is the estimated label of the i-th run of the test set and y_i is the true label of that run. As a benchmark for the model performances a reference label $\bar{y} = \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} y_{tr,i}$ is introduced, where N_{tr} is the number of runs in the training set and $y_{tr,i}$ the label of the i-th run in the training set. When using this benchmark, the average label of the training set is assigned to the runs of the test set: $\hat{y}_i = \bar{y}$.

2.5.1 Full Feature Sets

For the first evaluation the entire complexity feature set of table 2.2 is used for behavior prediction. This includes the driving path features,

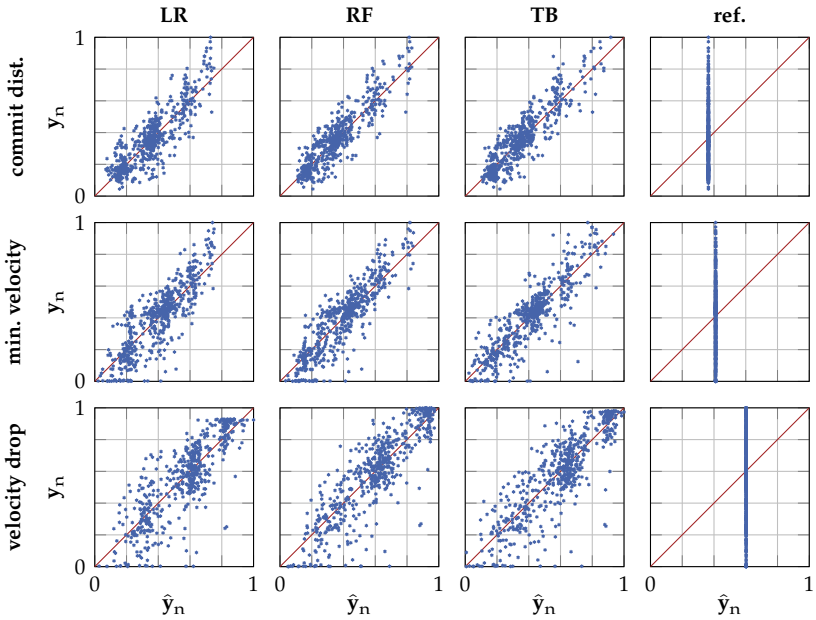


Figure 2.3 Scatter plots of the behavior prediction at the T-intersections with the full feature set. All three behavior features are used as labels and LR, RF and TB are used for regression. For comparison, the regression with the naive reference is also given. The true labels $\mathbf{y}_n$ and estimated labels $\hat{\mathbf{y}}_n$ are normalized to the maximum value of each individual regression.

the stationary features and the traffic features; 20 intersection features in total. In case of the models for the X-intersection there are only 19 features as the entry position p_e is not used due to the symmetry of the X-intersections. The three behavior features of section 2.4, the commit distance d_c, the minimum velocity v_m and the velocity drop v_d are used as the labels for behavior prediction. The performance metrics of these prediction models are shown in table 2.1. Additionally, the scatter plots showing the true label values $\mathbf{y}_n$ over the estimated label values $\hat{\mathbf{y}}_n$ are presented in figure 2.3 for the T-intersection and the corresponding plots for the X-intersection are given in figure 2.4. The data for these plots are of the model that has the best performance on the test set for its prediction

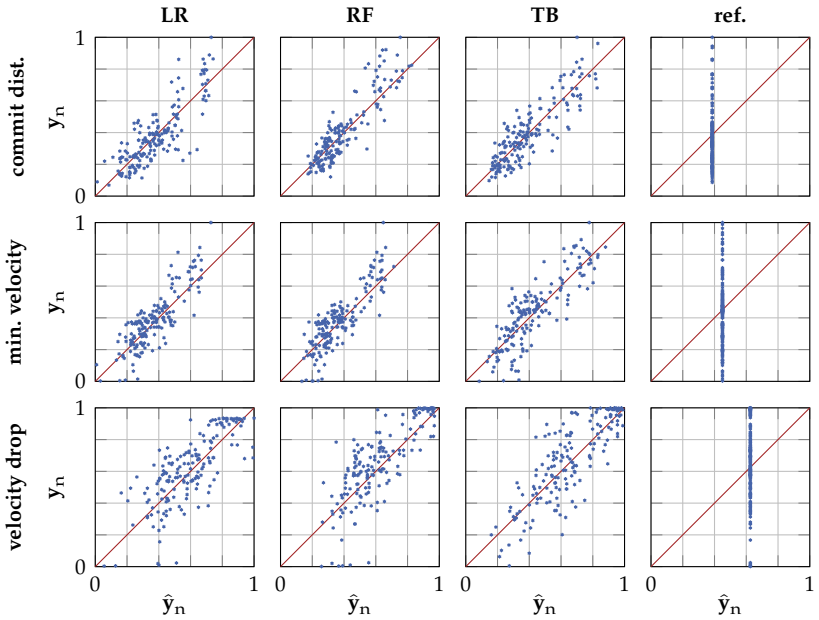


Figure 2.4 Scatter plots of the behavior prediction at the X-intersections with the full feature set. All three behavior features are used as labels and LR, RF and TB are used for regression. For comparison, the regression with the naive reference is also given. The true labels $\mathbf{y}_n$ and estimated labels $\hat{\mathbf{y}}_n$ are normalized to the maximum value of each individual regression.

variant. The scatter plots are normalized to the overall maximum value y_m of the true values $\mathbf{y}$ and the estimated values $\hat{\mathbf{y}}$:

$$y_{n,i} = \frac{y_i}{y_m}, \quad \hat{y}_{n,i} = \frac{\hat{y}_i}{y_m}, \quad y_m = \max\{\mathbf{y}, \hat{\mathbf{y}}\}, \quad i \in 1, \dots N_{tr}. \tag{2.30}$$

The normalization factor y_m is calculated individually for each scatter plot.

The quantitative results of table 2.1 show the expected behavior in that all classifiers outperform the reference values substantially. Of the three algorithms, LR shows the worst performance while RF and TB reach the best results. The predictions from the two latter methods are very close to each other with RF being slightly better in all cases but when

the velocity drop v_d or the minimum velocity v_m is used as the label at the X-intersections. The standard deviations are relatively low for all variants, but they are consistently larger in case of the X-intersections.

The qualitative results of the scatter plots show that the test sets are predicted relatively close to the ideal line; this is especially true for the commit distance. The performance of the velocity drop and, to a smaller extent, the minimum velocity are not as good. In these cases there are more outliers and especially those cases where $\mathbf{y}_\mathrm{n} \approx 0$ the estimated values $\hat{\mathbf{y}}_\mathrm{n}$ vary notably for these two behavior features. The better performance of TB and RF observed in table 2.1 is also visible in the scatter plots: The resulting point clouds are more compact and closer to the ideal line for RF and TB than in case of LR. The labels of the reference method $\hat{\mathbf{y}}$ are the same for each run of the test set. This manifests itself in the vertical alignments of the individual marks within these scatter plots.

Some conclusions can be drawn from these results: It is possible to predict the driving behavior at an intersection using only information about the intersection itself. From that it follows that the driving behavior at an intersection, to a large part, does in fact depend on the intersection itself. It also means that there is a typical behavior, which is independent from a driver's personality, intentions or mood. These aspects certainly play an important role as well and probably explain some of the prediction errors. The superior performance of RF and TB over LR may be explained by the fact that the latter can only represent linear dependencies while RF and TB are able to find non-linear dependencies as well.

2.5.2 Importance of Features

So far, the regression models include all intersection features, regardless of their influence on the prediction performance. In this section the possibility to reduce the number of features for regression is investigated. As stated before, an important motivation for this thesis is the desire to be able to explain the reasons for human driving behavior. As is evident from the previous section, the intersection features are helpful for that purpose. However, there are 20 features (19 for X-intersections) to describe the intersections themselves, the traffic there and the selected trajectory. It is certainly not possible to describe the behavior with so

Table 2.2 List of all features. The intersection features are listed first, followed by the behavior features. The list of intersection features is ordered by the importance of the features when both intersection types are viewed in combination. Additionally, the importance is given for both the T- and X-intersections separately. As the entry position p_e is only part of the feature set for the T-intersections, it is not part of the combined rank. There are several visibility features; they are either calculated by using obstacle polygons (OP) or the lidar point clouds (PC) and either all streets or the streets the test vehicle has to yield to (YS) are considered. Some feature pairs score the same number of points which results in a shared rank.

features		rank	rank T-int.	rank X-int.
p_e	entry position	-	3	-
V_f	free space	1	2	1
p_t	turning direction	2	1	8
$w_{s,b}$	street width before int.	3	5	5
$d^y_{v,p}$	vis. distance (OP) at the YS	4	6	2
$w_{a,b}$	available width before int.	5	8	4
$w_{s,e}$	street width after int.	6	4	9
$d_{v,p}$	vis. distance (OP)	7	10	3
$w_{a,e}$	available width after int.	8	9	7
$d^y_{v,c}$	vis. distance (PC) at the YS	9	7	12
n_t	# trees at int.	10	18	6
$d_{v,c}$	vis. distance (PC)	10	16	10
$w_{v,e}$	visible width after int.	12	15	11
$w_{v,b}$	visible width before int.	13	13	13
α_s	minimum angle	14	11	14
n_v	# visible vehicles	15	12	16
n_{vi}	# interaction vehicles	16	14	15
n_{rw}	# vehicles with right of way	17	17	17
n_{gw}	# vehicles that have to yield	18	19	19
n_p	# visible pedestrians	19	20	18
d_c	commit distance	-	-	-
v_m	minimal velocity	-	-	-
v_d	velocity drop	-	-	-

many features easily. It would thus be helpful if behavior prediction was possible with fewer features.

To that end the intersection features are first ordered by importance. As introduced in section 2.1.2.2, importance scores for each feature are available from training RF models. These scores are used to rank the complexity features by their importance for behavior prediction. For better comparability and to simplify the training process, a consistent ranking of all behavior features and both intersection types is tabulated. For that purpose the importance scores of all 60 RF models from section 2.5.1 (2 intersection types, 3 behavior features and 10 models per variant) are used. For each model the most important feature is assigned one point, the second most important two points and so on. The final order is then obtained by summing the points of all models. The feature with the lowest number is considered to be the overall most important feature. The list of complexity features in table 2.2 is ordered by this combined feature importance. Additionally, the orders that would result when only considering the T- or X-intersections, respectively, are shown as well. These are not used in the following and are only given for comparison. As the entry position p_e is only part of the feature set of the T-intersections, it is omitted from the calculation of the combined feature importance ranking.

The most important feature of the combined evaluation is the free space at the intersection V_f followed by the turning direction p_t. The latter is a noteworthy case because it is the most important feature when only considering the T-intersections, but it is only in the eighth place if the X-intersections are evaluated separately. The different importance of some features within the T-intersections data set and the X-intersections data set is not limited to the turning direction. The visibility distance using the object polygons $d_{\mathrm{v,p}}$ is ranked on position 10 for the T-intersections and on position 3 for the X-intersections.

For the following evaluations, models are trained the same way as those in the previous section but with varying numbers of features. First, a model is trained that has access to only the most important feature, the next model uses the two most important features. This pattern is continued until all features are used. For this analysis again both the T- and the X-intersection data sets are used. Also, all three behavior fea-

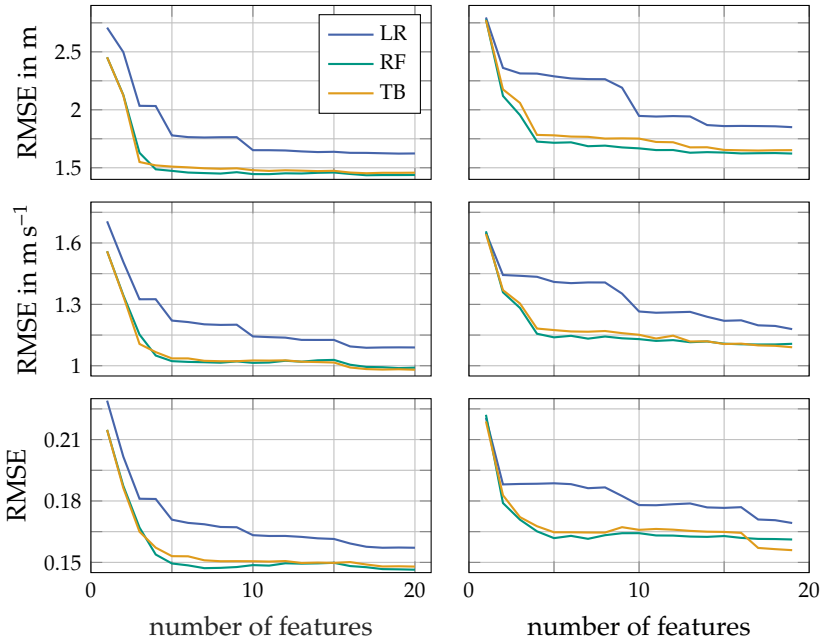


Figure 2.5 Averaged behavior prediction error (RMSE) over the number of intersection features used for prediction. The features are added by importance; the models with only one feature use the most important one. The first row of plots shows the prediction results using the commit distance d_c as the label, the second row those using the minimum velocity v_m and the last row shows the performance of the models for the velocity drop v_d. The left column shows the results using the T-intersections data set, the right column contains the results of the X-intersections. LR, RF and TB are used for regression.

tures are used as labels and models are trained with all three regression algorithms as before. In case of the T-intersections the unranked feature entry position p_e is placed at the top of the list. The results are plotted in figure 2.5. For each of the six combinations of intersection type and algorithm the RMSE is plotted over the number of features. For each variant again 10 models are trained, the data in the plots is the average of the models' performances.

The results show an improved performance, i.e. a lower RMSE, with an increasing number of features. This is true for all training combinations

and all regression algorithms. In all six combinations LR performs worst while RF and TB are again very similar in their performance with a small advantage of RF over TB. Adding an additional feature improves the performance to a greater extent when there are only few features compared to the case when already many features are used. The performance gain in the latter case is minimal, the performance over number of feature curve is almost constant for more than 5 features for the T-intersections and more than 6 features in case of the X-intersections. This is true for RF and TB, the performance when using LR improves slower over the number of features and does not converge to the superior performance of RF and TB even for a larger number of features.

A plausible explanation for the similar performance of the feature sets that include more than 5 or 6 features, respectively, lies in the fact that there are several features, especially those concerning the visibility and the widths, that are defined similarly and thus probably correlated to a certain degree. For that reason one might expect that removing some of these does not have a large effect on the prediction results. Further, most runs do not have any interaction vehicles present at the intersections. A vehicle is detected in only 22.95 % of all runs (T-intersection: 23.71 %, X-intersection: 20.53 %). Even more seldom, in only 2.10 % of all runs a pedestrian is detected on the street. This presumably leads to the low importance of the traffic features in table 2.2 and therefore predictions without most of these features do not decrease the performance by much.

The varying importance of the turning direction p_t at the different intersection types is probably caused by the distribution of p_t in the data sets. Of the 565 runs through an X-intersection 222 (39.29 %) entail a right turn, 67 (11.86 %) a left turn and in 276 of those runs (48.85 %) the vehicle is driven straight. In comparison, the distribution at the T-intersections is more balanced: Of the 1818 runs through the T-intersections, 542 (29.81 %) consist of a right turn, 774 (42.57 %) of a left turn and in 502 (27.61 %) cases the vehicle is driven straight through the intersection. This presumably enables this feature to be more important for the predictions in case of the T-intersections.

Besides the different distributions of feature values, the varying importance of features between the T- and X-intersections might also be caused by the distribution of intersections within the two data sets. There are

Table 2.3 Averaged RMSE of the behavior prediction using different feature sets and algorithms at the T-intersections. The full feature set (FFS) is shown again for comparison, the reduced feature set (RFS) contains the 8 most important features. The remaining feature sets are listed explicitly. The behavior features are used as labels: the commit distance d_c in m (reference method: 3.131 m), the minimum velocity v_m in m s^{-1} (reference method: 2.002 m s^{-1}) and the velocity drop v_d (reference method: 0.276).

		FFS	RFS	$\{p_t\}$	$\{V_f, p_e\}$	$\{V_f, p_t\}$	$\{p_e, p_t\}$	$\{V_f, p_e, p_t\}$
d_c	LR	1.624	1.763	2.193	2.498	2.037	2.194	2.034
	RF	1.437	1.450	2.135	2.128	1.915	1.815	1.635
	TB	1.458	1.492	2.135	2.125	1.928	1.750	1.550
v_m	LR	1.089	1.200	1.503	1.508	1.348	1.490	1.325
	RF	0.990	1.015	1.501	1.345	1.295	1.303	1.153
	TB	0.981	1.022	1.501	1.340	1.293	1.274	1.107
v_d	LR	0.157	0.167	0.208	0.202	0.188	0.204	0.181
	RF	0.147	0.147	0.209	0.187	0.182	0.185	0.167
	TB	0.148	0.151	0.209	0.186	0.182	0.182	0.165

some substantial differences between those: The X-intersections data set consists of only 7 intersection/entry position combinations and of the 565 runs through all intersections 404 were recorded at a single intersection (but from three different entry positions). In contrast, the T-intersection data set consists of 34 combinations of intersection/entry positions. Of these, one combination has only a single run and another has two runs; this is probably caused by a participant turning wrong or being rerouted due to a blocked street. The remaining combinations all have more runs which are distributed more evenly among them than in the X-intersection case.

2.5.3 Reduced Feature Sets

The previous section shows that predictions with a smaller feature set are also feasible. In this section some important or interesting variants with fewer intersection features are viewed in detail. First, a *reduced feature set* is introduced. It only contains the 7 most important complexity features and can be seen as a compromise between the variants of figure 2.5. This set

Table 2.4 Averaged RMSE of the behavior prediction using different feature sets and algorithms at the X-intersections. The full feature set (FFS) is shown again for comparison, the reduced feature set (RFS) contains the 7 most important features. The remaining feature sets are listed explicitly. The behavior features are used as labels: the commit distance d_c in m (reference method: 3.230 m), the minimum velocity v_m in m s^{-1} (reference method: 2.004 m s^{-1}) and the velocity drop v_d (reference method: 0 257).

		FFS	RFS	$\{V_f\}$	$\{p_t\}$	$\{V_f, p_t\}$
d_c	LR	1.850	2.265	2.796	2.650	2.361
	RF	1.623	1.686	2.780	2.634	2.129
	TB	1.653	1.767	2.774	2.634	2.176
v_m	LR	1.179	1.407	1.650	1.705	1.443
	RF	1.103	1.132	1.657	1.707	1.358
	TB	1.091	1.166	1.644	1.707	1.369
v_d	LR	0.169	0.186	0.221	0.213	0.188
	RF	0.161	0.162	0.222	0.213	0.179
	TB	0.156	0.165	0.219	0.213	0.183

thus includes the free volume at the intersection V_f, the turning direction p_t, the visibility distance using the obstacle polygons $d_{v,p}$, the variant of that distance using only the yield streets $d^y_{v,p}$, the mean street width before and after the intersection ($w_{s,b}$ and $w_{s,e}$, respectively) and the available width before the intersection $w_{a,b}$. In case of the T-intersections also the entry position p_e is included within the reduced feature set.

Additional models are trained with feature sets containing combinations of the two most important features, the free volume at the intersection V_f and the turning direction p_t. For the T-intersection models also combinations including the entry position p_e are trained and evaluated. Table 2.3 contains the performances of these models using the runs through the T-intersections. Table 2.4 shows the corresponding results for the X-intersections. These tables also contain the performance of the full feature sets as a benchmark. In case of the T-intersections the table shows the results of the best performing feature set with only one feature (the turning direction p_t), all three combinations with two features and the feature set which consists of all three most important features. The

table for the X-intersections is set up similarly, but as only the two best features are being considered, the performance of both feature sets with only a single feature and the results of the set with both features are reported in the table. The scatter plots for some of these feature sets are presented in figures A.1, A.2 and A.3 in the appendix.

From the results in table 2.3 one can see that the predictions using the reduced feature sets are almost as good as those with the full feature sets. Of the feature sets made up of the three best performing features, those with only one feature perform worst, followed by those with two features. Using all three features yields in the lowest errors, but the results are still worse than those of the reduced feature sets. When using only a single feature for prediction, the turning direction p_{t} is best in all cases. The free space at the intersection cannot be used as the single feature, as this results in predictions which are comparable to those of the reference method (not reported in the table). The entry position p_{e} is worse than p_{t} for all regression methods but for some of them the results are similar. The performances when using a combination of two features as the feature set are closer to each other. The combination $\{V_{\mathrm{f}}, p_{\mathrm{e}}\}$ has the largest errors of these variants, the other two combinations are, overall, relatively similar and each is best for some combinations of label and algorithm. Finally, the feature set which uses all three features is better than all these sets, but still has substantially larger errors than the reduced or even the full feature sets. RF and TB show similar performances and the predictions by either are better than those of LR.

The results of the X-intersection data set are similar, the feature sets using only a single feature are again those with the largest prediction errors. This is followed by the feature set using the two best features and the reduced and finally the full feature sets. In case of the feature sets containing only a single feature both variants show relatively large prediction errors. Also, LR was again the least reliable prediction algorithm; it is outperformed by both RF and TB whose performances are again on par with each other.

Overall, these results are in line with those of the previous sections. Using only a subset of all intersection features reduces the prediction performance. This reduction is small for the reduced feature set, especially if RF or TB is used for regression. In general, the fewer features the worse

the predictions, as can be seen from the feature sets that are made up of only the three most important features. This is presumably caused by the fact that using only one to three features takes less information into consideration during the training process of the models. Additionally, both p_t and p_e are categorical features with only three different categories each. In case of the $\{p_e, p_t\}$ feature set at the T-intersections only six combinations are possible (e.g. when entering from the bottom it is impossible to drive straight) and if these features are even used exclusively, only the three categories are available. In either case this has the consequence that there are only as many prediction values possible as there are combinations of feature values. This phenomenon is clearly visible in the scatter plots in the appendix (c.f. figures A.1, A.2 and A.3). This limited set of possible prediction values further contributes to the increased prediction error.

The performance of turning direction p_t and the free space at the intersection V_f show some interesting patterns: V_f is ranked as the overall most important feature and is on the second place when only considering the T-intersections and again on the first place when considering only the X-intersections. Despite this, using this feature individually yields very poor predictive potential. It appears that this feature only becomes relevant when it is combined with further ones. As discussed in the previous section, p_t is ranked differently for the T- and X-intersections (positions 1 and 8, respectively). Nonetheless, using this feature as the only one for prediction results in the best performance (compared to the other features used exclusively in this section) for both intersection variants.

The observations regarding the performance of the regression algorithms are confirmed: LR performs worst in all cases while TB and RF are substantially better in all other cases and have a performance similar to each other on all feature sets. These observed patterns are again common across all three labels.

2.5.4 Autoencoder for Feature Set Reduction

In the previous sections the desired reduction of the complexity feature set is achieved by simply omitting some of the features from the full feature set. This, however, might not be the most efficient approach to

Table 2.5 Averaged RMSE of the behavior prediction using feature sets that are reduced by autoencoders. The training of the autoencoders themselves is based on the full feature sets. The behavior features that are used as labels are the commit distance d_c in m, the minimum velocity v_m in m s^{-1} and the velocity drop v_d. The reference values are as before.

		T-intersection				X-intersection			
		FF	1D	2D	3D	FF	1D	2D	3D
d_c	LR	1.624	2.802	2.072	1.986	1.85	2.921	2.804	2.658
	RF	1.437	1.911	1.672	1.602	1.623	2.681	2.276	2.075
	TB	1.458	1.824	1.679	1.637	1.653	2.655	2.374	2.146
v_m	LR	1.089	1.821	1.409	1.365	1.179	1.885	1.809	1.729
	RF	0.990	1.355	1.215	1.174	1.103	1.735	1.503	1.359
	TB	0.981	1.298	1.216	1.191	1.091	1.712	1.556	1.404
v_d	LR	0.157	0.251	0.196	0.191	0.169	0.241	0.230	0.219
	RF	0.147	0.193	0.175	0.173	0.161	0.225	0.202	0.186
	TB	0.148	0.185	0.175	0.175	0.156	0.222	0.211	0.191

reduce the number of features. As stated above, many of the features are probably correlated with each other to a certain degree and thus combining these features into fewer, more diverse ones, seems promising. To that end an autoencoder (AE), a type of neural network that can be used for dimensionality reduction [65], is used. Making use of an AE has the advantage that it is able to find nonlinear dependencies, and it is reported to outperform principal component analysis (PCA) [65].

A neural network is typically made up of several layers of so-called neurons [57]. The number of neurons per layer is a design parameter. An AE consists of two parts, the encoder and the decoder [57]; the dimension at the output of the encoder is smaller than its input for an undercomplete AE [57], as is used here. At the output a compressed representation of the input data is available. During training a decoder is also employed: It is trained to reconstruct the input data of the encoder using the compressed representation at the output of the encoder as its input. If an AE is used for dimension reduction, the number of neurons at the bottleneck, i.e. the interface between the encoder and decoder, determines the dimension of the reduced representation. During training a cost function is made use

of that compares the original training data at the input of the encoder to its reconstructed version at the output of the decoder.

In this work an AE is used to generate a lower-dimensional representation of the intersection complexity feature space. For that purpose an encoder-decoder structure is trained where the dimension of the bottleneck is the same as the desired dimension of the feature set and is set to one of these values: $N_{\mathrm{AE}} = \{1, 2, 3\}$. Both the encoder and the decoder have three layers each. The input of the first layer of the encoder has as many neurons as there are features in the full feature set, i.e. 24 in case of the T-intersections and 21 in case of the X-intersections. There are more features for the training of the AE and the later inference than there are on the list of features in table 2.2 because the categorical features entry position p_{e} and turning direction p_{t} are one-hot-encoded [75]. Both have three possible values and are thus represented by three features each. The dimensions are reduced to 15, 6 and N_{AE} neurons, respectively, in the following layers. All layers are fully connected layers with batch normalization [57] and a rectified linear unit (ReLU) [75] as the activation function. The decoder is set up in reverse. In the cost function categorical and continuous features are viewed separately. For the categorical features cross entropy [75] is used, the reconstruction error of the continuous features is measured using the mean absolute error. The cost function is then the weighted sum of both elements.

The AE is trained with the training set (55 % of all runs in the data set) and validated with the validation set (15 % of the runs). After an AE model is trained, it is used to calculate the low-dimensional representation of the training, validation and test sets. The training and validation sets are combined and used to train the regression models from above. The final prediction errors are obtained by applying the regression model to the reduced features of the test set.

The results of these behavior predictions are shown in table 2.5. This table contains the results of the $N_{\mathrm{AE}} = 1$, $N_{\mathrm{AE}} = 2$ and $N_{\mathrm{AE}} = 3$ version of the AE models, as well as the results of the full feature set for comparison. Predictions are run with all three regression algorithms and the reference values again serve as a benchmark. Also, all three behavior features are used as labels. The scatter plots of the 1D case are shown in figures A.1, A.2 and A.3 in the appendix.

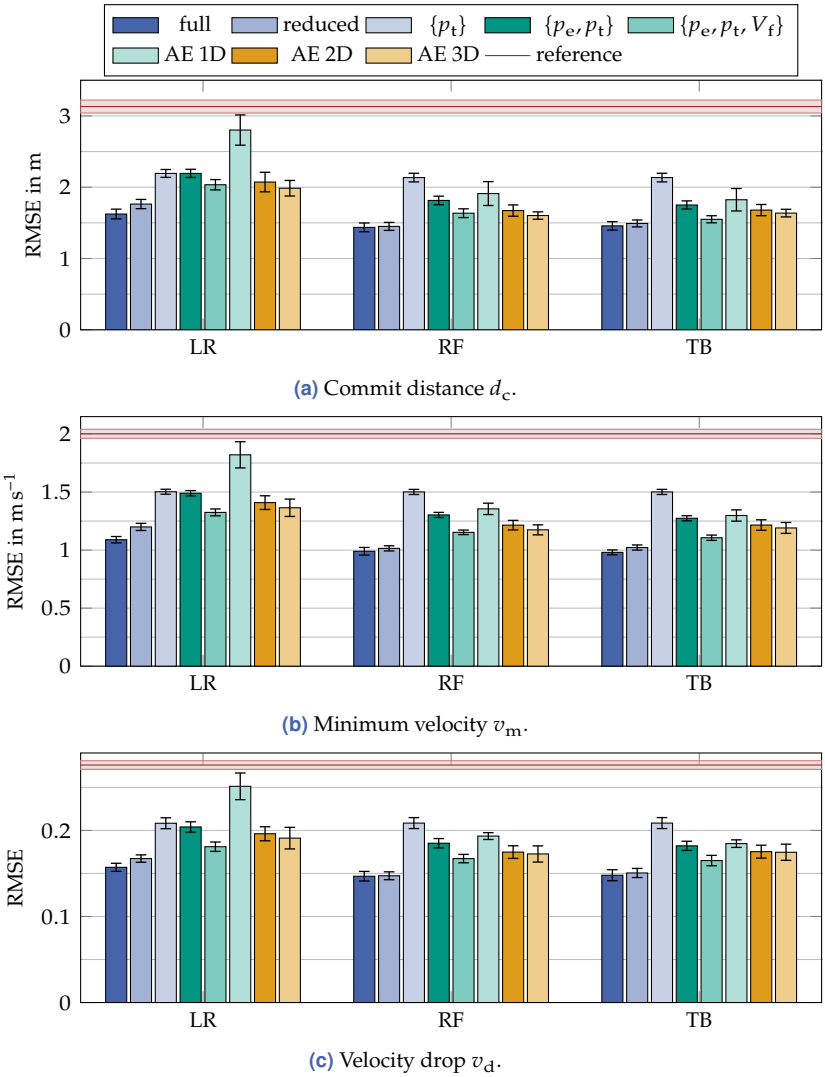
The performance shows similar patterns as the evaluations from before: LR is the worst of all regression methods while RF and TB are again similar in performance. In the 1D case TB outperforms RF in all cases slightly, in all 2D and 3D cases the roles are reversed and the predictions with RF are a little better than those with TB. All predictions are better than the reference method, the predictions, however, in the 1D case at the X-intersections show only moderately better results than the reference; this is especially the case for LR. The prediction improves with an increasing dimension of the latent complexity feature representation. The scatter plots in the appendix show relatively large deviations from the ideal line; this corresponds to the quantitative results from above.

Overall, these results meet expectations. The lower the dimensionality at the bottleneck, the more information is lost during compression and is thus not available during behavior prediction. The lower performance of the LR method compared to RF and TB can probably be attributed to the fact that LR is a linear method, i.e. it is only capable of finding linear dependencies between the input features[4] and the labels. If the relationship between the compressed complexity features and the behavior features is non-linear, LR as a linear method will not be able to find these relationships. The fact that the tree based methods are able to find non-linear dependencies is a reasonable explanation for their improved performances. Especially the 2D and 3D models give prediction performances that are better than those when using only two or three features directly, respectively. This further highlights the benefit of compressing all available features and thus keeping the information contained therein as compared to directly selecting a few promising intersection features.

2.5.5 Comparison of the Feature Sets

The experiments described in the previous sections show that the driving behavior can be predicted reliably. The models are trained with various intersection feature sets and achieve different results with them. To be able to better compare the results of all variants, figures 2.6 and 2.7 provide an overview on all results of the previous sections. They show the

[4] Here, the term *input features* refers to the compressed complexity features by the AE, in all other cases this describes the entire feature set that is currently used.

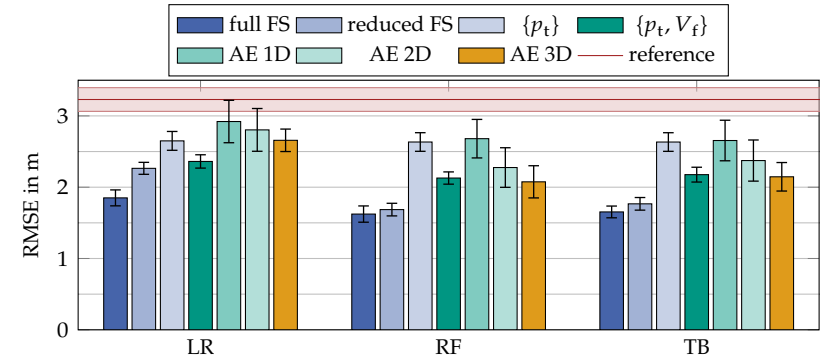


(a) Commit distance d_c.

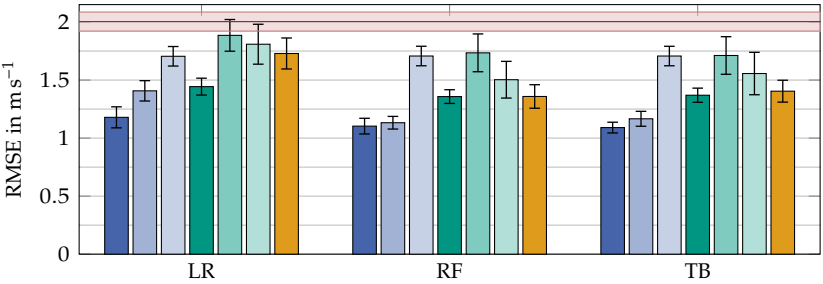
(b) Minimum velocity v_m.

(c) Velocity drop v_d.

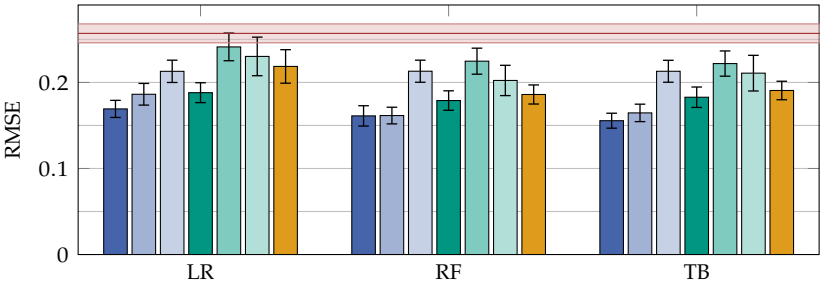
Figure 2.6 Behavior prediction at the T-intersections using different complexity feature sets, regression algorithms and behavior features as labels. For each label the result of the naive regression model is given as the reference for comparison.



(a) Commit distance d_c.



(b) Minimum velocity v_m.



(c) Velocity drop v_d.

Figure 2.7 Behavior prediction at the X-intersections using different complexity feature sets (FS), regression algorithms and behavior features as labels. For each label the result of the naive regression model is given as the reference for comparison.

results using the T-intersection and X-intersection data sets, respectively. For the T-intersections figure 2.6(a) shows all results where the commit distance d_c is used as the label, figure 2.6(b) displays the performance of the models with the minimum velocity v_m and figure 2.6(c) contains the results when the v_d is chosen as the label. Analogously, figures 2.7(a), 2.7(b) and 2.7(c) show the results when using d_c, v_m and v_d at the X-intersections, respectively. The performance of the reference method for prediction is displayed as a red line. As all variants for prediction are run 10 times with different training and test set assignments, the 1 σ standard deviation is given by the error bars as well. In case of the reference method the 1 σ standard deviation is shown by the light red area.

The standard deviation is small in most cases, which indicates a consistent prediction performance over the 10 set assignments. In case of the AE variants the standard deviation is notably larger compared to the remaining methods. LR is the worst prediction method for this application; it is outperformed by RF and TB. These two algorithms achieve very similar results for all investigated variants. It is further apparent that the prediction results at the T-intersections are better than those at the X-intersections. This can be seen by the fact that the RMSE values for the X-intersections data set are closer to the reference value than in case of the T-intersections. Also, the standard deviation is larger for the X-intersections. A likely cause is the smaller size of the X-intersections data set, which also contains data from fewer intersections.

Comparing the feature sets with each other, the full feature set has the lowest RMSE value in all cases and with all regression algorithms. This is followed by the reduced feature set which is second best in all cases. The order of the remaining sets is not consistent across all variants, however, the AE variants with $N_{AE} = 3$ are always better than the ones with $N_{AE} = 2$, which themselves outperforms those with $N_{AE} = 1$ in all cases. Both these results are as expected; it is plausible that a regression model that is based on more features performs better than one with fewer features. Also, having a latent representation with more dimensions enables an AE to keep more information from the original feature sets, thus enabling better predictions.

Using the T-intersections data set and RF or TB as the regression method, the performance of the 1D AE feature set is better than that of the best investigated feature sets that contains only a single feature, the one using only p_t. Similarly, the 2D AE feature set outperforms the best feature set from section 2.5.3 with two features (p_t and p_e) and the 3D AE variant achieves results very similar to the feature set using all three most important features (p_t, p_e and V_f). This is plausible as well, because the AE has access to the full feature set during its training and is therefore not limited to the information contained within a single feature but can combine information from all features into the single feature it generates. Similar arguments can be made for the 2D and 3D cases. This, however, is not the case when LR is used for prediction with the T-intersection data set. In that case the AE variants are not better than the feature sets with the same dimension and especially the 1D AE variant is notably worse than the models that only use the turning direction p_t for behavior prediction.

In case of the X-intersections data set only the two best performing features are investigated, c.f. section 2.5.3. The best model that uses only one of these two features (p_t) outperforms the 1D AE variants in all cases, i.e. all combinations of behavior feature and prediction algorithm. The same is true for the model that uses both most important features (p_t and V_f); it also outperforms the 2D AE in all cases. Even the 3D AE variants perform worse in most cases and are only able to be approximately equal to the models using the two most important features. One possible explanation for the different results between the T- and X-intersections data sets might be the smaller data set size of the X-intersections.

2.6 Conclusions

In this chapter an investigation is introduced that focuses on the influence of an intersection and its surroundings on the driving behavior of a human driver. Only inner-city intersections without traffic lights or signs are considered here. The investigation is based on data gathered from a field study in real-world traffic, during which the driving behavior of participants was recorded. Based on that data features are defined which describe the intersection itself, its surroundings, the traffic the

participant had to interact with and the path driven through the intersection. These features are assumed to define intersection complexity. The driving behavior of the participant is described by further behavior features.

The complexity features are used for training regression models to predict the behavior features. The analysis of the performances of these models shows that the driving behavior at intersections can indeed be predicted by only considering observable properties of and at an intersection, i.e. its complexity. This is especially noteworthy as no aspects of a driver's personality, mood or intentions are being considered. It therefore follows that there is a typical driving behavior for each intersection, within a certain range, which is shown by different drivers and which only depends on the complexity of an intersection. Further analyses with reduced feature sets show that the entire complexity feature set is not needed for reliable predictions; instead subsets of the full set or feature sets by a dimension reduction with an AE provide reliable predictions as well. However, the best performance is achieved with the full feature set.

As mentioned before, one of the important aspects is that it is desirable to be able to explain predictions to a human observer. This is especially relevant if such a prediction model is used as part of a decision-making system for automatic driving or if knowledge of typical driving behavior – described by behavior features – is incorporated into such systems. Doing so could entail using the observed complexity of an intersection and utilizing the knowledge about typical driving behavior that is associated with that complexity for decision-making. One could e.g. program an automatic vehicle to drive more offensively if it encounters a situation at an intersection in which human drivers are predicted to typically drive relatively slowly and cautiously. Alternatively, one could also design the algorithm in a way that it behaves similar to a human driver and have it drive defensively as well. The complexity features in their current form are, however, not ideal for either variant.

Ideally, a one-dimensional complexity feature that strongly correlates with the driving behavior could be defined for intersections. This would facilitate an easily and quickly understandable explanation and could thus be used in an AV for improved acceptance. All feature sets used in this chapter deviate from this ideal in some form: The cardinality of many

of the feature sets is too large, more than two features probably make it very difficult to quickly – without too much mental capacity – understand a decision or prediction. With that constraint and further excluding variants with relatively large RMSE values, only the 2D AE variant of the T-intersection and the feature set with the entry position p_e and the turning direction p_t at the T-intersections remain. The performance of the X-intersection feature sets with low cardinality is relatively poor. This can probably be attributed to the overall poorer performance of the X-intersection models.

The two remaining feature sets are problematic when looking at the understandability of predictions, too. The AE features are trained using the full feature set, it is however not possible for a human user to understand how these are calculated or how the feature values have to be interpreted as the AE does not necessarily learn a linear dependence of the input features and the latent representation [65]. The feature set with only p_e and p_t merely has six distinct prediction values (c.f. section 2.5.3), which might limit its applicability. In the next chapter a further study is introduced with which a complexity feature that is based on human perception can be defined.

3 Complexity Measure for Intersections

In the previous chapter the complexity of an intersection is described by a set of intersection features, which themselves describe certain properties of the intersections. Several variations of this feature set are used successfully for behavior prediction. However, as elaborated at the end of the previous chapter, these feature sets are not ideally suited to explain the reasons for a prediction to a human user. The two main factors that limit the usefulness of these sets are either their cardinality or, in case an AE version is used, the non-linear structure of the feature.

In this chapter a method is proposed by which a one-dimensional complexity score for intersections on an interval scale can be derived. The score is further based on how complex a given intersection is perceived by humans. This is done under the assumption that if a complexity score relates to the human perception of the complexity of an intersection, it might be more acceptable as an explanation for a decision compared to e.g. some features whose influence on the complexity is, at first glance, not obvious to an uniformed passenger. In other words, the explanation "the vehicle drives more cautiously because the intersection ahead is typically considered as highly complex by human drivers" is probably more acceptable than "the vehicle drives more cautiously because the intersection ahead is narrow, has poor visibility, the free volume at the intersection is small and the streets meet at a sharp angle".

This is achieved by first having human raters rank intersections by their perceived complexity. These rankings are then based on the complexity features from the previous chapter. With that approach a meta-feature, the complexity score, is introduced that is, like the AE variants, based on all features describing the intersection. Unlike the AE features, one can assume that this complexity score can be easily understood by humans because it is based on human perception of complexity. As the score

is on an interval scale, it is also ordered, i.e. a higher complexity score describes a more complex intersection. The score difference between two intersections is further a measure of their difference in complexity.

The complexity scores are derived from a second study in which participants were shown short videos of approaches towards intersections. They were then asked to compare two videos at a time and identify the more complex approach. To distinguish this study of the one from chapter 2, it is referred to as the *video study* in the following.

In this chapter, first the theory behind pairwise comparisons and especially their evaluation is introduced, see section 3.1 for that. As this chapter builds on the findings of the previous one and uses many of the concepts employed there, the regression methods and the considerations regarding complexity in traffic are not discussed again in this chapter. The introduction to these aspects is already given in sections 2.1.1 and 2.1.2, respectively. The video study itself is presented in section 3.2. The process of deriving the complexity scores from the data of the study is shown in section 3.3, alongside some early evaluations. These complexity scores are then used for behavior prediction in several variants, see section 3.4. Finally, some conclusions are drawn in section 3.5.

3.1 Pairwise Comparisons

The task of assigning a score to an object can be difficult for human raters. Oftentimes this is done by asking the rater to assign a score directly to the object in question, i.e. a Likert scale is used [32, 121, 125]. Using this approach, the scale is fixed on both ends by a descriptive label[1] [32] and the resulting scores lie between the two. A Likert scale, however, can be problematic in some cases: If different raters do not agree sufficiently in the scores they assign, the average scores of all raters might not be useful. E.g. Oishi et al. [121] found that Americans and Japanese people show similar scores on a questionnaire on individualism that uses a Likert scale. However, using pairwise comparisons showed that Americans do indeed show signs of being more individualistic.

[1] e.g. "Please rate the helpfulness of this example on a scale from 1 to 7 where 1 stands for *not helpful* and 7 for *very helpful*."

The alternative, which is used in this work, is to have the raters compare only two objects at a time. They are asked to select the object they associate a higher (or lower, depending on the investigated property) score with, and repeat this process several times with different pairs. This procedure is known as pairwise comparisons [25, 35]. Pairwise comparisons do not generate absolute scores, i.e. the position of the scores along the scale is not fixed [35, 38]. Therefore, it is only possible to judge an object relative to the remaining ones that are also part of the set of objects to be rated. If it is necessary, objects with a known or pre-defined score can be included into the set of objects to rate, thus anchoring the scores to a known value [145]. Finally, some previous works report that pairwise comparisons are often perceived as easier than using a Likert scale by the participants [32, 125].

The objects that are compared with each other can be many different things and also the attribute by which these objects are to be ranked can be a wide variety of aspects. For example, different foods (objects) could be ranked by their taste (property). In previous works pairwise comparisons were applied to many different problems, including education and teaching [33, 124, 145], rankings in sports [21, 29, 72, 87, 108, 164] or dominance in animal populations [5, 51, 118, 128, 162] among many more fields. The properties by which the objects were ranked cover a wide spectrum, including students' or competitors' abilities and fighting ability in the cases listed above. For consistency, in the following introduction there will only be *objects* that are ranked by their attribute *strength*.

Oftentimes the raters do not rate every possible pair of objects, as the number of comparison pairs N_c grows fast with an increasing number of objects N [35]:

$$N_\mathrm{c} = \binom{N}{2} = \frac{N(N-1)}{2}\,. \tag{3.1}$$

To limit boredom given the repetitiveness of the task and to avoid unreliable results or participants not finishing the study which could follow from that, participants can probably only rate a limited number of comparisons; a limit of 20 comparisons per participant is used in this work. As $N_\mathrm{c}\,(N=7) = 21$, this threshold is exceeded with only 7 objects. If there are many more objects than that in a study, a single rater can there-

fore realistically only judge on a subset of all possible comparisons. To be able to still obtain results for a large number of comparisons – and thus include more objects in a study – the answers of different raters are combined. The results are thus an average over the pool of participants.

The raters are only asked to identify the object which they associate a larger strength with. It is therefore sufficient to count the number of times an object is considered to be stronger than another object when accumulating the answers given by all raters. The results of all comparisons between the N_p objects are given by $\mathbf{R} \in \mathbb{N}^{N_p \times N_p}$. The individual elements of that matrix $r_{i,j}$ are the number of times object i is preferred over object j. If a comparison between two objects i and j does not occur, $r_{i,j} = r_{j,i} = 0$. Finally, $r_i = \sum_j r_{i,j}$ is the total number of comparisons in which object i is preferred over any other object.

In this thesis two methods to evaluate pairwise comparison data are used: The one after Bradley and Terry [25, 35] and the method first described by Elo [47]. Both are introduced in the following.

3.1.1 Bradley-Terry Model

An often used method to evaluate pairwise comparison experiments was introduced by Bradley and Terry [25] and will be referred to as the Bradley-Terry (BT) model. It has been utilized in many applications since its introduction. Also there exist several extensions to it. A selection of both is presented in the following: Whiting et al. [162] use the results of contests between male lizards to determine their fighting ability by applying a BT model. Zucco et al. [178] rank the Brazilian cabinet positions by their salience; this is based on two pairwise comparison surveys in which participants were asked to state which of two ministries they would prefer their party to hold. Clark et al. [32] investigate how physical strength is perceived from images of male bodies. For that they compare ratings on a Likert-scale to pairwise comparisons. The latter are examined by BT, Elo and a modified Elo version. Steedle and Ferrara [145] investigate the application of pairwise comparisons on essay scoring compared to rubric scoring. The pairwise comparisons are evaluated using a modified version of the BT model. Mattthews and Morris [106] use an extended version of the BT model to measure the experienced pain of haemodialy-

sis patients when they are administered different variants of treatments. The extension of the BT model by Cattelan et al. [29] is able to track performance over time; the model is applied to professional football and basketball results. They further implement a measure to take home field advantage into consideration. Baker and McHale modify a generalized version of the Bradley-Terry model so that it is able to track the strength of tennis players over time and apply the method to the match results of professional male tennis players [21]. McHale and Morton [108] present a BT model for predicting professional tennis matches, which outperforms predictions based on the official world rankings. Strobl et al. [147] extend the BT model so that the data set on which the model is based on is partitioned recursively using covariates. Models are then fitted for each subset. Crompvoets et al. [33] extend the BT model with an adaptive algorithm to select the next comparison using the uncertainty of the parameters of the model. Menke and Martinez [109] propose a variant of the BT model that is based on a single-layer neural network. They extend their model to enable it to rate individual players who compete in teams. The model is further able to take a home field advantage or the game duration, among further aspects, into consideration.

For the current application no extension to the BT model is needed and therefore only the basic form of it is presented. The main assumption of the BT model is that the probability that an object i is preferred over another object j can be expressed as [35]

$$\pi_{ij} = \frac{\pi_i}{\pi_i + \pi_j}. \tag{3.2}$$

Here, π_i is a score that describes the strength of object i and π_{ij} describes the probability that object i is preferred over object j. Besides using it to calculate the probability of the outcome of a comparison between two objects, it can also be employed to rank the objects. I.e. $\pi_i > \pi_j > \pi_k$ implies that object i has the highest strength, followed by objects j and k, respectively. It should be noted here that the π_i do not represent the true strength of the objects; this property is typically not known and can therefore only be estimated from observations [38]; in this case from pairwise comparisons between the objects. The model assumes that the scores are on a linear scale [35], i.e. greater differences in the scores imply a greater strength difference. It follows from that definition that

the numeric scores are not fixed, instead they can be placed arbitrarily by using a constraint; e.g. $\sum_i \pi_i = 1$ with $\pi_i \geq 0$ is used [35].

The scores π_i are estimated in an iterative process [35]:

$$\tilde{\pi}_i^{(k+1)} = r_i \sum_{i \neq j} \left(\frac{N_{c,i,j}}{\pi_i^{(k)} + \pi_j^{(k)}} \right)^{-1}. \tag{3.3}$$

The number of times objects i and j have been compared to each other is given by $N_{c,i,j} = r_{i,j} + r_{j,i}$ and the non-normalized score after the k-th iteration by $\tilde{\pi}_i^{(k)}$. To enforce the constraint of $\sum_i \pi_i = 1$ a normalization step is included after each iteration in this work, though David [35] remarks that it is only necessary after the final iteration:

$$\pi_i = \frac{\tilde{\pi}_i}{\sum_j \tilde{\pi}_i}. \tag{3.4}$$

The estimation is terminated after 50 iterations; this value proved to be sufficient for this application.

3.1.2 Elo Model

In this work also a second method to evaluate pairwise comparison data is used, the Elo model. It was introduced by Arpad Elo to rate chess players [47]. Given its origin, this method is intended to track a player's strength over time, i.e. ratings are updated regularly after e.g. a single match, a set period of time or a tournament [47]. This, of course, does not apply in a study of pairwise comparisons that have no temporal order, as is the case here. For comparison, the method is used anyway, though in a slightly adapted version. There exist further methods to track a player's (or, more general, an object's) strength over time such as the TrueSkill model [63] or the system introduced by Glickman [55]. These methods also take further aspects, such as the standard deviation of players' ratings or more than two players per match who compete in teams, into consideration. Here, however, only the Elo model is used due to its ease of use.

The Elo model has found many applications over the years. It is currently used in several different sports to rank individual players or teams.

Examples include chess and football, the Fédération Internationale des Échecs' (International Chess Federation, FIDE) world rankings[2] and the Fédération Internationale de Football Association's (International Federation of Association Football, FIFA) world rankings[3] are based on this system. Besides organized sports, where this system originates from, it is also applied to sports related questions in scientific settings. Hvattum and Arntzen use it to predict the outcome of football matches [72]. They compare different approaches and conclude that predictions based on Elo ratings are only outperformed by the odds offered by bookkeepers but also note that the difference in Elo ratings of two competing teams "is a highly significant predictor of match outcomes". Lasek et al. [87] apply the Elo model to men's football matches of the national teams and show that it outperforms the former version of the FIFA world rankings. Gásquez and Royuela [53] investigate how Elo rankings of national teams depend on aspects describing e.g. the economy and the demographics of the countries. They show that Elo rankings are better suited for that task than the previous FIFA rankings. Williams et al. [164] predict results of professional tennis matches using Elo scores.

Further works from literature include the application of the Elo model in adaptive educational systems [124] or to evaluate an online study in which street scenes had to be rated by their beauty [58]. Another field of research for which the Elo system is used is the dominance order within animal populations, Albers and de Vries [5] emphasize that it also uses the temporal information that is contained within the order in which encounters occur. Pörschmann et al. [128] use it to determine the male dominance in their investigation into the reproductive behavior of Galápagos sea lions; Neumann et al. [118] investigate the dominance hierarchies of macaques monkeys. Franz et al. [51] use a modified version of the Elo rating model to investigate the dominance hierarchies of baboons. Their modifications enable individual score updates for each animal. Further updates on the Elo method have been proposed: As the

[2] https://handbook.fide.com/chapter/B022017 (visited on 1 June 2023)

[3] men's rankings: https://digitalhub.fifa.com/m/f99da4f73212220/original/edbm045h0udbwkqew35a-pdf.pdf;
women's rankings: https://digitalhub.fifa.com/m/3d9cb1decbbb2ac7/original/Womens-World-Ranking-Procedures.pdf (both visited on 26 May 2023)

order of the comparisons influences the final scores, one can average multiple runs with different orders of the comparisons, if the order of the comparisons is irrelevant [32]. Doebler et al. [38] propose a multivariate extension of the Elo method.

In the following, the procedure to obtain Elo scores for this work is presented. As before, there are N_p objects that are compared against each other. The results of these comparisons are given in $\mathbf{R}$, where $r_{i,j}$ is the number of times object i is preferred over object j. For each object i the current Elo score E_i is saved. Initially, all objects are assigned the same score E_0. As the Elo model uses an interval scale [47], E_0 can be set arbitrarily. Its value only influences the score values that can realistically be assumed but does not influence the model any further.

For each pair of objects i and j, the Elo scores E_i and E_j are used to calculate the expected outcomes $\tilde{r}_{i,j}$ and $\tilde{r}_{j,i}$ of their comparisons (c.f. [32]):

$$\tilde{r}_{i,j} = N_{\mathrm{c},i,j} \frac{1}{1 + 10^{\frac{E_j - E_i}{d_\mathrm{E}}}}, \quad \tilde{r}_{j,i} = N_{\mathrm{c},i,j} \frac{1}{1 + 10^{\frac{E_i - E_j}{d_\mathrm{E}}}}. \tag{3.5}$$

$\tilde{r}_{i,j}$ is the expected number of times object i is preferred over object j, $N_{\mathrm{c},i,j} = r_{i,j} + r_{j,i}$ is the number of times objects i and j have been compared to each other. Parameter d_E as well as the selection of base 10 for the logistic function are further aspects that influence the range Elo score values E_i can assume but do not alter the general workings of the model [47, 124]. It can further be easily shown that $\tilde{r}_{i,j} + \tilde{r}_{j,i} = N_{\mathrm{c},i,j}$. The logistic function in (3.5) is commonly used (e.g. [32, 38, 58, 124]) to model the influence of the difference in ratings $\left(E_j - E_i\right)$. Elo [47] originally uses the normal distribution but already identifies the logistic function as a possible alternative.

The Elo model uses the comparisons to update its current estimates of the objects' ratings. For that the expected number of won comparisons $\tilde{r}_{i,j}$ is set in relation to the actual number of comparisons $r_{i,j}$ object i won over object j. If one of the objects exceeded its expectations, i.e. it won more comparisons than expected, it is assumed to be underrated and thus its Elo score has to be increased; if it won fewer comparisons than expected, the Elo score is reduced [47]:

$$\Delta E_i = k_\mathrm{E} \left(r_{i,j} - \tilde{r}_{i,j}\right), \quad \Delta E_j = k_\mathrm{E} \left(r_{j,i} - \tilde{r}_{j,i}\right). \tag{3.6}$$

ΔE_i is the update to the Elo score of object i and k_E is a parameter that determines the influence a single comparison has on the overall score. The larger this parameter is, the more a single comparison influences the Elo score. A large value of k_E can also lead to instability in the scores while a small value can lead to slow convergence [124]. There are several strategies to set k_E depending on the current Elo score [124] or have it depend on further predictor variables [51]. In this work, however, a constant value is used.

Depending on the updating strategy, the scores E_i are updated regularly or the score updates ΔE_i are accumulated for some time. In both cases the new score is obtained by [47]:

$$E_{i,k} = E_{i,k-1} + \Delta E_i, \quad \forall i. \tag{3.7}$$

In the typical application, where the strength of some objects is tracked over time, this procedure is run once for each comparison or set of comparisons. Here, the Elo scores are only updated after all comparisons have been evaluated. Further, the full set of available comparisons is applied multiple times until the Elo scores converge to constant values. A similar approach to initialize a rating among a pool of unrated objects has already been suggested by Elo [47]. Alternatively, one can update the scores after each comparison and average multiple scores that result from different, randomly generated orders of the comparisons [32]. Such strategies are only necessary if the order does not contain any information, i.e. the comparisons are assumed to have happened simultaneously. If the order is relevant to e.g. track the performance of chess players, the updates should happen more regularly to keep this temporal component. Also, the comparisons can only be applied once as the sequential information is otherwise lost.

In this thesis the parameters for the Elo model are set to these values: The initial Elo score equates to $E_0 = 1600$, $d_\mathrm{E} = 400$ and $k_\mathrm{E} = 10$. Finally, in many applications objects enter or leave the rating pool over time. Several strategies can be employed to deal with those scenarios. As this is irrelevant for this work, it is not discussed here.

3.2 Video Study

The first goal of this chapter is to establish a complexity score for intersections. During the field study described in chapter 2, the participants were already asked to directly rate the perceived complexity of some of the intersections on an absolute scale. Yet, no consistent score could be generated from these ratings. For that reason, a pairwise comparison approach was selected for a subsequent study, as intersections do not have to be compared to a (possibly hard to define or communicate) common baseline that is ideally constant among all participants. Instead intersections are only compared to themselves.

The data for the complexity scores was recorded through an online study of pairwise comparisons and was conducted between December 2021 and February 2022. During that study the participants were shown short videos of approaches to intersections from a driving vehicle's perspective. In the following this study will be referred to as the *video study*. The videos for the study were taken at the same intersections that were also part of the field study of chapter 2. Online studies are a common tool; e.g. Goodspeed [58] used it to assess the beauty of Google Street View images through pairwise comparisons. Clark et al. [32] also conducted a pairwise comparison study and compared an online version of their study to a laboratory version. They report comparable results of both methods.

The videos were recorded while driving with the measurement vehicle in order to additionally record the same data as for the field study. That way the intersection features of the field study could be calculated for these runs as well. To generate consistent and comparable videos, the vehicle was stopped at the intersections. Most intersections were approached from all directions and all videos have in common that there is no additional traffic. The latter aspect improves the comparability of the videos as it is difficult to classify real world traffic consistently. It also helps to limit the number of videos that are used in the study and thus reduces the amount of comparisons. Also, not all combinations of entry position p_e and intersection of the field study were included in the video study and none of the X-intersections were included either. The study finally consisted of 29 videos resulting in 406 possible comparisons. The videos for the study did not contain sound and were cut in such a way

that they were each 8 s long and showed the approach to the intersection until the vehicle was nearly stationary. Additionally, each video started and ended with 1 s of blacked out screen, making the videos 10 s long in total.

These videos were then used for the video study, which had been approved by the ethics commission of the KIT. On the start page of the online study the potential participants were informed about the study. They could only proceed with the study if they consented to participate. Then they were asked for some demographic information. After that, the first of 20 pairwise comparisons was shown to the participants. After answering this, the next comparison was provided, and so on until all 20 comparisons were completed. The page for the comparisons showed the two videos in question next to each other with a headline identifying them as *Video A* and *Video B*, respectively, and a single task was shown above the videos: "Please watch the displayed situations in both videos one after the other. Then select the video that shows the more complex situation to you."[4] Below the videos it was asked: "Which video did you perceive as more complex?"[5] The participants could then either check a box labeled *Video A* or one labeled *Video B*. For each comparison page one of the possible 406 comparisons was randomly selected. Each participant was not shown a comparison more than once. After finishing the study, the participants had the chance to win ten prices of 20 Euros each. The participation was again voluntary and was not part of the actual study.

Not all participants answered all 20 comparisons, some ended their participation early. In the following only the data of those who answered all comparisons is reported. This includes 66 participants (52 identified themselves as male, 14 as female and none as diverse). They were on average 29.1 yr ($\sigma = 10.0$ yr) old.

The 1320 comparisons are combined into the answers matrix $r_{i,j}$. As there are 29 videos in the study, $\mathbf{R}$ is a square matrix of dimension 29. Because there were no comparisons of videos with themselves, the main diagonal is zero. Besides the complexity ratings, also the sensor measure-

[4] The study was conducted in German, the original version of the question was: "Bitte sehen Sie sich die dargestellten Situationen in den beiden Videos nacheinander an. Wählen Sie dann das Video aus, das für Sie die komplexere Situation darstellt."

[5] "Welches Video empfanden Sie als komplexer?"

ments of the 29 runs for the videos are again available. From these the features described in section 2.3 are calculated. To distinguish them from those of the runs of the field study, they are identified by the superscript $(\cdot)^{\mathrm{v}}$ when this distinction is important. Likewise, those of the field study are identified by $(\cdot)^{\mathrm{f}}$.

As it is now important to reference individual intersections, they are named according to the following scheme: The T-intersections are numbered from "T01" to "T13", the intersection where the deadlock situation was generated during the field study is also included in the video study and is identified as "TD". 12 of the 14 (numbers 9 and 13 are missing) T-intersections are part of the video study with at least one video from one of the three entry positions. The entry position ("L": left, "B": bottom or "R": right) is added to the intersection names for clear identification of the videos. Some entry position/turning direction combinations are excluded from the video study to limit the overall number of comparison in the study.

Of the possible 406 comparisons all but 16 appeared in the study at least once, 5 comparisons were shown 8 times and one comparison was even shown to 10 participants. Of the 350 comparisons that were shown at least twice 75.71 % had a clear winner, i.e. the ratio of won comparisons for the more complex video was larger than 0.6. Even when setting this threshold to 0.8, the percentage of comparisons that have a clear winner is still at 39.71 %. The fact that many comparisons were rated mostly the same among the different raters implies that the complexity of the videos differed substantially in many cases and that a winner was not only chosen by the participants because a tie was not available as an answer. These results further indicate that the responses were rather consistent among participants.

Video TDB appeared least often with 72 times, while video T02L appeared 113 times, which makes it the one that was shown most often. As the selection of the comparisons occurred using an equal distribution, also the probability with which a video appeared is drawn from an equal distribution. The differences between the expected and the sampled distributions can be explained with the relatively low number of participants for the number of possible comparisons.

Table 3.1 Data sets used in chapter 3. For each data set the number of combinations of intersections and entry positions p_e as well as the number of runs is given.

name	description	# int./p_e	# runs
$\mathcal{D}^F$	field study runs (T-intersections)	34	1818
$\mathcal{D}^V$	video study runs	29	29
$\mathcal{D}^F_{iv}$	runs from $\mathcal{D}^F$ with a corresponding intersection/p_e combination in $\mathcal{D}^V$	25	1206
$\mathcal{D}^F_{ov}$	runs from $\mathcal{D}^F$ without a corresponding intersection/p_e combination in $\mathcal{D}^V$	9	612

3.3 Complexity Score

Using the data from the video study, the complexity scores based on the BT and Elo methods can now be calculated. In this section first the results found are presented and discussed, then models are trained to base these scores on the intersection features. This is necessary for using the complexity scores outside the context of the video study. In this section, several data sets are used, all are listed in table 3.1. The data set that comprises of the 29 runs from the video study is referred to as $\mathcal{D}^V$, the T-intersection data set from chapter 2 as $\mathcal{D}^F$.

3.3.1 Intersections Ranked by Complexity

The BT and Elo scores are calculated as introduced in section 3.1 from the video study's response matrix $\mathbf{R}$ which is based on the data set $\mathcal{D}^V$. The scores of all videos are shown in table 3.2, the entries are ordered by their complexity rank. Both methods, BT and Elo, result in the same ranking. This is in line with findings from literature that report the similarity between the two scores [32, 55, 124]. Clark et al. [32] further find that their modified Elo score and the logarithm of the BT score are almost perfectly correlated. The modified Elo score in their work is the mean of the scores from several evaluation runs where the sequence of the comparisons is shuffled. This is done to avoid the dependence of the final Elo score on the order of the comparisons. In this work a similar strategy, see section 3.1.2, is utilized, thus the correlation is reproduced here.

Table 3.2 Resulting BTL and Elo scores for all intersections. The intersections are ranked from most complex to least complex. The rank of all intersections is identical for both BTL and Elo.

rank	video	π^v_{In}	E^v_{In}	rank	video	π^v_{In}	E^v_{In}
1	T07B	0.1030	1834.0	16	T12L	0.0249	1587.3
2	T10B	0.0713	1770.1	17	T05B	0.0244	1584.1
3	T07L	0.0592	1737.8	18	T04B	0.0234	1576.8
4	T10L	0.0590	1737.2	19	T10R	0.0224	1569.4
5	TDL	0.0584	1735.4	20	T12R	0.0222	1567.6
6	TDB	0.0573	1732.2	21	TDR	0.0220	1566.3
7	T08B	0.0537	1720.8	22	T06B	0.0204	1553.0
8	T03B	0.0517	1714.2	23	T04L	0.0194	1544.3
9	T04R	0.0432	1683.0	24	T05R	0.0182	1533.0
10	T12B	0.0389	1664.8	25	T06R	0.0175	1526.4
11	T11B	0.0375	1658.4	26	T01B	0.0086	1402.6
12	T05L	0.0364	1653.4	27	T11R	0.0064	1352.0
13	T02L	0.0364	1653.3	28	T01L	0.0051	1312.4
14	T02B	0.0286	1611.6	29	T01R	0.0029	1213.1
15	T08R	0.0276	1605.6				

Pearson's correlation coefficient [27] between two vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as follows:

$$c(\mathbf{x}, \mathbf{y}) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}, \tag{3.8}$$

where x_i and y_i are the i-th component of the vectors and $\bar{x}$ and $\bar{y}$ are the mean of all vector components. With that the correlation coefficient between the vector of all Elo scores $\mathbf{E}^{\mathbf{v}}_{\text{In}}$ of the 29 runs of the video study, i.e. those of the data set $\mathcal{D}^{\text{V}}$, and the corresponding vector of the logarithmic BT scores $\ln\left(\boldsymbol{\pi}^{\mathbf{v}}_{\text{In}}\right)$ is $c\left(\mathbf{E}^{\mathbf{v}}_{\text{In}}, \ln\left(\boldsymbol{\pi}^{\mathbf{v}}_{\text{In}}\right)\right) = 1.0000$. For comparison, the correlation coefficient between the Elo scores and the BT scores is $c\left(\mathbf{E}^{\mathbf{v}}_{\text{In}}, \boldsymbol{\pi}^{\mathbf{v}}_{\text{In}}\right) = 0.8878$. Both values confirm the findings by Clark et al. [32] and also that the initial impression from table 3.2, i.e. that the scores are

very similar, is true. Despite their apparent similarity, both scores are used for the analysis in the following.

Looking at the scores in more detail, one finds that the scores for the videos rated most and least complex, respectively, differ substantially, indicating that there are indeed notable differences in the perceived complexity and that the scores are not predominately determined by random noise. When looking more closely on the order derived from the complexity scores, the order appears to be plausible: intersections 7 and 10 and the one with the deadlock occupy the first six places in the ranking, i.e. they are rated to be the most complex. These intersections all have in common that they are in areas with buildings directly at the sidewalks and that there are many parked vehicles by or on the streets. Both factors limit the visibility, the free volume at the intersection and the width features. Also, entering from the bottom or left appears to be perceived as more complex than entering from the right. The most complex video that shows an entry from the right is only on the ninth rank. This is in line with the findings from the previous publication on the field study [190]. There it is shown that entering from the right resulted in more offensive driving behavior than entering from the bottom or left. These findings are not discussed in detail in this thesis. The videos that received the lowest complexity scores are in line with these aspects as well, especially intersection 1 is outside the city center, there are several meters of lawn between the curbs and the fences or hedges and the buildings are mostly even further away from the street. There are also fewer vehicles parked on or at the street. This contributes to good visibility and much space at the intersection which in turn likely decreases the perceived complexity.

To further investigate which intersection features $\mathcal{F}$ influence the perceived complexity to what extend, correlation coefficients between the complexity scores and the continuous intersection features are calculated for the 29 runs of the video study. For both the BT and the Elo scores correlation coefficients are calculated, $c_{\mathrm{BT}}\left(\mathcal{F}^{\mathrm{V}}\right) = c\left(\boldsymbol{\pi}_{\mathrm{In}}^{\mathrm{v}}, \mathcal{F}^{\mathrm{V}}\right)$ describes the correlation coefficients of the BT scores $\boldsymbol{\pi}_{\mathrm{In}}^{\mathrm{v}}$ with one of the intersection features $\mathcal{F}^{\mathrm{V}}$, the corresponding Elo variant is calculated with $c_{\mathrm{Elo}}\left(\mathcal{F}^{\mathrm{V}}\right) = c\left(\mathbf{E}_{\mathrm{In}}^{\mathrm{v}}, \mathcal{F}^{\mathrm{V}}\right)$. Both c_{BT} and c_{Elo} are given in table 3.3.

Additionally, this table also contains the correlation coefficients for all field study runs through the T-intersections that are part of the analysis

Table 3.3 Correlation coefficients (c_{BT} and c_{Elo}) between the BT and Elo complexity scores ($\boldsymbol{\pi}^{\mathrm{v}}_{\mathrm{In}}$ and $\mathbf{E}^{\mathrm{v}}_{\mathrm{In}}$, respectively) and some intersection features $\mathcal{F}^{\mathrm{V}}$ using the data from the 29 runs of the video study. The last three columns contain the correlation coefficients (c_{CD}, c_{MV}, c_{VD}) between the behavior features ($\mathbf{d}^{\mathrm{F}}_{\mathrm{c}}$, $\mathbf{v}^{\mathrm{F}}_{\mathrm{m}}$ and $\mathbf{v}^{\mathrm{F}}_{\mathrm{d}}$) and some of the intersection features $\mathcal{F}^{\mathrm{F}}$ of the 1818 runs through the T-intersections of the field study data set $\mathcal{D}^{\mathrm{F}}$. The correlation coefficients are calculated for the vectors that combine the data from all runs of the study in question.

features		c_{BT}	c_{Elo}	c_{CD}	c_{MV}	c_{VD}
V_{f}	free space	−0.61	−0.64	0.34	0.40	0.40
α_{s}	minimum angle	0.43	0.49	−0.16	−0.16	−0.23
$d^{\mathrm{y}}_{\mathrm{v,c}}$	vis. distance (PC) at the YS	−0.40	−0.38	0.36	0.33	0.27
$w_{\mathrm{v,b}}$	visible width before int.	−0.38	−0.49	0.42	0.37	0.33
$d_{\mathrm{v,c}}$	vis. distance (PC)	−0.38	−0.36	0.30	0.29	0.28
$d^{\mathrm{y}}_{\mathrm{v,p}}$	vis. distance (OP) at the YS	−0.30	−0.42	0.21	0.17	0.13
n_{vi}	# interaction vehicles			−0.24	−0.37	−0.36
n_{rw}	# veh. with right of way			−0.20	−0.33	−0.36
n_{v}	# visible vehicles			−0.20	−0.33	−0.31

in chapter 2, i.e. the runs of data set $\mathcal{D}^{\mathrm{F}}$. As there are no complexity scores available for these runs, instead the correlation coefficients between the vectors of intersection features $\mathcal{F}^{\mathrm{F}}$ and the behavior features, i.e. the vectors of the commit distance $\mathbf{d}^{\mathrm{F}}_{\mathrm{c}}$, the minimum velocity $\mathbf{v}^{\mathrm{F}}_{\mathrm{m}}$ and the velocity drop $\mathbf{v}^{\mathrm{F}}_{\mathrm{d}}$, are calculated:

$$c_{\mathrm{CD}}\left(\mathcal{F}^{\mathrm{F}}\right) = c\left(\mathbf{d}^{\mathrm{F}}_{\mathrm{c}}, \mathcal{F}^{\mathrm{F}}\right), \tag{3.9}$$

$$c_{\mathrm{MV}}\left(\mathcal{F}^{\mathrm{F}}\right) = c\left(\mathbf{v}^{\mathrm{F}}_{\mathrm{m}}, \mathcal{F}^{\mathrm{F}}\right), \tag{3.10}$$

$$c_{\mathrm{VD}}\left(\mathcal{F}^{\mathrm{F}}\right) = c\left(\mathbf{v}^{\mathrm{F}}_{\mathrm{d}}, \mathcal{F}^{\mathrm{F}}\right). \tag{3.11}$$

Not all intersection features are included in the table, but only those continuous features with at least one of the correlation coefficients $c > 0.3$. Additionally, some intersection features also have to be excluded from the evaluation of the complexity scores because they are irrelevant for the video study: All features describing the traffic at the intersection cannot be used as the videos only show situations without any traffic. As the videos are cut shortly before the vehicle reaches the intersection, only the widths before the intersection can be used. Those describing

the width after the intersection cannot be used for this analysis, as the participants of the video study have no way of knowing which way the vehicle eventually turned during the run through the intersection.

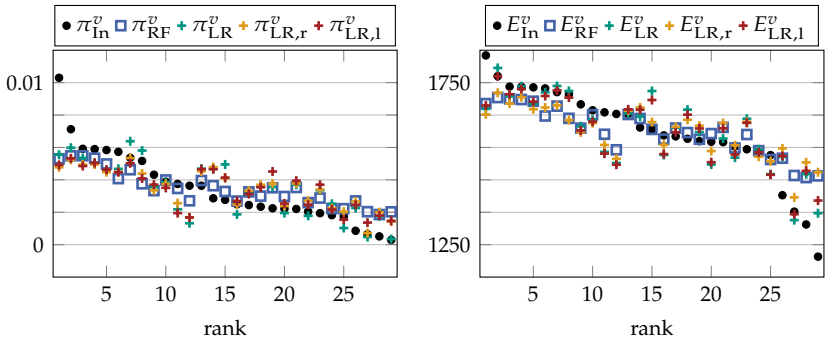
Most of the correlations between the complexity scores and the intersection features are in line with the expected patterns, i.e. the free volume at the intersection V_f, all three visibility distances ($d^\mathrm{y}_\mathrm{v,c}$, $d_\mathrm{v,c}$ and $d^\mathrm{y}_\mathrm{v,p}$) and the visible width before the intersections ($w_\mathrm{v,b}$) are negatively correlated with both complexity scores, indicating larger complexity scores when the intersection is denser, narrower and has poorer visibility.

However, these results have to be interpreted with some caution. Most importantly, a correlation does not necessarily imply causation. This can be seen in the case of the minimum angle α_s, which correlates notably with both complexity scores, however the correlation coefficients indicate a lower complexity score when the minimum angle at the intersections is smaller. This contradicts expectations, as a smaller angle between two streets likely limits visibility. On closer inspection of the intersections and the distribution of that feature this phenomenon can be explained: There are only four intersections, of the twelve in total, where the angle is notably smaller than 90°, of which the two most extreme cases, with minimum angles in the range of 65°, are intersections 11 and 12. The remaining two intersections have a minimum angle of approx. 74° (intersection 1) and 83° (intersection 4). None of these intersections are ranked as overly complex, cf. table 3.2; instead, the four videos ranked least complex are all at two of these intersections. Their relatively low complexity ratings likely contribute to the positive correlation coefficient. From these results alone it is therefore not possible to determine if the low complexity ratings are caused by the sharp angle between the streets or if the observed low correlation is rather caused by the remaining features (where the correlation is more in line with general expectations). An argument for the latter is that the remaining intersections, which vary considerably in their complexity rating, all have a minimum angle that is close to 90° which limits the discriminative ability of this feature. Also, the distributions of the remaining features are far less extreme in the sense that they are spread out more and are in general not as concentrated around a single value. However, some outliers are still present. From table 3.2 it is apparent that especially the free volume and the

visibility distances have the strongest correlations, indicating their likely importance for the complexity ratings.

The second aspect of this analysis concerns the influence the intersection features have not only on the complexity ratings but also on the actual driving behavior. The intersection features viewed above all show the same general trends on the correlation coefficients with the behavior features: The denser, narrower and the poorer the visibility conditions, the more defensive the driving behavior, i.e. the correlation coefficients are positive. For most intersection features the magnitude of the correlation features are similar between the two complexity scores and the three behavior features. Comparing the correlation coefficients of the complexity scores with those of the behavior features however, one observes considerable deviations for some of the intersection features. There are several likely reasons for the latter observation: First, the underlying distributions of the intersections differ. In the video study each intersection/entry position combination is considered once, while in the field study some combinations were driven much more often than others which results in a larger influence of these combinations on the correlations. Also, each of the two data sets contain combinations that are not included in the other data set. Second, different aspects are considered here. In case of the video study data set $\mathcal{D}^{\mathrm{V}}$ the influence of the intersection features on the complexity scores is studied, but for the field study data set $\mathcal{D}^{\mathrm{F}}$ the correlations are calculated between the intersection features and the driving behavior. Even though an influence of the perceived complexity scores on the driving behavior is likely, they are probably not equivalent as the driving behavior might also be influenced by further factors like a driver's mood. Differences in the correlation coefficients might therefore be in part explained by such aspects as well.

In conclusion it can be said that the existence of a complexity score for intersections that describes the perceived complexity there is very likely. This is supported by the reasonable order of the videos and because the correlations between the intersection features and the complexity scores are mostly as one would expect them to be. Also, the correlation coefficients between the intersection and the behavior features follow the same pattern as those for the complexity scores. This further supports the existence of a complexity score under the assumption that the perceived



(a) BT scores. **(b)** Elo scores.

Figure 3.1 True complexity scores ($\pi^{\text{v}}_{\text{In}}$, E^{v}_{In}) of the 29 runs of the video study data set $\mathcal{D}^{\text{V}}$. The runs are ordered by their true complexity, the most complex run is ranked first. Also, the reconstructed scores using the intersection features of these runs are shown. These include those estimated by the RF models ($\pi^{\text{v}}_{\text{RF}}$, E^{v}_{RF}), by the LR models ($\pi^{\text{v}}_{\text{LR}}$, E^{v}_{LR}), by the lasso variant LR models ($\pi^{\text{v}}_{\text{LR,l}}$, $E^{\text{v}}_{\text{LR,l}}$) and the ridge variant LR models ($\pi^{\text{v}}_{\text{LR,r}}$, $E^{\text{v}}_{\text{LR,r}}$).

intersection complexity actually influences the driving behavior. This assumption is investigated further in the following sections.

3.3.2 Complexity Reconstruction Using Intersection Features

So far, the complexity scores are only available for the 29 runs that make up the video study data set $\mathcal{D}^{\text{V}}$. To obtain a complexity score for any other intersection, to e.g. use it for behavior prediction, a method is needed to calculate the scores for any run through, ideally, any intersection without the need to assess its complexity through additional studies. One way to achieve that is to base the complexity scores on the intersection features. If this methodology were to be implemented in an automatic vehicle, the data required to calculate these features would very likely be available anyways from the onboard sensors. In this work LR and RF ($N_{\text{t}} = 300$ and $N_{\text{L}} = 5$) regression models are trained for that. The score prediction using these models is based on the intersection features selected for the correlation coefficients between the complexity scores and the intersection features. The reasons for excluding some features are

the same here. Additionally, the entry position is used as a feature now as well. The turning direction cannot be utilized as the videos were cut before the participants of the video study could infer where the vehicle was going. This results in the following selection of intersection features: The enter position p_{e}, the free volume at the intersection V_{f}, the minimum angle between the street at the intersections α_{s}, the average street width $w_{\mathrm{s,b}}$, the average visible width $w_{\mathrm{v,b}}$ and the average available width $w_{\mathrm{a,b}}$; the latter three only use the variant considering the area before the intersection. In addition, the number of trees n_{t} and the visibility distances using the point clouds and the polygons, both in the normal and in the yield variants ($d_{\mathrm{v,c},i}$, $d^{\mathrm{y}}_{\mathrm{v,c}}$, $d_{\mathrm{v,p},i}$ and $d^{\mathrm{y}}_{\mathrm{v,p}}$, respectively) are used.

To train the models, the original scores obtained by the BT and Elo methods (π^{v}_{In} and E^{v}_{In}, respectively) for all video study runs are used as the labels and the intersection features $\mathcal{F}^{\mathrm{V}}$ serve as predictors for the regression models. As $\mathcal{D}^{\mathrm{V}}$ is small, all 29 runs are used to train the models. This runs the risk of overfitting the models to the training set but it is done regardless because the reconstruction of the complexity scores described in this section is not the main goal. Essential is rather that the behavior prediction that is based on these reconstructed scores should be seen as the main metric by which to judge the validity of this approach.

Figure 3.1 shows π^{v}_{In} and E^{v}_{In} as well as their reconstructions using the RF (π^{v}_{RF} and E^{v}_{RF}) and LR (π^{v}_{LR}, E^{v}_{LR}, $\pi^{v}_{\mathrm{LR,l}}$, $E^{v}_{\mathrm{LR,l}}$, $\pi^{v}_{\mathrm{LR,r}}$ and $E^{v}_{\mathrm{LR,r}}$) models, respectively. These results show that the complexity scores of most runs are reconstructed with relatively small deviations. The more extreme scores, however, are reconstructed less reliably and especially the RF models show the tendency to overestimate small complexity scores and underestimate large scores.

3.4 Behavior Prediction Based on Complexity Scores

Now the complexity scores can be used to predict the driving behavior. To this end the runs of the field study need to be assigned a complexity score. There are two possible ways to do that: First, the scores of the video study can be assigned directly to the runs from the field study (π_{In} and

E_{In}). For that the complexity score of the run of the video study data set $\mathcal{D}^{\text{F}}$ is selected for which the intersection/p_{e} combination matches. The second option is to use the models of the previous section to calculate the complexity scores based on the intersection features. For that purpose the models that use the LR algorithm (π_{LR} and E_{LR}), the LR ridge variant ($\pi_{\text{LR,r}}$ and $E_{\text{LR,r}}$), the LR lasso variant ($\pi_{\text{LR,l}}$ and $E_{\text{LR,l}}$) and RF (π_{LR} and E_{RF}) are available to generate the complexity scores.

The first method of assigning complexity scores to the field study runs is only applicable to those runs for which a run of the video study with the corresponding intersection/p_{e} combination is available. As each combination is present in the video study only once, the assignment is unambiguous if it is possible at all. Therefore, only runs can be considered where the combination in question is present in both studies. This constitutes the data set $\mathcal{D}^{\text{F}}_{\text{iv}}$. The analysis with this data set is presented in section 3.4.1. This also avoids evaluating intersections that were not used for training the complexity score models. However, it is certainly an important question whether these models are able to generalize to further intersections, i.e. if the complexity scores calculated for intersection/p_{e} combinations that were not part of the video study are still useful for behavior prediction. Therefore, the complexity score models are applied to those runs of the field study that were not part of the initial evaluation, i.e. runs where the combination was only part of the field study and not of the video study, which form the data set $\mathcal{D}^{\text{F}}_{\text{ov}}$. The performance of models trained with these runs are presented in section 3.4.2. This is possible as the complexity score models are applicable to all runs for which the intersection features are available, regardless if their intersection/p_{e} combination was part of the video study or not.

3.4.1 Intersections of Video and Field Study

This section only considers those runs of the field study for which a corresponding run in the video study is available, i.e. the data set $\mathcal{D}^{\text{F}}_{\text{iv}}$. It is thus smaller than the data set $\mathcal{D}^{\text{F}}$ and contains only 1206 runs through T-intersections. In the following, the prediction error (RMSE) when using the commit distance d_{c} as the label is reported in table 3.4. The results when using the minimum velocity v_{m} and the velocity drop v_{d} do not differ in a substantial way and are shown in appendix B. The predictions

are performed as before: The data set $\mathcal{D}_{\text{iv}}^{\text{F}}$ is split into a training, test and, if required, a validation set using the ratios from chapter 2 and the model in question is trained and evaluated with them. This process is repeated 10 times with different set assignments here as well. The standard deviations are not reported here as they are relatively small and are in the same order of magnitude as in chapter 2. The behavior regression is again performed using LR, RF and TB. The lasso and ridge variants of LR are not used for behavior prediction as their predictions are very similar to those of the base LR method. In addition, the scatter plots of the best performing RF models trained with selected feature sets are reported in the first row of figure 3.2.

As the current data set $\mathcal{D}_{\text{iv}}^{\text{F}}$ contains fewer runs than the T-intersection data set $\mathcal{D}^{\text{F}}$, the results of the models trained using the feature sets considered previously in chapter 2 are not directly comparable. Thus the training with these feature sets is run again, but with the smaller data set $\mathcal{D}_{\text{iv}}^{\text{F}}$ of this section. The results of that are shown in the first rows of table 3.4. This table further shows the prediction performance if the complexity score variants, either those assigned directly (π_{In} and E_{In}) or those reconstructed with the LR (π_{LR} and E_{LR}), the LR variants ($\pi_{\text{LR,r}}$, $E_{\text{LR,r}}$, $\pi_{\text{LR,l}}$ and $E_{\text{LR,l}}$) or the RF (π_{RF} and E_{RF}) models, are used as the only input features. These ten cases are shown in the second set of rows. The bottom rows finally contain the performance of models that are trained using these complexity scores in combination with the turning direction p_{t}; these models therefore have two input features available. Using p_{t} is possible as this feature describes an aspect of driving that the participants of the video study were not able to take into consideration as the videos ended before the vehicle went through the intersection. Thus it cannot be used to calculate the complexity scores. As p_{t} is therefore independent from the complexity scores, it can be used alongside them for behavior prediction.

The results show that the behavior prediction errors using the feature sets of the previous chapter are very similar for both data sets, i.e. $\mathcal{D}^{\text{F}}$ and $\mathcal{D}_{\text{iv}}^{\text{F}}$. It is further evident that LR has again the worst performance of all regression methods. It has the largest RMSE for all feature sets where the error is notably smaller than that of the reference method. Only for those feature sets with which the predictions are relatively unreliable, i.e.

Table 3.4 Behavior prediction performance RMSE of different regression algorithms using the data set $\mathcal{D}_{\text{iv}}^{\text{F}}$. The commit distance d_{c} is used as the behavior feature here, thus all results are in m. Using the reference method results in RMSE = 3.048 m. The top rows show prediction results with some of the feature sets from chapter 2 for comparison. The middle rows show the prediction performance when using the BT/Elo complexity score variants as a single feature, the results using these score variants in combination with the turning direction p_{t} are given in the bottom rows.

feature set	LR	RF	TB	feature set	LR	RF	TB
full FS	1.661	1.478	1.475	$\{p_{\text{t}}, p_{\text{e}}\}$	2.178	1.811	1.770
red. FS	1.772	1.479	1.503	$\{V_{\text{f}}, p_{\text{t}}, p_{\text{e}}\}$	2.059	1.618	1.568
$\{p_{\text{t}}\}$	2.215	2.145	2.145	AE 1D	2.649	1.972	1.936
$\{V_{\text{f}}, p_{\text{e}}\}$	2.466	2.136	2.170	AE 2D	2.240	1.631	1.645
$\{V_{\text{f}}, p_{\text{t}}\}$	2.099	1.887	1.909	AE 3D	2.015	1.606	1.656
$\{\pi_{\text{In}}\}$	2.885	1.870	1.870	$\{E_{\text{In}}\}$	2.961	1.870	1.870
$\{\pi_{\text{LR}}\}$	2.797	2.922	2.823	$\{E_{\text{LR}}\}$	2.838	2.885	2.782
$\{\pi_{\text{LR,r}}\}$	2.792	2.870	2.771	$\{E_{\text{LR,r}}\}$	2.830	2.938	2.806
$\{\pi_{\text{LR,l}}\}$	2.871	2.879	2.782	$\{E_{\text{LR,l}}\}$	2.832	2.873	2.778
$\{\pi_{\text{RF}}\}$	2.729	2.819	2.749	$\{E_{\text{RF}}\}$	2.784	2.846	2.748
$\{\pi_{\text{In}}, p_{\text{t}}\}$	2.046	1.727	1.649	$\{E_{\text{In}}, p_{\text{t}}\}$	1.961	1.727	1.649
$\{\pi_{\text{LR}}, p_{\text{t}}\}$	2.027	1.877	1.942	$\{E_{\text{LR}}, p_{\text{t}}\}$	1.975	1.862	1.896
$\{\pi_{\text{LR,r}}, p_{\text{t}}\}$	1.969	1.848	1.840	$\{E_{\text{LR,r}}, p_{\text{t}}\}$	1.907	1.810	1.812
$\{\pi_{\text{LR,l}}, p_{\text{t}}\}$	2.038	1.820	1.817	$\{E_{\text{LR,l}}, p_{\text{t}}\}$	1.969	1.862	1.899
$\{\pi_{\text{RF}}, p_{\text{t}}\}$	1.917	1.793	1.797	$\{E_{\text{RF}}, p_{\text{t}}\}$	1.852	1.796	1.820

the RMSE is close to that of the reference method, LR is able to slightly outperform the other two methods in some cases. Also, RF and TB are again very similar in their performances and no clear pattern is visible as to which of the two algorithms is to be preferred.

When looking at the errors of those models using only one of the complexity score variants as their single input feature, it is evident that directly assigning the complexity scores of the video study runs to those of the field study is by far the best method. For both, π_{In} and E_{In}, the performance when only using the complexity score is superior to the 1D AE and $\{p_{\text{t}}\}$ feature sets and it is almost on a par with using the $\{p_{\text{e}}, p_{\text{t}}\}$ feature set which is the best investigated variant that uses two intersection

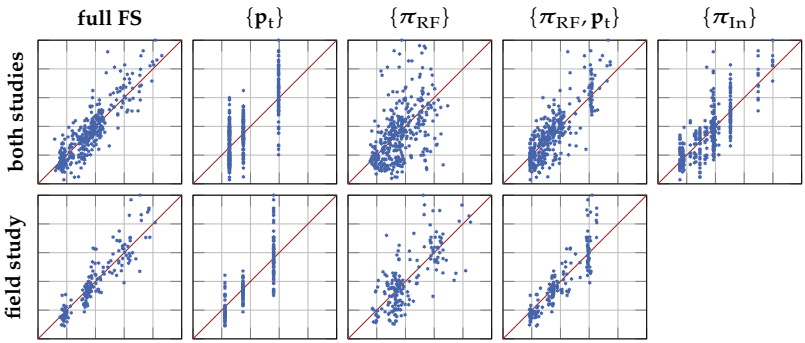


Figure 3.2 Scatter plots using RF for prediction and different feature sets with the commit distance d_c as the label. In the top row the results when using the data set $\mathcal{D}_{iv}^{F}$ are shown, the bottom row shows the results when using the runs of data set $\mathcal{D}_{ov}^{F}$. The abscissa shows the predicted normalized labels $\hat{\mathbf{y}}_n$, the ordinate shows the normalized true labels $\mathbf{y}_n$.

features. The remaining methods for assigning the complexity scores to the field study runs are not suitable to be used as the single feature for a behavior prediction model. Models with these variants result in prediction errors that are almost on the level of the reference method.

In case of the models that use the complexity score variants in combination with the turning direction p_t, the predictions improve for all variants and are in all cases clearly better than the predictions with the reference method. The direct assignment method, in combination with p_t, still is the superior method, i.e. the models that use $\{\pi_{In}, p_t\}$ and $\{E_{In}, p_t\}$ have the lowest RMSE values. However, the differences between these two variants and those using the complexity score models ($\{\pi_{LR}, p_t\}$, $\{E_{LR}, p_t\}$, $\{\pi_{LR,l}, p_t\}$, $\{E_{LR,l}, p_t\}$,$\{\pi_{LR,r}, p_t\}$, $\{E_{LR,r}, p_t\}$, $\{\pi_{RF}, p_t\}$ and $\{E_{RF}, p_t\}$) are a lot smaller than before. Especially the performance of TB is on a par with the 2D and even the 3D AE models in case of the best variants; the performance with RF is lower but still far better than the performance using only the directly assigned complexity scores as features. The ridge and lasso variants of LR are slightly better for complexity score regeneration in many cases. This is true for both the models that use only one of the complexity scores as the feature set and those that additionally

also use p_t. The complexity scores generated with the RF model allow for better prediction than the models based on LR or one of its variants.

The scatter plots in figure 3.2 support these findings. It is clearly visible that the point cloud associated with the full feature set has the smallest variance and follows the ideal line relatively closely. Using only the π_{RF} as the feature set results in a point cloud that is spread out much further and especially larger values of $\mathbf{y}_n$ are only predicted with relatively large errors. This improves visibly when p_t is added to the feature set, i.e. the variance of the point cloud is smaller and also larger values are predicted closer to the real values. Using only p_t results in the characteristic 3 vertical lines, c.f. section 2.5.3.

A similar effect is noticeable in case of using the directly assigned complexity scores π_{In} as the only feature: There appear to be only vertical lines in the scatter plot. This is probably caused by the fact that there is only a limited number of possible complexity score values that the field study runs of data set $\mathcal{D}_{iv}^{F}$ are being assigned to.[6] Because of the limited number of possible input feature values only a limited number of different output values (behavior predictions) are possible. This is further supported by the fact that the RF and TB models that use complexity scores which are assigned by intersection (π_{In} and E_{In}) have almost exactly the same prediction performance. This implies that both algorithms learn the same output assignments for each of the relatively small number of possible input values.

The feature set consisting of one of the complexity score variants and the turning direction p_t shows a better prediction performance than those that use p_t as the only feature in their feature set. This indicates that there is indeed information the prediction algorithms can use within the complexity sores as including them improves the predictions. From the results it further follows that there is no linear dependence between the complexity scores and the driving behavior. As the LR method assumes a linear dependence between the predictor (the complexity score) and the output (the driving behavior), its poor predictive performance is indicative of the missing linear dependence between the two properties.

[6] The runs of $\mathcal{D}_{iv}^{F}$ can only be assigned one of the scores associated with one of the 25 runs of the video study data set $\mathcal{D}^{V}$ whose intersection/p_e combination is also present in the field study data set $\mathcal{D}^{F}$.

Table 3.5 Behavior prediction performance RMSE of different regression algorithms using the data set $\mathcal{D}_{\text{ov}}^{\text{F}}$. The commit distance d_{c} is used as the behavior feature here, thus all results are in m. Using the reference method results in RMSE = 3.164 m. The top rows show prediction results with some of the feature sets of chapter 2 for comparison. The middle rows show the prediction performance when using the BT/Elo complexity score variants as a single feature, the results using these score variants in combination with the turning direction p_{t} are given in the bottom rows.

feature set	LR	RF	TB	feature set	LR	RF	TB
full FS	1.550	1.453	1.489	$\{p_{\text{t}}, p_{\text{e}}\}$	2.004	1.790	1.751
red. FS	1.635	1.467	1.525	$\{V_{\text{f}}, p_{\text{t}}, p_{\text{e}}\}$	1.828	1.612	1.557
$\{p_{\text{t}}\}$	2.047	2.020	2.020	AE 1D	2.654	2.023	2.012
$\{V_{\text{f}}, p_{\text{e}}\}$	2.261	1.878	1.929	AE 2D	2.191	1.660	1.675
$\{V_{\text{f}}, p_{\text{t}}\}$	1.848	1.827	1.931	AE 3D	1.922	1.609	1.697
$\{\pi_{\text{LR}}\}$	2.398	2.519	2.483	$\{E_{\text{LR}}\}$	2.233	2.331	2.315
$\{\pi_{\text{LR,r}}\}$	2.255	2.179	2.146	$\{E_{\text{LR,r}}\}$	2.172	2.125	2.113
$\{\pi_{\text{LR,l}}\}$	2.323	2.304	2.300	$\{E_{\text{LR,l}}\}$	2.220	2.301	2.293
$\{\pi_{\text{RF}}\}$	2.454	2.272	2.244	$\{E_{\text{RF}}\}$	2.400	2.436	2.450
$\{\pi_{\text{LR}}, p_{\text{t}}\}$	1.892	1.775	1.871	$\{E_{\text{LR}}, p_{\text{t}}\}$	1.882	1.823	1.958
$\{\pi_{\text{LR,r}}, p_{\text{t}}\}$	1.860	1.710	1.809	$\{E_{\text{LR,r}}, p_{\text{t}}\}$	1.842	1.708	1.773
$\{\pi_{\text{LR,l}}, p_{\text{t}}\}$	1.876	1.792	1.874	$\{E_{\text{LR,l}}, p_{\text{t}}\}$	1.866	1.827	1.910
$\{\pi_{\text{RF}}, p_{\text{t}}\}$	1.827	1.736	1.823	$\{E_{\text{RF}}, p_{\text{t}}\}$	1.859	1.792	1.912

In summary one can conclude that behavior prediction using the complexity scores is possible, but with some limitations: Of all the models that use one of the complexity scores as their only feature, only those that use the directly assigned scores enable useful predictions. The variants that assign the complexity scores with LR or RF models, which themselves use intersection features as their inputs, are not sufficient for reliable predictions. If the complexity score is combined with the turning position p_{t}, models trained with any variant improve in performance compared to the previous case. Especially the models that rely on the scores calculated from the intersection features improve considerably.

3.4.2 Generalization to Unknown Intersections

The selection of intersection/entry position p_e combinations that were part of the field and video studies overlaps for a large part of the two sets, but there are combinations that were only part of one of the two studies.[7] This circumstance can be used for some interesting evaluations. Those runs of the field study data set $\mathcal{D}^F$ that were recorded at one of the 9 combinations which were part of the field but not of the video study are used in this section to investigate how well the models for reconstructing the intersection complexity scores from the intersection features generalize to unknown intersection/p_e combinations. The runs from these 9 combinations form the data set of runs from the field study that are not in the video study ($\mathcal{D}^F_{ov}$). These combinations can be considered to be unknown to the models for complexity score reconstruction as they were not used for training these models due to them not being part of the video study data set $\mathcal{D}^V$. In this evaluation only T-intersections are considered as the video study only contained those. The resulting data set of 612 runs thus contains all those runs from $\mathcal{D}^F$ that were excluded in the last section. If it is possible to compute useful complexity scores for these runs based on the intersection features alone, it would simplify the application of complexity based behavior prediction substantially. If it is not possible, new intersections will have to be evaluated by additional studies, making this approach unfeasible.

For the experiments with the data set $\mathcal{D}^F_{ov}$ the complexity scores that performed best in the previous section, i.e. the BT and Elo scores that are assigned directly (π_{In} and E_{In}, respectively), cannot be used as there is no run with a fitting intersection/p_t combination within the video study data set $\mathcal{D}^V$ whose complexity score could be assigned. The LR and RF models to reconstruct the complexity scores are however only based on the intersection features and can thus be computed for any run. As the data set considered here is different compared to the data sets from

[7] In the field study data set $\mathcal{D}^F$ there are 34 combinations of intersection and entry position p_e at T-intersections (1818 runs in total). Of these, 25 combinations were also part of the video study (1206 field study runs with these combinations, data set $\mathcal{D}^F_{iv}$), the remaining 9 combinations (612 runs, data set $\mathcal{D}^F_{ov}$) are only present in the field study data set. Of the 29 combinations of the video study 4 are only included in that study. See table 3.1 for an overview over all data sets.

chapter 2 ($\mathcal{D}^{\mathrm{F}}$) and section 3.4.1 ($\mathcal{D}^{\mathrm{F}}_{\mathrm{iv}}$), models are trained again with the feature sets of chapter 2 using the current data set $\mathcal{D}^{\mathrm{F}}_{\mathrm{ov}}$. The results are presented in table 3.5. This table is set up the same as table 3.4, only the models using feature sets containing the directly assigned complexity scores are missing. The results of the models that predict the minimum velocity v_{m} and the velocity drop v_{d} are given in the bottom part of tables B.1 and B.2 in the appendix, respectively.

The results are similar to those of the previous section 3.4.1 in many aspects. The prediction errors of the models using the feature sets of chapter 2 are again similar to those presented in that chapter, despite the smaller data set with only approximately a third of the runs. Comparing the prediction errors to those achieved with the data set $\mathcal{D}^{\mathrm{F}}_{\mathrm{iv}}$ of section 3.4.1, the errors are a little smaller for the current data set $\mathcal{D}^{\mathrm{F}}_{\mathrm{ov}}$. This difference, however, is relatively small. Also, using the BT and Elo scores as the only feature mostly results in poor prediction performances again, i.e. the models trained with the feature sets $\{\pi_{\mathrm{LR}}\}$, $\{\pi_{\mathrm{RF}}\}$, $\{E_{\mathrm{LR}}\}$ and $\{E_{\mathrm{RF}}\}$, respectively, have large errors and are therefore not useful for behavior prediction, even though the errors are notably smaller than those achieved with the data set $\mathcal{D}^{\mathrm{F}}_{\mathrm{iv}}$. In contrast to the results of section 3.4.1, using the ridge and, to a lesser extent, the lasso variants of LR to train the complexity score models results in behavior prediction performances that are substantially better than those of the model that uses the unmodified LR algorithm. The feature set $\{\pi_{\mathrm{LR,r}}\}$ with TB as the prediction algorithm outperforms all other models that use any of the BT complexity score variants as the only feature. In case of the Elo scores the model using the feature set with only the ridge variant score ($\{E_{\mathrm{LR,r}}\}$) produces the best results.

Adding the turning direction p_{t} to these feature sets again improves the predictions considerably in all cases. Similar patterns to those observed above are present again when adding p_{t} to the feature sets, i.e. the models that use the feature set containing the BT score reconstructed by the ridge model, $\{\pi_{\mathrm{LR,r}}, p_{\mathrm{t}}\}$ outperform all other models that use feature sets with BT score variants. Similarly, both the RF and TB behavior prediction models that use the feature set $\{E_{\mathrm{LR,r}}, p_{\mathrm{t}}\}$ perform best of all feature sets that contain an Elo score variant. However, when adding the turning direction p_{t} all complexity scores show similar prediction performances.

Comparing these results to those of the previous section, two aspects stand out: First, the ridge and lasso variants clearly outperform the original LR method only for the current data set $\mathcal{D}_{\text{ov}}^{\text{F}}$. Second, it is obvious that LR is able to perform much better when predicting the driving behavior using the current data set. It is still the worst prediction algorithm over all, but its performance is much closer to that of RF and TB, it is often on a par with these two algorithms and is even able to outperform them in some cases.

Some scatter plots of selected RF prediction models are shown in the second row of figure 3.2. The general results are the same as in section 3.4.1, i.e. the model using the full feature set has the scatter plot where the predictions follow the ideal line most closely, while the model using the $\{p_{\text{t}}\}$ feature set again has the three characteristic vertical lines. The points in the scatter plot of the model based on the $\{\pi_{\text{RF}}\}$ feature set are spread out, while adding p_{t} to this feature set causes the points to be much closer to the ideal. A model using the directly assigned complexity score π_{In} is not possible, as discussed above.

The results of using only these runs that were not part of the video study (data set $\mathcal{D}_{\text{ov}}^{\text{F}}$) perform on a par or even slightly better than those that were part of both studies ($\mathcal{D}_{\text{iv}}^{\text{F}}$). This shows that the models of the complexity scores are able to generalize to further intersections that were not part of the training process (by their intersection/entry position p_{e} combination) of these models. Thus, basing the complexity scores on the intersection features is a valid approach and this concept can probably be extended to further intersections without the need for additional studies if there is sufficient variability in the original video study the complexity score models are based on.

However, these results have to be interpreted carefully, most importantly because both the complexity score models and the behavior prediction models are based on a relatively small number of runs. Especially the behavior prediction in the current section is only based on the data set $\mathcal{D}_{\text{ov}}^{\text{F}}$ with only nine different intersection/p_{e} combinations, recorded at only six different intersections. This is a likely and at least partial explanation why the results in the latter evaluation with the smaller data set performed better, because fewer intersections probably also reduce the variability of the data set, which in turn makes predictions easier.

Therefore, the exact performance measures of these prediction models are not likely to be so relevant and the results should rather be taken for their general message: A meaningful complexity score is not limited to the intersections/entry positions p_e that were part of the video study but the models can also be applied to further intersections.

The improved predictive performance of LR compared to RF and TB as well as the improvements achieved by the lasso and ridge variants of LR when reconstructing the complexity scores stand out. The better performance of the *linear* LR method implies that the behavior prediction task itself is more linear than with the previous data sets, i.e. the complexity scores of the various methods are more strongly correlated with the driving behavior features. A possible explanation for this phenomenon is again the structure of the current data set: The less diverse data set may lead to fewer intersections that have a similar driving behavior on average but are assigned differing complexity scores. Data sets with intersections for which this is the case would thus make the driving behavior prediction problem less linear and so causing worse prediction performances when applying LR.

3.5 Conclusions & Outlook

In this chapter a method of assigning a complexity score to intersections is introduced. This score measures how complex an intersection is perceived by a human driver; it is therefore a subjective feature to describe an intersection. The complexity scores are obtained from an online study, in which participants were asked to state which of two videos of approaches to an intersection showed the more complex scenario to them. In total, 29 videos of approaches to intersections were part of the study.

From these pairwise comparisons complexity scores for all these runs are derived using the BT and Elo methods. The two scores for each video are in the same order for either method. The scores themselves appear to be realistic; in general intersections that are narrower and have poorer visibility are rated as more complex. Further, especially the entry position is an important factor, runs that entered from the right are among the least complex.

Then the correlation of these complexity scores with the intersection features of chapter 2 are calculated. The descriptive dependencies from above are mostly confirmed. Following this analysis, the intersection features are then used to base the complexity scores on them. For that LR and RF models are trained which accept the intersection features as their input and output estimated complexity scores. All this is done using only the 29 runs of the video study data set $\mathcal{D}^{\mathrm{V}}$. By that method, the objective intersection features are used to predict the subjective complexity scores. The results show that this reconstruction is generally possible, yet considerable deviations are also present.

Finally, the different variants to assign complexity scores to the runs through an intersection are employed to assign these scores to the runs of the field study of chapter 2. With the complexity scores as predictors another set of behavior prediction models is trained. These experiments show that using the complexity scores as the only input feature for the regression algorithms is mostly not sufficient for reliable predictions, only the variant where the scores are directly assigned shows good results. When combining a complexity score variant with the entry position as a second feature, the models with either complexity score variant improve substantially and are all suitable for behavior prediction. A final evaluation of the data set consisting of those runs that were not part of the video study (data set $\mathcal{D}^{\mathrm{F}}_{\mathrm{ov}}$) suggests that the concept of assigning complexity scores based on the intersection features is able to be generalized to further intersections.

In summary, the concept of a complexity score for intersections that is based on human perception has shown to be viable: The scores themselves are realistic and behavior predictions with them are possible. However, as shown by the poor performance of the LR algorithm, there is no linear dependence between the complexity scores and the driving behavior, making possible decision-making on these scores hard to explain. As the complexity scores are still based on human perception and are a one-dimensional property, using it should still be superior to the full feature set with regard to explaining the reasons for a decision.

Given the relatively small numbers of intersections and participants in both studies, the results can certainly be seen as a proof of concept, however the exact numeric results of this and the previous chapter should

be interpreted cautiously. Thus further studies of this type with further intersections and more participants would certainly help to improve the reliability of the current results. Given that the reconstruction of the complexity scores using the intersection features is currently only possible with considerable deviations, a more thorough investigation into the typical properties of inner-city intersections appears to be reasonable. These properties should be considered when selecting further intersections for a possible new study and also as part of an extended intersection feature set. Furthermore, the correlation coefficients between the intersection features and the driving behavior within the data of the field study suggest that also the traffic at an intersection might have an influence on the perceived complexity. This aspect has been omitted from the current video study as consistent traffic is difficult to record.

The fact that it is possible to derive a complexity score that is consistent among the participants of the video study suggests that utilizing this concept has potential. The promising behavior prediction results when using only the directly assigned complexity scores suggests that the poor performance of the remaining score variants is not caused by the concept of a complexity score itself but rather by the current state of reconstructing them from the intersection features. The training of these models is likely also made more difficult by the low number of runs through the intersections that are available for that.

4 Decision-Making at Intersections

Driving through an intersection is a challenging endeavor. This is true for human drivers, as illustrated in the previous two chapters, as well as for autonomous vehicles. In this chapter an algorithm is presented with which an AV is able to safely interact at an intersection. The focus will again be on intersections that are not regulated by traffic lights or signs and thus the RBL rule has to be used. Also, mixed traffic is assumed for the entire chapter, i.e. there are AVs as well as vehicles driven by humans on the road. As human-driven vehicles cannot be assumed to communicate their state or turning intention to other vehicles (vehicle to vehicle, V2V) or central control units at an intersection (vehicle to infrastructure, V2I)[1], the worst case scenario is used. This means that no explicit communication is assumed, i.e. no V2V or V2I communication, but also no explicit communication signals by human drivers like hand gestures. The latter is assumed as this would likely be difficult to detect reliably and also the results by Imbsweiler et al. [73] indicate that explicit communication between drivers is of little importance. Therefore, only the basic state of visible vehicles is assumed to be known by the automation. This includes the position, velocity and acceleration of the vehicles.

Two of the challenges that come with driving at non-signalized intersections are already introduced in chapter 1: The problem of determining if a gap before a vehicle with priority arrives is large enough and the possibility for deadlocks to occur. Additionally, one has to take vehicles that do not follow the traffic regulations into consideration. This includes vehicles that stop and wait at an intersection despite having the right of way and vehicles that drive although they have to yield. Both can lead to dangerous situations and a decision-making algorithm has to be able

[1] The general case, i.e. communication of the vehicle with any entity, is commonly referred to as V2x.

to deal with all four challenging scenarios. Determining if a gap is appropriate is relatively simple for an AV as the perception of the position and velocity of vehicles in its surroundings is assumed to be relatively accurate and thus predicting the resulting gap is straight forward. The remaining three scenarios pose more problems and the strategies to deal with them are introduced in this chapter.

In contrast to many of the solutions to similar problems found in literature (c.f. section 4.1) the decision-making algorithm is modeled as a discrete event system (DES, [130]). The advantage of this approach is that the entire process of decision-making is modeled explicitly, i.e. each action taken by the vehicle is triggered by an event, e.g. passing a distance threshold. Especially in contrast to statistical or learning-based methods this increases the ability to explain the reasons for a given behavior as the model is constructed of relatively simple and easy to understand sub-components. This, however, implies that one would have to consider a wide variety of scenarios that could occur at an intersection like scenarios with different numbers of vehicles at different entry roads, leading vehicles and many more. As this variability cannot realistically be covered, a strategy is needed for an algorithm that is as simple as possible while still being generic. In this thesis this is accomplished by only taking those vehicles into consideration that are currently important and ignoring all remaining ones. In the current implementation this results in a maximum of four vehicles that have to be considered at any time. Also, only the pairwise interactions with these vehicles are dealt with, interactions between two cooperation vehicles, i.e. the vehicles the AV interacts with at an intersection, are not modeled. This is true with the exception of the deadlock scenario, in which all vehicles that are involved in it are taken into consideration. These concepts are introduced in detail in the following sections.

This chapter is structured as follows: First, a detailed literature review on decision-making in several driving contexts is presented, followed by a brief introduction into DES and its application in traffic scenarios, see section 4.1. Then, the algorithm is presented in detail in section 4.2. This entails the overall structure of the model, the maps that are used for simulation and the properties of these that are required for decision-making. Then, the algorithm itself is introduced, including the features

and events of the DES model and how the actual behavior is generated. In section 4.3 the simulation environment is presented, including the algorithm for the remaining vehicles at the intersections. Finally, the results of the evaluation in the simulation framework are shown in section 4.4.

The contents of this chapter have been published before. The initial version [189] supported only one vehicle on each road, while the improved version [191, 192] enabled multiple vehicles on the roads, was tested on multiple maps and included further improvements. The latter publications are mostly equivalent to the version presented here. In this thesis, however, the algorithm is presented in more detail and the analysis covers a wider range of aspects.

4.1 Related Work

The problem of decision-making for automatic vehicles has been discussed in literature before. In this section first an overview on different scenarios where decision-making for automatic driving is necessary is given, including examples from literature for each aspect. This is followed by decision-making at intersections, which is the most relevant aspect of decision-making for this work, and thus this is a special focus in this review. Finally, discrete event systems are introduced. These are used here for decision-making and have also been used in traffic-related contexts as well.

4.1.1 Decision-Making for Autonomous Driving

In order to drive with a high degree of automation or ideally fully autonomously, all the different scenarios one might encounter in traffic have to be controlled by the automation. This includes driving on highways or rural roads where velocities are typically relatively high as well as driving within settlements where the velocity is usually lower but intersections and other infrastructure are more abundant, which limit the traffic flow. Further scenarios do exist in the real world and an AV has to be able to drive through all before fully autonomous vehicles are possible.

Almost all of these situations have in common that drivers[2] have to make decisions on their driving behavior. To provide an overview on the diversity of the field of decision-making for autonomous driving, some exemplary works from literature on decision-making models for different aspects of autonomous driving are presented here. This section further gives a first overview on some of the methods commonly used for decision-making. It is noteworthy that many of these methods find application in several of the different scenarios.

A typical challenge for highway driving is entering it. This involves traffic on the throughgoing lanes that typically travels at high velocities and a ramp on which vehicles have to find a suitable gap, while simultaneously accelerating to be able to enter that gap at a velocity that ideally is as close to that of the surrounding traffic as possible. This problem can be solved without the need for V2x by modifying the intelligent driver model (IDM) [84, 174]. The IDM is a car following model and is also used in this thesis; an introduction is given in section 4.2.7. Wang and Chan [157] use reinforcement learning to merge onto highways. It is alternatively also possible to base the behavior decision on predictions of the intention of surrounding vehicles [40, 160]. Marinescu et al. [105] propose a method with which vehicles are required to follow a spacial and temporal slot that determines their trajectory. For that the vehicles are coordinated via V2x while merging onto a highway.

Further aspects of highway driving are also the focus of previous work. The slot-based approach can also be used in the case that the number of lanes on a highway is reduced [104], another aspect of driving where decisions have to be made. Nilsson et al. [119] present a controller for lane change maneuvers, which they demonstrate using a simulation and a real vehicle on a test track. Liu et al. [98] present a model for decision-making to overtake on highways using reinforcement learning.

Another important aspect of driving are intersections. Especially those intersections with large traffic volumes are typically equipped with traffic lights to control the intersection. While implementing the driving rules for traffic lights in an AV is certainly easier than those of non-signalized intersections, reliably recognizing the current state of a traffic light is none

[2] or the automation in case of an AV

the less challenging and can be solved in different ways; e.g. Levinson et al. [90] probabilistically detect the state of traffic lights from video data. More recent approaches also use video sequences [19, 93] but apply neural networks to this problem. An alternative to directly recognizing traffic lights is to have the infrastructure broadcast its current status to the vehicles in its vicinity directly via V2I communication. Bae et al. [20] use the information of the current state of the traffic lights to design a fuel-efficient adaptive cruise control algorithm, while Zhou et al. [175] train a car following model using reinforcement learning for that. Reinforcement learning can alternatively be used to optimize the traffic light controller itself [167]. Pourmehrab et al. [129] propose a traffic light controller that optimizes both the trajectory of AVs passing through an intersection (which they have to follow) and the traffic light phase and timing for conventionally driven vehicles.

The latter approach, i.e. AVs that do not have to follow the traffic lights, is chosen by many works in literature on controlling intersections in the presence of AVs. A review on these approaches is provided in the following section.

4.1.2 Decision-Making at Intersections with V2x Communication

An advantage of AVs is that they will likely be equipped with communication hardware that enables them to share information. This is not limited to AVs, but certain aspects can only be transmitted by vehicles that are driven by a computer; this especially includes the desired turning direction at an intersection. The indicator lights are probably not reliable enough as a human driver might not use them. In this section several approaches from literature are presented that are based on vehicles communicating with each other to share information and to negotiate a solution at an intersection. This typically includes the driving order in which vehicles traverse the intersection and often also the point in time when a given vehicle is supposed to enter the intersection. Most of these solutions for decision-making at intersections that are based on V2x communication require that all vehicles driving there are autonomous.

In the following only those approaches for which it is explicitly stated can also accommodate vehicles driven by human drivers.

There are several proposed decision-making algorithms for intersections where vehicles are assigned a trajectory through an intersection for a certain time. For that the intersection is typically partitioned into smaller segments and vehicles may only use the segments along their trajectory for a specified time if they are scheduled. One of the early approaches of that type is by Dresner and Stone [44, 45], who propose an intersection management system in which a reservation request by a vehicle is granted if the desired segments are not already blocked by the trajectory of another vehicle. This system also allows human-driven vehicles at the intersections by switching to traffic lights as soon as such a vehicle is present at the intersection. The initial version [42, 43] of their algorithm only allowed autonomous vehicles to cross the intersection.

Bento et al. [23] propose a similar approach, theirs supports roundabouts in addition to intersections. Vehicles that are not equipped with V2I capabilities can also be covered by their algorithm; for these vehicles all trajectories with a range of possible speeds is reserved. Azimi et al. [18] propose two algorithms that run on the vehicles only. The vehicles coordinate with surrounding vehicles via V2V communication; all vehicles have to run this algorithm. If two vehicles require the same intersection segment, an external priority decides on the driving order. An extension to their work [16] allows driving through roundabouts and dealing with positioning inaccuracies.

The efficiency of an intersection, in terms of throughput, can be maximized by the ballroom intersection protocol (BRIP) by Azimi et al. [17]. It requires vehicles to follow a predefined arrival pattern with a given constant velocity while driving through the intersection. The pattern is designed to be optimal and thus vehicles are scheduled to occupy the same segments with as little time as possible between them. Aoki and Rajkumar [11, 12] present the configurable synchronous intersection protocol (CSIP) which extends the BRIP by allowing for larger safety gaps between the vehicles so that the extreme timing and positioning requirements of the original algorithm can be loosened. Both BRIP and CSIP do not necessarily require V2x communication, the arrival patterns for a given intersection can be loaded onto the vehicle earlier. A further

extension [9] improves upon the CSIP in that it also allows vehicles driven by humans to pass the intersection. In case such a vehicle is detected, AVs follow the traffic lights at the intersection; otherwise they follow the synchronized pattern.

A similar concept is utilized by two groups: They do not section the entire intersection but only consider the areas where conflicts between two vehicles actually occur, i.e. the area where the trajectories overlap, the so-called conflict zones or conflict sections. One solution [95] requires the vehicles transmit their desired trajectory to a central controller, which gives them instructions on how to traverse the intersection. AVs are required to comply and may not enter the intersection without permission. Vehicles driven by humans are detected by the central controller and follow traffic lights. Lin et al. [96] propose a method to schedule AVs at an intersection using a central controller based on a graph that describes the trajectories of the vehicles along the conflict zones of the intersection. The driving order is adjusted to guarantee the absence of deadlocks which they check by either a graph-based method or with a Petri net (PN).

Using optimization strategies is a further alternative that is commonly used for decision-making at intersections. In the following, approaches are given where the optimization runs on a central unit with which the AVs communicate via V2I communication. Yan et al. [166] propose an algorithm for deciding on the passing order of AVs that minimizes the time it takes for all vehicles at an intersection to pass it. The algorithm is based on dynamic programming. Lee and Park [88] suggest a central intersection control algorithm that optimizes the trajectories of all vehicles so that collisions are avoided. Fayazi and Vahidi [50] assign each approaching AV an arrival time via centralized coordination by solving a mixed-integer linear problem. They also suggest a version for mixed traffic and traffic lights. Qian et al. [134] present an algorithm for autonomous vehicles at unsignalized intersections. Like in solutions presented above, the intersection area is segmented into collision sections. The vehicles are assigned arrival times and the problem is formulated as an absolute value problem.

Running the optimization locally on the vehicles is also possible but the vehicles still have to communicate via V2V communication. The algorithm for coordination of autonomous vehicles at an intersection

by Makarem and Gillet [102] uses model predictive control (MPC) and requires that all vehicles to use the same algorithm and to share their current state. The algorithm introduced by Rodrigues de Campos et al. [139] makes the vehicles sequentially solve an optimal control problem to optimize their trajectory at an intersection while respecting those trajectories that have been planned before. The order in which the optimization is performed is determined heuristically. Kloock et al. [81] prioritize vehicles based on their time to react. The vehicles then plan their velocity along a path using a non-cooperative distributed model predictive control approach in the order determined before.

Many more approaches are conceivable when V2x communication is assumed, some of these are presented in the following. Ahmane et al. [4] use a model implemented as a centralized PN with multipliers for decision-making to derive a controller that runs locally on the vehicles at the intersection and that is based only on a small set of rules. The vehicles at the intersection have to communicate for that. Gregoire et al. [59] propose an intersection management scheme that uses the coordination space approach. The driving order of the vehicles has to be known before. Qian et al. [135] present an algorithm that respects priorities to coordinate vehicles at an intersection. It supports both, AVs which have to be equipped with V2V communication, and human-driven vehicles. Khayatian et al. [79] propose a protocol for centralized intersection management by a central controller that assigns vehicles a time and velocity of arrival. They validate their algorithm in simulation and with model scale vehicles. Zheng et al. [172] present an algorithm that centrally coordinates the driving order at an intersection with single lane streets. The algorithm is based on a state machine for the intersection controller and one for each vehicle. Communication delay is taken into consideration in the system design. An extension [173] is further applicable to intersections that consist of streets with multiple lanes. The protocol running on the vehicles is controlled by a state machine. Chen et al. [30] consider intersections with multiple lanes. Their approach first changes the vehicles into the correct lanes, then the driving order is optimized. The algorithm is run on a central controller. Yan et al. [168] propose a centralized decision-making algorithm that is based on reinforcement learning for mixed traffic at non-signalized intersections. Human-driven

vehicles follow the usual traffic rules while automatic vehicles are controlled by the proposed algorithm. It is designed to optimize the overall traffic flow and may thus force automatic vehicles controlled by it to yield their right of way.

Especially those approaches that require each participant to communicate or to follow a specific algorithm are probably challenging to implement. Even if one of these approaches would be standardized, all older vehicles would be excluded from passing an intersection equipped with that technology. The same would be true if the V2x functionality is unavailable in a vehicle.

4.1.3 Decision-Making at Intersections without V2x Communication

As an alternative to sharing information among the vehicles at an intersection, AVs can detect the relevant data by themselves. AVs have to be equipped with sophisticated sensors anyways, as they need to be able to e.g. localize themselves or detect objects that cannot communicate with them directly or at all like pedestrians, cyclists, parked vehicles, obstacles on the road or construction sites to name just a few. Given the unavailability of communication with these entities anyways, one can view this pessimistic approach about the future state of traffic as more realistic. A further argument is that the communication technology can fail or be attacked, for both cases alternatives are needed. There are many works in literature that use this approach, i.e. they assume the current status of traffic regulations to be in force and no communication via V2x.

A popular method for decision-making are partially observable Markov decision processes (POMDP). This approach models the behavior of other vehicles probabilistically, therefore no communication via V2x or the assumption of traffic consisting of only AVs is necessary. Liu et al. [99] use a POMDP for decision-making at intersections and roundabouts. At unsignalized intersections a combination of a dynamic Bayesian network (DBN) to predict the intentions of other vehicles and a POMDP for decision-making can be used [85]. Hubmann et al. [70] also use a POMDP for decision-making to adapt the driving behavior to the most likely behavior of the other drivers. Bouton et al. [24] use POMDPs for

decision-making at intersections and pedestrian crossings with limited visibility. To make decisions at unsignalized intersections with a POMDP where vehicles are occluded by obstacles, virtual vehicles can be placed at the edge of the obscured space [97]. Shu et al. [144] use a POMDP for decision-making at an intersection while turning left. They define several critical turning points from which a turn can be executed and select the most efficient one. Xia et al. [165] utilize a POMDP approach for prediction and decision-making at non-signalized intersections.

There are further possibilities to solve the decision-making problem without requiring V2x for coordination. One can determine the driving order by analyzing a situation according to the traffic regulations and have an AV drive accordingly [6]. Sezer et al. [143] use a mixed observability Markov decision process (MOMDP) to predict the intention of cooperation partners and base their behavior decision on that. Galceran et al. [52] predict the driving behavior of surrounding vehicles using behavior policies. All possible policies for the ego vehicle are simulated forward and the optimal one is selected. Their work is based on a previous publication by the same group [34]. Further proposed methods for decision-making at intersections include the assignment of a speed profile from a reference list obtained from human drivers [36] or using game theory where a game between the left turning ego vehicle and an oncoming vehicle is considered [137]. Noh [120] presents a framework for decision-making at intersections that consists of prediction, threat detection and decision-making without any explicit communication. Using a Bayesian network the threat level of other vehicles is classified and the driving decision is based on that. Yet another alternative is to extend the intelligent driver model (IDM) for decision-making at intersections [84].

Decisions on the driving behavior can further be based on an abstract description of a situation. A version of that is presented by Vacek et al. [155] who use case-based reasoning. For that purpose they use a database of cases that describe traffic scenarios and different behaviors there. The optimal behavior is selected from that. One can describe traffic scenarios by an ontology, i.e. categorizing the currently relevant segment of the street network and the traffic participants present there including the relations between the members of the two groups. Based on this description one can draw conclusions about the scenario and e.g. de-

duce behavioral decisions. Regele [138] abstracts traffic scenarios to only the lanes and the relations between them. With that abstraction he describes the relevant data of the other vehicles of the scenario and is able to make behavioral decisions with only a small set of rules. Hülsen et al. [71] introduce an ontology to describe traffic situations, which enables them to reason about priority between vehicles at intersections. Armand et al. [13] use an ontology to describe traffic situations that allows to draw conclusions about the traffic ahead. Zhao et al. [170] propose an ontology-based approach for decision-making at intersections without traffic lights or signs and where some of the roads are narrow. The idea behind these works is similar to the algorithm introduced in this chapter. The similarity lies in that behavior decisions are made on an abstract level. The difference to these works is that the description of the scenario is far less detailed here and only those vehicles that are relevant for the decision-making process are considered, see section 4.2.1.

4.1.4 Discrete Event Systems and their Application in Traffic

As outlined above, the underlying assumptions of this work include mixed traffic where V2x communication is not available and no change in regulation compared to the current rules at intersections. This prohibits the use of many of the solutions presented in the previous section as most require at least one of these conditions to not be fulfilled. Of the solutions that do meet the requirements, many use probabilistic models such as a POMDP like e.g. Hubmann et al. [70]. These are able to solve the decision-making problem but their decisions cannot necessarily be easily understood or explained. To overcome this, a solution based on a discrete event system (DES) is presented in this thesis. This type of system requires explicit modeling of every aspect that is considered during the interaction at an intersection, but because of that the inner workings of the algorithm are easily accessible and can be understood and interpreted.

The characteristic feature of a DES is that its current state only changes in discrete points in time when an event occurs [130]. These events can either be already (approximately) discrete, or a simplifying model can be

used to generate discrete events [130]. Events that are already discrete e.g. occur in a traffic light, when the change between states happens (almost) without any transition time [130]. An example of the latter case is the handling of an aircraft at an airport: It e.g. transitions from state *at the gate* to state *on the taxiway* after it has crossed a threshold. In this work only systems with discrete states are considered. Using the airport example this means that the real (continuous) world is split into several discrete states like *at the gate*, *on the taxiway*, *on the runway*. There are several ways to describe and model DES [130]. One can for example utilize Petri nets (PN) [115] or automates.

DES have found application in traffic-related topics. Some examples of using PNs for that are listed in the following: PNs can be used to control traffic lights at intersections [37, 68, 69, 101, 133, 156]. Different aspects are considered, PNs can e.g. be used to model a city environment consisting of intersections with traffic lights and connecting roads [37]. Wang et al. [156] control the traffic lights at intersections with PNs and also model the traffic flow using PNs. Further models consider the interconnection between neighboring intersections [68] or give priority to emergency vehicles [69]. Qi et al. [133] adapt their traffic light controller in case of incidents that would otherwise block neighboring streets. The model used by Luo et al. [101] prevents vehicles from entering blocked streets, by which they reduce traffic jams.

DES are not only useful to control a traffic flow but they can be used to control individual vehicles as well. Mugarza and Mugarza [114] combine a colored PN with the D* Lite algorithm to control automated guided vehicles in an industrial environment. From the works on decision-making at intersections introduced above, some make use of DES: Lin et al. [96] use a PN to check the absence of deadlocks in the driving order their algorithm outputs, while the algorithm of Ahmane et al. [4] is based on the modeling of the processes at the intersection with a PN with multipliers. Aoki and Rajkumar [8, 10] propose a decentralized method of controlling conflict areas outside of intersections, which the authors call dynamic intersections. Their method implements a *cyber traffic light*, a virtual traffic light which ensures that both vehicles from the lane that has priority and those from the lane that has to yield are able to pass the conflict area. For that the front vehicle on the yielding lane requests to

be permitted to pass, which is granted after some time. Then the front vehicle on the priority lane lets some vehicles pass before it requests its right to drive. This is implemented by a state machine all vehicles have to use and follow.

Cellular automates have shown to be a powerful tool for modeling traffic-related aspects as well, especially for simulating traffic in road networks. For that purpose the available infrastructure is divided into cells that can only be occupied by one vehicle at a time. Nagel and Schreckenberg [116] present a cellular automaton model to simulate highway traffic; it can also be applied to simulate the road network of an urban area [141]. Enayatollahi et al. [48] present a cellular automaton model of the terminal air traffic control at an airport. Wei et al. [159] propose a model for traffic light control of a street network where each intersection is a cell in a cellular automaton and the green phases depend on the traffic from neighboring intersections.

4.2 Decision-Making at Intersections Using DES

The problem of decision-making for autonomous driving is a multi-faceted challenge. Intersections, even within a single city, vary widely in their geometry, visibility conditions, the available space and many more aspects, as can be seen from the investigations in chapters 2 and 3. Besides the wide range of possible street geometries, the number of cooperation vehicles (C-Vs) and their behavior are among the most challenging aspects. As the number of C-Vs varies for each situation at an intersection, a strategy has to be found to deal with that. Directly considering all vehicles in the proximity of the intersection would lead to several problems: The model would likely have to be rather large to be able to deal with the varying number of cooperation partners. Also, if one, for example, took all pairwise interactions into consideration, this number would become large even for only a few vehicles (c.f. eq. (3.1)). This, however, is not necessary, as there is only a limited number of vehicles that are currently relevant at any given point in time and which therefore have to be taken into consideration. If there are e.g. multiple vehicles driving one behind the other while approaching an intersection, only the behavior of the first vehicle has to be considered for the decision-

making of an AV. This is evident as all the vehicles following behind another vehicle cannot directly interact with the vehicles on the other streets of an intersection as their behavior options are limited by the vehicle in front of them.

The process of decision-making is described from the perspective of an autonomous vehicle. To differentiate the general case of an autonomous vehicle (AV) from the vehicle that is controlled by the present algorithm, the abbreviation A-V is used for the latter case. The behavior of the A-V depends on several aspects, the most important of which is the behavior of the C-Vs. To safely interact with these vehicles, the model not only needs a strategy for interacting with C-Vs that are driven according to traffic regulations, but also if this is not the case (e.g. a driver yields despite having the right of way). A strategy for not clearly regulated scenarios, e.g. deadlocks, is required as well.

Some constraints have to be imposed here as it is not possible to consider every aspect of the decision-making process: All vehicles follow the center of their respective lane, which means that only the longitudinal acceleration has to be controlled. This approach is a variation of the path-velocity decomposition introduced by Kant and Zucker [78]. However, by omitting the path planning it is not possible to consider objects on the streets like parked vehicles that e.g. block a street at the intersection and thus hinder a vehicle from driving despite being allowed to. Also, only T-intersections are considered here, but the overall concept should be applicable to further scenarios like X-intersections or roundabouts as well.

In the remainder of this section the concept of identifying the relevant vehicles is presented first, followed by the maps and important properties of them, which are needed for decision-making. Also, the method by which visibility is implemented within the simulation is explained. Next, the actual model is introduced, first the features, then the events and the discrete event system that is based on both. Finally, the behavior generation is presented.

4.2.1 Relevant Vehicles for Decision-Making

The decision-making model only considers the relevant vehicles for its decisions and evaluates them separately. The interactions with each of

these vehicles are viewed as atomic scenarios as the vehicles are considered independently from each other and the A-V only drives if it can do so relative to all relevant C-Vs. To model that concept, a traffic light is assigned to each of them and the A-V only drives if all lights are green. This analogy is used for the remainder of this chapter. In total there are four relevant vehicles:

- **The vehicle that has priority (P-V):** In order for a vehicle to have priority over the A-V, it has to approach the intersection on the street to the right relative to the A-V's entry position. If multiple vehicles are approaching the intersection from that direction, the vehicle closest to the intersection is classified as the P-V. If the A-V turns right, no vehicle is marked as P-V as the driving paths cannot intersect and thus interaction is not necessary. An example is showcased in figure 4.1(a).
- **The vehicle that has to yield (Y-V):** Any vehicle approaching the intersection on the street to the left of the A-V's entry position has to yield to the A-V. If this vehicle turns right, it is not assigned the Y-V label as the paths do not overlap. If there is another vehicle behind that vehicle, the second one can be the Y-V even if the first vehicle has not yet passed the intersection. Note that, in contrast to the case of the P-V, not the turning direction of the A-V is relevant for determining the existence of the Y-V but the Y-V's turning direction. Refer to figure 4.1(b) for an example.
- **The vehicle that potentially blocks the intersection (B-V):** The German traffic regulations[3] state that one may not enter an intersection if one would have to wait within it. To comply with these regulations the model also considers the vehicle closest to the intersection that currently leaves the intersection on the same street as the A-V will.
- **The vehicle that leads (L-V):** The vehicle that drives directly in front of the A-V is also relevant as the speed and acceleration of the A-V have to be chosen so that it keeps a safe distance to its L-V.

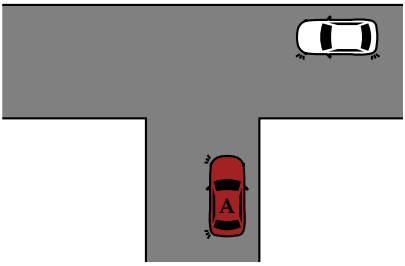
[3] § 11 Abs. 1 StVO

It is also relevant for the decision-making. If the L-V is inside or behind the intersection, it is simultaneously also the B-V if it and the A-V leave the intersection on the same street. An example of this combined case is shown in figure 4.1(c).

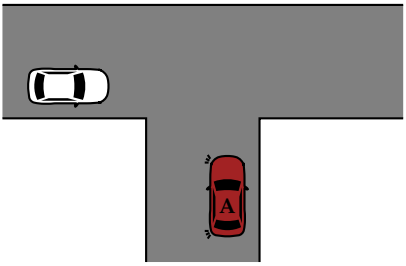
If for any of the four relevant C-V types no vehicle exists, this role is considered to be non-existent. This is the case if additionally a reference point on the street it would drive on is visible, otherwise it is assumed that it might exist. If no L-V is detected, it is always assumed to be non-existent. The reference points are placed on the center lines of the streets. In the case of the P-V and the Y-V the reference points are placed 25 m from the intersection center; the reference point for the B-V is placed at a distance of 15 m. The reference point to determine if a vehicle exists in the latter case is located closer to the intersection as a vehicle further from it would not block the exit of the intersection anyways because there would be enough space behind it. To place the reference points the distances to them is measured by extending the center line of the street on which the point is to be placed as a straight line towards the intersection itself. The distance is then measured along the extended center line of the street from the projection of the intersection center point onto that line.

As no V2x communication is assumed, there is no direct way to communicate a vehicle's intended turning direction to the surrounding vehicles and only observations can be used to deduce it. For that reason, the A-V is presumed to not know the turning direction of a C-V while it is further than 10 m away from the start of the intersection. Instead, the worst case from the A-V's perspective is assumed. Only after the C-Vs are closer to the start of the intersection than this threshold the A-V is assumed to know the real direction. This is implemented in the simulation framework, see section 4.3. The assumption that the turning direction is known at some distance before the intersection is presumably realistic considering previous works from literature [146, 179] that investigate the prediction of the intended turning direction of vehicles at intersections.

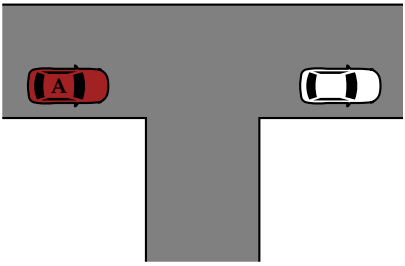
In many situations at a T-intersection there are several relevant C-Vs; commonly a scene is thus a combination of the atomic scenarios from figures 4.1(a) - 4.1(c). An example of a situation with multiple vehicles is showcased in figure 4.1(e). The only situation that is not covered by the atomic scenarios is the deadlock. This situation occurs if there are



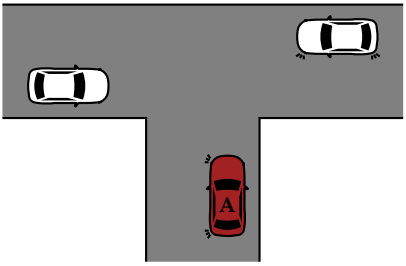
(a) Interaction with a P-V.



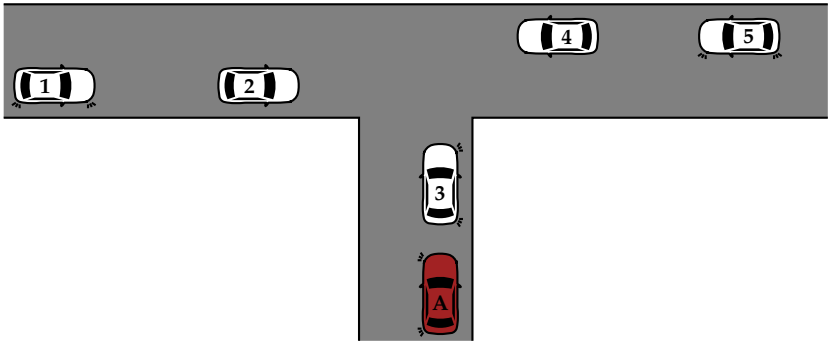
(b) Interaction with a Y-V.



(c) Interaction with a B-V that is also the L-V.



(d) Deadlock situation.



(e) Exemplary scenario: Vehicle 3 is in front of the A-V, which makes it the L-V. Vehicle 2 is the Y-V as it enters from the left and drives straight. As vehicle 1 does not interfere with the A-V's path, it will not be classified as the Y-V even after vehicle 2 passes the intersection. Vehicles 4 enters from the right and is classified as the P-V.

Figure 4.1 Examples of the atomic interaction patterns (figures 4.1(a) - 4.1(c)). The A-V is the red vehicle and the intended turning directions are shown by the indicator lights. The atomic patterns are possible from all three entry positions. Any given situation at a T-intersection is a combination of these patterns except for the deadlock scenario (figure 4.1(d)). An example of such a situation is shown in figure 4.1(e).

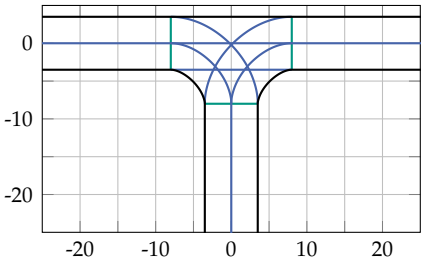


Figure 4.2 Generic map of a T-intersection that is used for simulation. The map is made up of lanelets, the inner borders of those are marked in blue and green. All dimensions are in meters. For simulation purposes the parts of the streets leading into the intersection are extended further out as a straight street, which is only shown in part here.

three vehicles at a T-intersection; one vehicle enters from the left and drives straight, one enters from the bottom direction and drives left and one vehicle enters from the right and also turns left. In this case nobody has the right of way as each vehicle has priority over one vehicle while having to yield to the other, as is shown in figure 4.1(d). A special strategy is required to solve this situation.

4.2.2 Maps

The decision-making algorithm is closely connected to important points on the maps. As in the previous chapters, the maps again consist of a simplified version of the lanelets concept [22, 127]. In this chapter the beginnings and the ends of the lanelets that make up the actual intersection area are especially important; these are marked in green in figure 4.2. The beginning of the intersection is defined as the point where the lanelets within the intersection start, i.e. the point where the lanes for the two directions start to diverge. Similarly, the intersection ends at the point where the lanelets within the intersection end, i.e. the point where the two lanes exiting at a certain street are fully merged. These definitions are applied to all three entry and exit positions at a T-intersection.

The decision-making algorithm is targeted towards the same type of intersections as were viewed in the previous two chapters. Characteristic

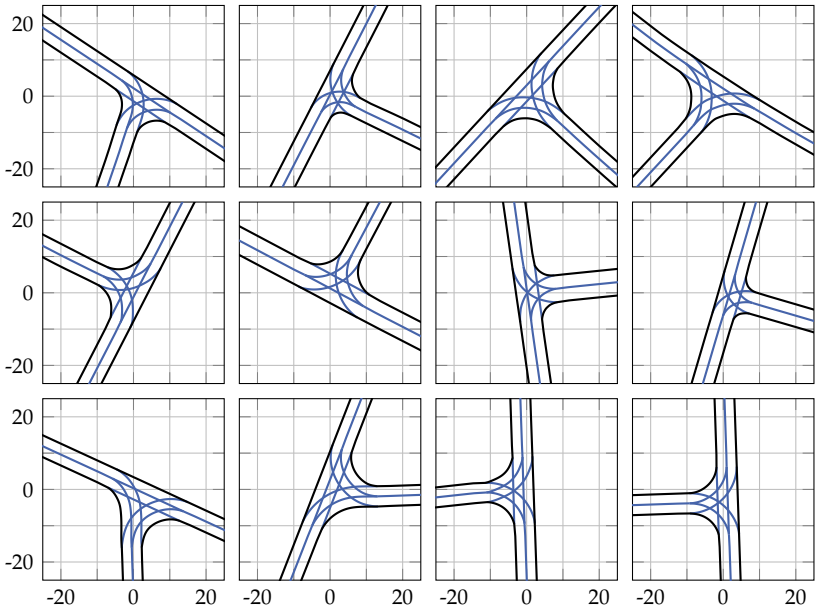
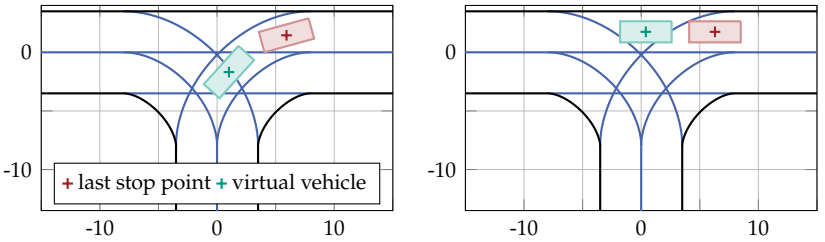


Figure 4.3 Real-world maps from inner city intersections in Karlsruhe that were used for simulation. First row: maps 1 to 4; second row: maps 5 to 8; third row: maps 9 to 12. All dimensions are in meters. The interior borders making up the lanes are marked in blue.

properties of these intersections include that they are usually found in residential inner-city areas; each street leading into the intersection has only one lane in either direction, they have a speed limit of 30 km h^{-1} and there are no traffic signs or lights, thus the RBL rule applies. For the initial evaluation (see section 4.4.1) the generic map in figure 4.2 is used. It has the aforementioned properties and does not show a real-world intersection. To validate that the algorithm is also applicable to real intersections, 12 of the intersections that are part of the study of chapter 2 are used. These include 11 of the 14 T-intersections as well as the intersection where the deadlock occurred. The intersections are renumbered from 1 to 12 and are shown in figure 4.3.

The model for decision-making depends on some important points on the maps. These are introduced using the generic map (figure 4.2). The



(a) LSP when entering from the right and turning left. The vehicle stops just before it would block another lane.

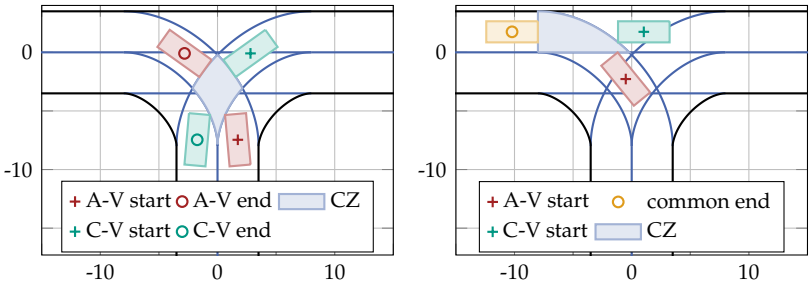
(b) LSP when entering from the right and driving straight. The vehicle stops just before it would leave either of the two paths from the right entry position.

Figure 4.4 Two examples of the calculation of LSPs at a T-intersection. These are the last points a vehicle can stop at without blocking further lanes. If a vehicle is to stop there, a virtual L-V is added to enforce a stop. If a safety margin is desired, the virtual vehicle can be moved closer to the start of the intersection along the drive path. Only the coordinates of the centers are relevant, the rectangular bounding boxes of the vehicles are added for illustration purposes only. All dimensions are in meters. The calculation is done for all six entry position/driving direction combinations.

definition of these points, however, is not limited to this map; they are applicable to the map of any T-intersection.

The first type of relevant points are the *last stop points* (LSP). These describe the position on the map at which a vehicle with a given driving path[4] has to stop if it is supposed to not block any other lane but its own. On any map there exist only six LSPs, one for each of the six possible driving paths at a T-intersection. For three of the driving paths the calculation is adapted, so that there is not more than one vehicle with the same relevant C-V label (e.g. P-V). These three paths are: entering from the left and turning right, entering from below and turning right and entering from the right and driving straight. The last of these three cases is showcased in figure 4.4(b). Additionally, the case when again entering from the right but turning left is presented (figure 4.4(a)). The LSPs in these two cases are shown in red. The green vehicles are virtual vehicles that are placed on the map if the A-V is supposed to stop at the LSP,

[4] As the vehicles follow the road center, a driving path is fully defined by specifying an entry position and a turning direction.



(a) The A-V enters the intersection from the bottom and turns left. A C-V enters from the right and also turns left.

(b) The A-V enters the intersection from the bottom and turns left. A C-V enters from the right and drives straight.

Figure 4.5 Collision zones (CZ) at a T-intersection. If the two intersecting lanes leave the intersection on the same street, the CZ ends where the intersection itself ends. The distances to the start and end points of the CZ are zero where the vehicle in question is directly at the beginning the zone. These positions are marked for both vehicles. In this work only the coordinates of the centers are relevant because the vehicles follow their lane center. The rectangular bounding boxes of the vehicles are added for illustration purposes only. All dimensions are in meters.

see section 4.2.7 for more details. To force the A-V to stop further from the LSP, the position of the virtual vehicle could be moved further back and thus closer to the beginning of the intersection. The necessity for adapting the calculation of the LSPs as described above can be seen in figure 4.4(b). If there are two vehicles waiting that both entered from the right but one drives straight and one turns left, these vehicles could be standing besides each other at a wider intersection and thus be allowed to enter the intersection simultaneously. In that case there might be two C-Vs of the same type, which is not supported by the decision-making algorithm. This is prevented by placing both LSPs from the same direction at that place where the first of the two paths would intersect with another one. In figure 4.4(b) this is visible as the the LSP is placed further back than it needs to be following the original definition. In that case it would be positioned a little behind where the virtual vehicle is placed.

The second class of important points on the maps are the beginnings and ends of the *collision zones*. These zones are caused by the fact that many pairs of driving paths overlap. For that reason the entire width of the lanelets in question is considered and the collision zone is the area

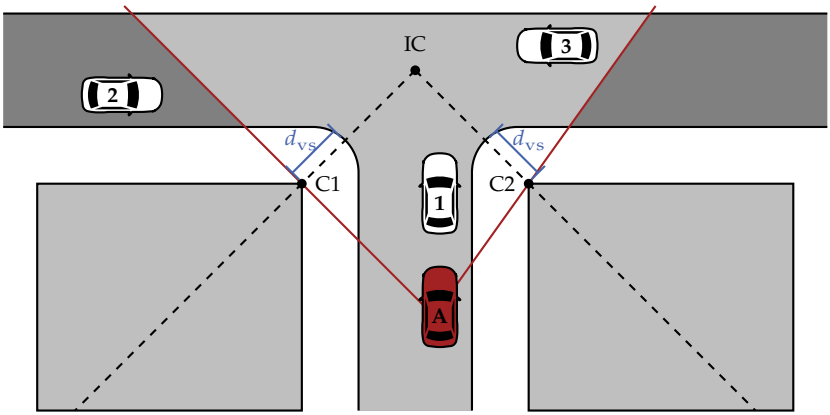


Figure 4.6 Schematic example of the visibility calculations. Obstacles are defined by the corner (C1 and C2) that is closest to the intersection center (IC). The corners are placed along the bisecting line between their two neighboring streets at a distance of d_{vs} from the curb. Originating from the A-V, one or two sight lines are used to determine the visible area (light gray) on the road. Only the vehicles on that area are considered as visible. In this example, the A-V approaches from the bottom direction. Vehicles 1 and 3 are visible, vehicle 2 is obstructed by the left object.

where the two lanelets actually overlap, see figure 4.5. As the collision zones are the reason for the need for cooperation, it is important for the decision-making to know where the vehicles are relative to this zone. As the vehicles follow the center of their lanes, the positions where vehicles on both paths enter and exit the collision zone are always the same for a given pair of driving paths. The distances to the beginning and the end of the collision zone (see section 4.2.4) can thus be calculated as the distance to these points. In figure 4.5 the calculation for two vehicles is showcased including the positions where the two vehicles enter and leave the collision zone.

4.2.3 Visibility

The focus of this work is on intersections in inner city traffic, thus occlusions are a common occurrence. These are caused by buildings, vegetation, parked vehicles, signs and many more objects in the real world. In

contrast to the calculation of the visibility features in chapter 2, here the limited visibility is considered by defining a point that represents the most exposed corner of all the obstacles. All calculations regarding the visibility of other vehicles and road segments are only based on these points. They are placed between the street to the left and the one in the bottom direction, as well as between the bottom street and the one to the right. The points are placed on the bisecting line between the direction vectors of the two streets they lie between. Both points are at a distance of d_{vs} from the curb of the intersection. For the remainder of this chapter d_{vs} is referred to as the *visibility distance*, again in contrast to the previous two chapters. The visibility calculation is showcased in figure 4.6 using a schematic representation of an intersection map. The two obstacles at the left and right bottom are not needed for the calculation and are only shown for better visualization. The advantages of this approach are that the visibility only depends on a single parameter; it adapts easily to any intersection and it can be calculated with little effort. However, one loses aspects of the surroundings of a real intersection, which can make the visibility less clear. For example visibility might be lost only temporarily due to a single obstacle like a tree trunk. As only T-intersections are considered in this work, there is no need to place a third corner point between the left and right streets, as these streets are all at an angle of approx. 180° to each other, c.f. figure 4.3. If one also considered X-intersections, two additional corners would have to be added, their placement could be done analogously.

4.2.4 Features

The DES for decision-making is based on several features. These describe aspects of the driving behavior of the A-V itself and its C-V that are relevant for the decision-making process. In the remainder of this chapter the vehicle a feature refers to is indicated by the superscript associated with each feature: $(\cdot)^x, x \in \{\mathrm{a}, \mathrm{p}, \mathrm{y}, \mathrm{b}, \mathrm{l}\}$. These are the A-V, P-V, Y-V, B-V and L-V, respectively. Additionally, all distances are measured not as Euclidean distances but along the drive path of the respective vehicles. As the vehicles always follow the center of their lane, the distances are also measured along the lane center the vehicle in question drives along.

All vehicles are modeled as rectangles with a constant vehicle length of $l_\mathrm{v} = 4.4\,\mathrm{m}$ and width $w_\mathrm{v} = 1.8\,\mathrm{m}$.

The first feature is the **distance to scenario** $d_\mathrm{s}^x(t)$. It describes the distance to and from the intersection and is positive during the approach, zero inside and negative after the intersection. The beginning of the intersection is defined by the end of the lanelet of the street leading into the intersection, and the end of the intersection coincides with the start of the lanelet leading away from the intersection. All features are calculated for the current time t, this dependence is omitted for better readability in the following.

At intersections many driving paths overlap and the corresponding lanes cause collision zones (c.f. section 4.2.2 and figure 4.5). As only a single vehicle can be inside a collision zone at any time, these zones are the most relevant place for decision-making and thus features describing the actions there are important. The first of these is the **distance to collision zone**. This feature is relative to two vehicles, the A-V and either its P-V or its Y-V. The distances $d^x_{\mathrm{c},x_\mathrm{c},\mathrm{b}}$ and $d^x_{\mathrm{c},x_\mathrm{c},\mathrm{e}}$ describe the distance of vehicle $x, x \in \{\mathrm{a},\mathrm{p},\mathrm{y}\}$ to the beginning and end of the collision zone of the A-V to its collision zone with its C-V $x_\mathrm{c}, x_\mathrm{c} \in \{\mathrm{p},\mathrm{y}\}$. With that definition e.g. $d^\mathrm{p}_{\mathrm{c},\mathrm{p},\mathrm{b}}$ is the distance of the P-V to the beginning of the collision zone between it and the A-V.

The second feature related to the collision zones is the **time to collision zone**. It describes the time it takes a vehicle to reach or leave a collision zone assuming the current velocity is kept constant:

$$t^x_{\mathrm{c},x_{\mathrm{c}'}} = \frac{d^x_{\mathrm{c},x_{\mathrm{c}'}}}{v^x}\,. \tag{4.1}$$

A further feature is the **distance to stop** d^x_b. This is the required distance to bring a vehicle to a complete stop assuming an arbitrary velocity v^x_a and acceleration a^x_a:

$$d^x_\mathrm{b}\left(v^x_\mathrm{a}, a^x_\mathrm{a}\right) = \begin{cases} -\frac{\left(v^x_\mathrm{a}\right)^2}{2a^x_\mathrm{a}}\,, & a^x_\mathrm{a} < 0\,\mathrm{m\,s^{-2}} \\ 0\,\mathrm{m}\,, & a^x_\mathrm{a} = 0\,\mathrm{m\,s^{-2}} \wedge v^x_\mathrm{a} = 0\,\mathrm{m\,s^{-1}} \\ \infty\,, & \text{otherwise} \end{cases}. \tag{4.2}$$

This feature, together with the **distance to last stop point** d^x_l is required to judge if a stop before the intersection is still possible.

The final feature that is needed for the decision-making model is the **free distance behind the B-V** :

$$d_{\mathrm{f}}^{\mathrm{b}} = d_{\mathrm{i}}^{\mathrm{b}} - \frac{1}{2} l_{\mathrm{v}} + d_{\mathrm{b}}^{\mathrm{b}} \left(v^{\mathrm{b}}, a_{\mathrm{e}} \right) . \tag{4.3}$$

$d_{\mathrm{i}}^{\mathrm{b}}$ is the distance from the end of the intersection to the B-V. As all distances are relative to the center of the vehicles, half of the vehicle length l_{v} has to be subtracted. Finally, the distance the B-V would travel in case of braking with the emergency deceleration $a_{\mathrm{e}} = -7.5\,\mathrm{m\,s^{-2}}$ is added. With this feature one can assess if the B-V is far enough from the intersection so that the A-V can drive through it without risking to be stuck behind the B-V within the intersection.

4.2.5 Events

As the name suggests, a DES is based on events to determine the current state of the system. Not all events are relevant for the entire approach to and drive through the intersection. For that reason, the way through the intersection is split into six zones. This is helpful to adapt the driving behavior of the A-V depending on the distance to the intersection. The zone the A-V is currently in is determined by the distance to scenario $d_{\mathrm{s}}^{\mathrm{a}}$. Each zone has states associated with it, zones 2, 3, 4 and 5 each have at least an offensive and a defensive state assigned to them. The model with its states is introduced in detail in section 4.2.6. In the following, first the general behavior within each zone is presented, then the events to facilitate that behavior are introduced.

The model is in the first zone if $d_{\mathrm{s}}^{\mathrm{a}} > 40\,\mathrm{m}$ holds. In it the A-V is still far from the intersection and it can drive freely [5], and is not controlled by the decision-making algorithm. The second zone includes the distance $40\,\mathrm{m} \geq d_{\mathrm{s}}^{\mathrm{a}} > 25\,\mathrm{m}$ and zone 3 is within the range of $25\,\mathrm{m} \geq d_{\mathrm{s}}^{\mathrm{a}} > 10\,\mathrm{m}$. In these zones the A-V adapts its velocity for the first time. However, to avoid changing the behavior too often the prediction for the P-V is only run at the beginning of these zones. The strategy for both zones is the same: The A-V drives defensively if there is a P-V that is not clearly arriving after the A-V or if it is not yet certain if a P-V exists in case the

[5] *Driving freely* means that the A-V only takes the speed limit and its L-V into consideration.

street in question is not yet fully visible. Otherwise the A-V chooses a more offensive behavior. The other relevant vehicles are not considered at that time other than avoiding a collision with the L-V. This, however, is not done by events but by controlling the acceleration, see section 4.2.7. The reason for only considering the P-V at that stage is to signalize to all C-Vs, in case they are already visible, that the algorithm is aware of the priority rules. That is, it can drive before the Y-V but has to yield to the P-V. This strategy, of course, works best if there is only one of the two types of C-Vs present.

While the A-V is within $10\,\mathrm{m} \geq d_\mathrm{s}^\mathrm{a} > 1\,\mathrm{m}$ from the intersection, it is in zone 4. In this zone it is close to the intersection and the decision to drive or stop has to be made. For that reason all four relevant vehicles are now taken into consideration and the A-V drives offensively only if all relevant C-Vs allow it. This is necessary, as the A-V e.g. also has to stop for the Y-V if it does not wait. In this zone the remaining distances are short and yielding in this case, even if the A-V has priority, is necessary to avoid a collision. Changing between offensive and defensive behavior is now possible in every time step as the A-V has to be able to react quickly to a changing environment.

In zone 5 ($1\,\mathrm{m} \geq d_\mathrm{s}^\mathrm{a} \geq 0\,\mathrm{m}$) the A-V is within the intersection. It either drives through the intersection or waits for other vehicles to pass before it. The resolution of deadlocks also happens in that zone. The final sixth zone is the street beyond the intersection: $0\,\mathrm{m} > d_\mathrm{s}^\mathrm{a}$. In it the A-V has passed the intersection and is no longer controlled by the decision-making algorithm.

These behavior patterns are enabled by the events of the DES. All events are listed in table 4.2; for better readability most events are a combination of so-called base events, see table 4.1 for an overview and their formal definition. The decision-making uses a traffic light analogy for the relevant C-Vs, as introduced in section 4.2.1. There is an event assigned to each of the four relevant C-Vs; the P-V is the only one of these to have two events, one for zones 2 and 3 and the other for zones 4 and 5. Further events are used for resolving deadlocks and to check important aspects of the interaction at a T-intersection. All are introduced in detail in the following.

Table 4.1 Base events (BE) for the DES for decision-making. A base event is true if its condition is fulfilled.

BE	condition	description
e_{b1}	the P-V cannot exist or the reference point is visible and no P-V is detected	P-V: non-existence
e_{b2}	$t^{a}_{c,p,e} + \Delta t^{p} < t^{p}_{c,p,b} \wedge d^{a}_{c,p,e} + \Delta d^{p} < d^{p}_{c,p,b}$	P-V not obstructed
e_{b3}	$v^{p} < v_{s} \wedge a^{p} \leq 0\,\frac{m}{s^2} \wedge d^{p}_{s} < d_{n} \wedge d^{p}_{c,p,b} > 0\,m$	P-V stopped
e_{b4}	$t^{p}_{w} > t_{y} \wedge t^{a}_{w} > t_{y}$	P-V yields
e_{b5}	$d^{y}_{c,y,b} < 0\,m \wedge d^{y}_{c,y,e} > 0\,m$	Y-V in CZ
e_{b6}	$t^{a}_{c,y,e} < t^{y}_{c,y,b}$	Y-V: no collision
e_{b7}	$d^{a}_{l} > d^{a}_{b}\,(v_{a}, a_{c})$	comf. stop possib.
e_{b8}	$d^{a}_{l} > d^{a}_{b}\,(v_{a}, a_{h}) + d_{o} \wedge v^{y} < v_{sl} \wedge a^{y} < 0\,\frac{m}{s^2} \wedge d^{y}_{c,y,b} > d^{y}_{b}\,(v^{p}, a^{p})$	Y-V brakes
e_{b9}	$v^{y} < v_{s} \wedge a^{y} \leq 0\,\frac{m}{s^2} \wedge d^{y}_{s} < d_{n} \wedge d^{y}_{c,y,b} > 0\,m$	Y-V stopped
e_{b10}	ref. point is visible and no B-V detected	B-V: non-existence
e_{b11}	$d^{b}_{f} > l_{v} + d_{min}$	B-V: enough space
e_{b12}	no L-V detected	L-V: non-existence
e_{b13}	$d^{l}_{s} < 0\,m$	L-V passed int.
e_{b14}	$d^{a}_{l} > d^{a}_{b}\,(v^{a}, a_{e})$	em. stop possible
e_{b15}	A-V, P-V, Y-V: turning directions intersect	DL possible
e_{b16}	$v^{a} < v_{s} \wedge a^{a} \leq 0\,\frac{m}{s^2} \wedge d^{a}_{s} < d_{n}$	A-V stopped

4.2.5.1 Events of the Relevant C-Vs

The P-V has the right of way over the A-V at an intersection, thus the A-V has to let it drive first if it exists and the A-V cannot pass the intersection safely and without obstructing the P-V. Event $e_{1,p,I}$ is true if the P-V gives green light in zones 2 and 3. This is the case if one of the two following base events occur: Base event e_{b1} is true if the non-existence of the P-V is certain. This is the case if the vehicle either cannot exist due to the turning direction of the A-V (c.f. section 4.2.1) or if the reference point of the P-V's street is visible and no vehicle can be detected. If the A-V is currently predicted to leave the common collision zone in both distance

Table 4.2 Events of the DES for decision-making. Most events are a combination of base events.

definition	description
$e_{1,\mathrm{p,I}} = e_{\mathrm{b1}} \vee e_{\mathrm{b2}}$	P-V: green light in zones 2/3
$e_{1,\mathrm{p,II}} = e_{\mathrm{b1}} \vee e_{\mathrm{b2}} \vee (e_{\mathrm{b3}} \wedge e_{\mathrm{b4}})$	P-V: green light in zones 4/5
$e_{1,\mathrm{y}} = \neg e_{\mathrm{b5}} \wedge (e_{\mathrm{b6}} \vee e_{\mathrm{b7}} \vee e_{\mathrm{b8}} \vee e_{\mathrm{b9}})$	Y-V: green light
$e_{1,\mathrm{b}} = e_{\mathrm{b10}} \vee e_{\mathrm{b11}}$	B-V: green light
$e_{1,\mathrm{l}} = e_{\mathrm{b12}} \vee e_{\mathrm{b13}}$	L-V: green light
e_2	next zone entered
$e_3 = e_{\mathrm{b14}}$	emergency stop possible
$e_4 = e_{\mathrm{b15}}$	deadlock possible
$e_5 = e_{\mathrm{b3}} \wedge e_{\mathrm{b9}} \wedge e_{\mathrm{b16}}$	deadlock detected
$e_6 = e_{\mathrm{b3}} \wedge e_{\mathrm{b9}}$	deadlock of C-Vs detected
$e_{\mathrm{g}} = e_{1,\mathrm{p,II}} \wedge e_{1,\mathrm{y}} \wedge e_{1,\mathrm{b}} \wedge e_{1,\mathrm{l}}$	green light by relevant C-Vs
$e_{\mathrm{dl}} = e_4 \wedge e_5 \wedge e_{1,\mathrm{l}} \wedge e_{1,\mathrm{b}}$	deadlock can be resolved

and time before the P-V enters it, including a lead distance of $\Delta d^{\mathrm{p}} = 10\,\mathrm{m}$ and a lead time of $\Delta t^{\mathrm{p}} = 2.5\,\mathrm{s}$, the condition for e_{b2} is met:

$$t^{\mathrm{a}}_{\mathrm{c,p,e}} + \Delta t^{\mathrm{p}} < t^{\mathrm{p}}_{\mathrm{c,p,b}} \wedge d^{\mathrm{a}}_{\mathrm{c,p,e}} + \Delta d^{\mathrm{p}} < d^{\mathrm{p}}_{\mathrm{c,p,b}}. \tag{4.4}$$

The P-V gives green light in zones 4 and 5 ($e_{1,\mathrm{p,II}}$) if again either base event e_{b1} or e_{b2} occurs. It is alternatively also green if both vehicles have been standing close before the intersection long enough, i.e. the P-V yields despite not having to do so. This is the case if the P-V is stationary close before the intersection (e_{b3}) and both vehicles have been waiting for at least $t_{\mathrm{y}} = 2\,\mathrm{s}$ (e_{b4}). The A-V is stationary shortly to the intersection if its velocity is below the threshold for stopping $v_{\mathrm{s}} = 0.15\,\mathrm{m\,s^{-1}}$, its acceleration is not positive, it is closer than $d_{\mathrm{n}} = 12\,\mathrm{m}$ from the start to the intersection but it has not yet entered the collision zone with the A-V, i.e. $d^{\mathrm{p}}_{\mathrm{c,p,b}}$ is still positive:

$$v^{\mathrm{p}} < v_{\mathrm{s}} \wedge a^{\mathrm{p}} \leq 0\,\mathrm{m\,s^{-2}} \wedge d^{\mathrm{p}}_{\mathrm{s}} < d_{\mathrm{n}} \ \wedge d^{\mathrm{p}}_{\mathrm{c,p,b}} > 0\,\mathrm{m}. \tag{4.5}$$

Base event e_{b4} occurs if the timer $t^{\mathrm{p}}_{\mathrm{w}}$ that measures the duration of both the A-V and its P-V standing at the intersection exceeds the waiting

threshold: $t_{\mathrm{w}}^{\mathrm{p}} > t_{\mathrm{y}}$. The timer is only incremented if the algorithm is in the defensive states of zones 4 or 5 (see section 4.2.6), the P-V ($e_{\mathrm{b}3}$) and the A-V ($e_{\mathrm{b}16}$) are standing at the intersection and a deadlock is not possible ($\neg e_4$). Base event $e_{\mathrm{b}16}$ and event e_4 are introduced below. The timer is not incremented in case of a deadlock as the P-V is probably stopped because it has to yield itself and not because it does not want to drive first.

As the A-V has priority over its Y-V, the traffic light should always be green unless the Y-V does not stop and a collision is thus predicted. Therefore, the A-V can drive before the Y-V ($e_{1,\mathrm{y}}$) – if no other vehicles would be considered – in case the Y-V is not inside its collision zone with the A-V ($\neg e_{\mathrm{b}5}$). If the Y-V is inside the collision zone, the distance to the start of the collision zone is negative while the distance to the end is still positive:

$$d_{\mathrm{c,y,b}}^{\mathrm{y}} < 0\,\mathrm{m} \wedge d_{\mathrm{c,y,e}}^{\mathrm{y}} > 0\,\mathrm{m}. \tag{4.6}$$

Event $e_{\mathrm{b}5}$ on its own is not sufficient to ensure safety, additionally to $\neg e_{\mathrm{b}5}$ at least one of these base events also has to be true for a green light by the Y-V ($e_{1,\mathrm{y}}$) to occur:

- $e_{\mathrm{b}6}$: The A-V is predicted to leave the common collision zone before the Y-V enters it, i.e. for the times to their common collision zone

  $$t_{\mathrm{c,y,e}}^{\mathrm{a}} < t_{\mathrm{c,y,b}}^{\mathrm{y}} \tag{4.7}$$

  must hold.

- $e_{\mathrm{b}7}$: Stopping regularly before the intersection is still possible. For that the distance to the last stop point of the A-V $d_{\mathrm{l}}^{\mathrm{a}}$ has to be larger than the distance required to stop when driving at the velocity v_{a} and braking with the comfort deceleration $a_{\mathrm{c}} = -2.5\,\mathrm{m\,s^{-2}}$, that is

  $$d_{\mathrm{l}}^{\mathrm{a}} > d_{\mathrm{b}}^{\mathrm{a}}\left(v_{\mathrm{a}}, a_{\mathrm{c}}\right). \tag{4.8}$$

  v_{a} is the assumed velocity the A-V will drive at within the intersection; $v_{\mathrm{a}} = 6.5\,\mathrm{m\,s^{-1}}$ if it drives straight and $v_{\mathrm{a}} = 4.0\,\mathrm{m\,s^{-1}}$ in case the A-V turns at the intersection.

- $e_{\mathrm{b}8}$: The Y-V brakes and the A-V could still stop. This event is true if stopping with a hard deceleration of $a_{\mathrm{h}} = -4.5\,\mathrm{m\,s^{-2}}$ is still

possible, including a distance offset of $d_\mathrm{o} = 0.2\,\mathrm{m}$. Additionally, the Y-V has to be slower than $v_\mathrm{sl} = 2\,\mathrm{m\,s^{-1}}$ and it has to be braking so that it can stop before the beginning of the common collision zone:

$$d_\mathrm{l}^\mathrm{a} > d_\mathrm{b}^\mathrm{a}\left(v_\mathrm{a}, a_\mathrm{h}\right) + d_\mathrm{o} \wedge v^\mathrm{y} < v_\mathrm{sl} \wedge a^\mathrm{y} < 0\,\frac{\mathrm{m}}{\mathrm{s}^2} \wedge d_\mathrm{c,y,b}^\mathrm{y} > d_\mathrm{b}^\mathrm{y}\left(v^\mathrm{p}, a^\mathrm{p}\right). \tag{4.9}$$

- e_b9: The Y-V is stopped close before the start of the intersection. The conditions for this base event are the same for the Y-V as they are in e_b3 for the P-V.

The selection of features and base events and their parameterization are especially challenging for the event $e_\mathrm{1,y}$ as there is a conflict of interests that is difficult to resolve: The A-V should drive offensively relative to the Y-V as it has priority over it. Driving (too) defensively might signal to the Y-V that the algorithm is uncertain or even that it waives its right of way. But the A-V should not be driving so offensively either that it risks a collision. A collision is predicted (base event e_b6) relatively easily as the A-V has to pass the intersection before the Y-V even enters it; the required headway is thus relatively large. If e_b6 were the only base event that is considered the algorithm would behave rather defensively as it would often yield to the Y-V. Base events e_b7 and e_b8 counteract that tendency by being less conservative and allowing offensive driving while stopping is still possible in the worst case.

The role of both the L-V and the B-V is different than that of the other two relevant C-Vs. They do not have priority over the A-V but they can hinder the A-V from driving through the intersection. The A-V can only drive if the L-V has already passed the intersection. If it has the same turning direction as the A-V it then becomes the B-V after it passed the intersection. If the turning directions differ, the C-V in question will no longer be labeled as the L-V as soon as it has left the lane of the A-V. The event $e_\mathrm{1,l}$ that is associated with the L-V occurs if there is no L-V(e_b12) or if the L-V has passed the intersection (e_b13). The L-V has passed the intersection if its distance to the scenario is negative:

$$d_\mathrm{s}^\mathrm{l} < 0\,\mathrm{m}. \tag{4.10}$$

The B-V leaves the intersection on the same road as the A-V, it can thus obstruct it from exiting the intersection. As the A-V is not supposed to stand within the intersection, it only enters it if it is not blocked by the B-V(event $e_{1,\mathrm{b}}$). This is the case if either the B-V does not exist (e_{b10}) or if the distance behind the B-V is large enough (e_{b11}). The B-V does not exist if the reference point on its street is visible while no vehicle on its lane is detected. The distance behind the B-V is large enough if

$$d_{\mathrm{f}}^{\mathrm{b}} > l_{\mathrm{v}} + d_{\mathrm{min}} \tag{4.11}$$

holds, i.e. there is enough distance between the B-V and the end of the intersection for the length of the A-V (l_{v}) and the minimum headway $d_{\mathrm{min}} = 1.5\,\mathrm{m}$ between following vehicles.

4.2.5.2 Events During the Approach

As the behavior depends on the zone the A-V is currently in, zone changes are an important event. In each time step in which the current zone is not the same as in the previous time step, event e_2 is true. In zones 4 and 5 the A-V is very close to the intersection and it is therefore possible that the A-V is no longer able to stop before the last stop point. If it braked regardlessly, it would stop within the intersection, thus blocking it and in the worst case causing an accident. The algorithm is only supposed to switch to defensive behavior if stopping before the last stop point with an emergency deceleration of $a_{\mathrm{e}} = -7.5\,\mathrm{m\,s^{-2}}$ is still possible. Base event e_{b14} and event e_3 check for that and are only true if the distance to the last stop point $d_{\mathrm{l}}^{\mathrm{a}}$ is larger than the distance required to stop when driving with the current velocity v^{a} and braking with a_{e}:

$$d_{\mathrm{l}}^{\mathrm{a}} > d_{\mathrm{b}}^{\mathrm{a}}\left(v^{\mathrm{a}}, a_{\mathrm{e}}\right). \tag{4.12}$$

This behavior is considered safe for two reasons: First, the parameterization is rather conservative, i.e. the A-V is probably only driving offensively if there is some additional safety margin. If one of the lights then turn red, it does not mean a collision is immediate but rather that the safety margin has become too small. Secondly, one can argue that it is safer to quickly pass the intersection in that situation and clear it for the C-V instead of blocking it by slowing down or even stopping, forcing

the conflicting vehicle to evasive maneuvers as well. Additionally, if one wants to drive at an intersection without waiting for everyone else to clear the intersection before, one has to decide to drive even if a collision is a (remote) possibility. Doing so only in case stopping is no longer an option anyways is arguably the most conservative approach possible.

4.2.5.3 Events for Deadlock Resolution

The final set of events deals with the detection and resolution of deadlock situations. At a T-intersection a deadlock can only occur if there is a vehicle approaching from each of the three directions. Additionally, the vehicles entering from below and from the right have to turn left and the vehicle from the left has to drive straight. In this case each vehicle has priority over one vehicle while simultaneously having to yield to the other remaining vehicle. Thus none of the vehicles has priority over all others and the intersection is blocked. As the A-V can be any of the three vehicles, base event e_{b15} checks the turning directions of both the P-V and the Y-V depending on the entry position of the A-V. Base event e_{b15}, alongside the event e_4 is true if a deadlock is possible.

The last two events cover the actual occurrence of a deadlock. e_5 is true if a deadlock is detected. This is the case if the P-V (e_{b3}), the Y-V (e_{b9}) and the A-V are stopped close before the intersection. The A-V is considered as stopped (e_{b16}) if its current velocity is below the threshold for being stopped v_{s}, it is not accelerating and it is closer to the start of the intersection than d_{n}:

$$v^{\mathrm{a}} < v_{\mathrm{s}} \wedge a^{\mathrm{a}} \leq 0\,\mathrm{m\,s^{-2}} \wedge d_{\mathrm{s}}^{\mathrm{a}} < d_{\mathrm{n}}. \tag{4.13}$$

To correctly run the decision-making model, as described in the following section, event e_6 is also needed. It is true if both the P-V (e_{b3}) and the Y-V (e_{b9}) are stopped before the intersection.

4.2.6 Decision-Making Model

The decision-making model itself consists of 11 states and at any time the algorithm is in one of them. Each state is assigned to one of the six zones, and the model can only assume a state that is associated with the current zone. The behavior of the A-V depends on the current state,

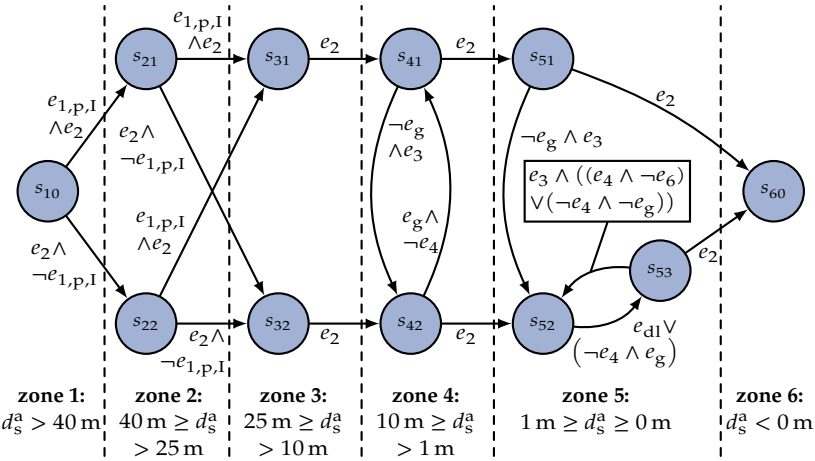


Figure 4.7 Decision-making algorithm for the A-V. If none of the events associated with the current state occurs, the system remains in its current state. For better readability these transitions are not shown here.

unless the L-V's position and velocity prevent it. For that reason there are at least two states in all zones in which the behavior is controlled by the decision-making algorithm, i.e. zones 2 to 5. Zones 1 and 6 only have one state each as the model does not control the driving behavior. In zones 2 to 5 the states are either associated with offensive or defensive behavior. States s_{21}, s_{31}, s_{41} and s_{51} are the states in which the A-V shows offensive behavior, i.e. it attempts to drive through the intersection without waiting. States s_{22}, s_{32}, s_{42} and s_{52} are the defensive states in which the model prepares to wait or actually waits. In state s_{53} offensive behavior is shown after being defensive in zone 5. The decision-making model is shown in figure 4.7 and is presented in detail below.

While the A-V is still far from the intersection, the algorithm always starts in state s_{10}. In it the vehicle is not controlled by the decision-making model. As soon as the threshold to zone 2 is crossed, the prediction for the P-V is evaluated and the system progresses to state s_{21} if the P-V gives green light (event $e_{1,p,I}$). To ensure that the prediction is only run after the A-V entered zone 2, event e_2 has to be triggered as well. That way, the

Table 4.3 Target velocities v_t in $m\,s^{-1}$ for the states of the DES. The values in states s_{42} and s_{52} are set in conjuncture with a virtual vehicle to enforce stopping before the intersection. The large values are set to ensure that the A-V progresses to its designated stop points but are typically nor reached in these states.

state	s_{21}	s_{22}	s_{31}	s_{32}	$s_{41}, s_{42}, s_{51}, s_{52}$ & s_{53}
v_t straight	8.3	6.0	7.5	6.0	6.5
v_t turning	8.3	6.0	6.0	6.0	4.0

prediction is only evaluated once at the beginning of this zone to avoid changing the driving behavior too often and thus potentially confusing the C-Vs. Also, regularly checking the prediction is not yet necessary as the distance to the intersection is still relatively large and there is enough distance and time remaining to react to a changing situation. If the P-V does not give green light, the model progresses to state s_{22}. The behavior, i.e. the current acceleration, depends on a target velocity v_t the A-V is supposed to drive at, see table 4.3 for a complete list. The calculation of the acceleration is introduced in the next section. The target velocity can only be reached if the L-V does not prevent it. For offensive driving v_t is set to $8.3\,m\,s^{-1}$, which is almost the speed limit of $30\,km\,h^{-1}$. In case of defensive behavior the velocity is reduced slightly to $v_t = 6.0\,m\,s^{-1}$ to communicate the intention of defensive driving. After crossing the threshold to zone 3, the same evaluation is run again, if the P-V's traffic light is green, the system progresses to state s_{31}, otherwise it assumes state s_{32}. The target velocity for the defensive state is the same as in zone 2, but in case of offensive behavior the target velocity is set to $v_t = 7.5\,m\,s^{-1}$ in case of driving straight and $v_t = 6.0\,m\,s^{-1}$ in case the A-V turns at the intersection. These velocities are slower than before because even if one can drive first, being below the speed limit increases safety and gives more time to react. In case of turning a lower velocity within the intersection is important anyways to reduce the lateral acceleration especially considering the relatively tight curves at the compact intersections that are considered here.

The transition to zone 4 is done without a prediction, instead the system progresses from state s_{31} to s_{41} or from s_{32} to s_{42}, respectively. As

the prediction is run in every time step in zone 4, skipping the prediction is not a problem. For a more compact representation

$$e_{\mathrm{g}} = e_{1,\mathrm{p,II}} \wedge e_{1,\mathrm{y}} \wedge e_{1,\mathrm{b}} \wedge e_{1,\mathrm{l}} \tag{4.14}$$

is defined. This event occurs, if all four traffic light events of the relevant C-V are true. The model changes from offensive to defensive behavior, that is from state s_{41} to s_{42} if at least one relevant C-V requires it ($\neg e_{\mathrm{g}}$) and if an emergency stop is still possible (e_3). To switch from defensive (s_{42}) to offensive (s_{41}) behavior all lights have to be green (e_{g}) and a deadlock must not be possible ($\neg e_4$). The latter condition prevents the A-V to drive before the P-V, which probably only drives defensively due to the deadlock. In the offensive state the target velocity is set to $v_{\mathrm{t}} = 6.5\,\mathrm{m\,s^{-1}}$ if driving straight and $v_{\mathrm{t}} = 4.0\,\mathrm{m\,s^{-1}}$ in case of turning. If the model is in the defensive state s_{42}, it is set up to stop $d_{1,\mathrm{o}} = 1\,\mathrm{m}$ before its LSP. This is accomplished by placing a stationary virtual vehicle at the intersection, c.f. figure 4.4. To allow the A-V to drive to the desired stop point and to restart if it has to stop earlier within a queue, the target velocity is set to the same values as in the offensive state s_{41}.

Zone 5 is again entered without prediction, the system transitions from state s_{41} to s_{51} or from s_{42} to s_{52}, respectively. The transition from offensive to defensive driving behavior is the same as in zone 4, that is if at least one of the four traffic lights is no longer green ($\neg e_{\mathrm{g}}$) while an emergency stop is still possible (e_3), this transition occurs. However, the transition from defensive to offensive behavior is no longer possible in zone 5. Instead, state s_{53} describes offensive behavior after being in the defensive state s_{52}. The target velocities in the states of zone 5 are the same as the ones in zone 4. State s_{53} is reached from s_{52} either if no deadlock is possible and all four lights are green ($\neg e_4 \wedge e_{\mathrm{g}}$) or if a deadlock occurred and can be solved by the A-V (e_{dl}). This is the case if a deadlock is possible (e_4), it has occurred (e_5) and the A-V is not blocked by either the L-V or the B-V:

$$e_{\mathrm{dl}} = e_4 \wedge e_5 \wedge e_{1,\mathrm{l}} \wedge e_{1,\mathrm{b}}. \tag{4.15}$$

The latter two events are necessary to ensure that the A-V can clear the intersection after passing it. The strategy for resolving a deadlock lies within these transitions: As soon as the A-V detects a deadlock, it

attempts to solve it by starting to drive. This strategy is motivated by the finding of Imbsweiler et al. [181], who found that human drivers prefer to not drive first in case of a deadlock situation at a T-intersection. The model can also go from offensive behavior in state s_{53} to defensive behavior in state s_{52} in case an emergency stop can still be executed (e_3) and either a deadlock is possible but one of the other vehicles is also moving ($e_4 \wedge \neg e_6$) or a deadlock is not possible and one of the relevant C-V no longer gives green light($\neg e_4 \wedge \neg e_g$). From the two offensive states s_{51} and s_{53} the model transitions to state s_{60} of zone 6 as soon as the A-V leaves the intersection and with that also zone 5 (event e_2).

In zone 6 the behavior is no longer controlled by the decision-making algorithm. In this work only the drive through a single intersection is considered but in a more complex street network there could, of course, be further intersections following. To facilitate that, the system could just revert back to state s_{10} after it passes an intersection and it would be ready to approach another intersection.

4.2.7 Behavior Generation

As the A-V follows a fixed path, i.e. the center of its lane, its behavior is only determined by the longitudinal acceleration of the vehicle. The decision on the current acceleration $a^{\mathrm{a}}(t)$, i.e. offensive or defensive driving, itself depends on the current state of the DES model, as described above. Based on the state and the desired turning direction a target velocity v_{t} is selected, see table 4.3. Additionally, the differences in velocity $\Delta v = v^{\mathrm{a}} - v^{\mathrm{l}}$ and distance Δd to the leading vehicle have to be considered. Δd is measured along the center of the lane and describes the distance from the front of the A-V to the rear of the L-V. Finally, the A-V is supposed to stop $d_{\mathrm{l,o}} = 1\,\mathrm{m}$ in front of the LSP if it is in the defensive states s_{42} or s_{52}.

To comply with those constraints, the intelligent driver model (IDM) by Treiber et al. [152] is introduced. It was originally used for traffic simulations on highways considering a single lane. It assumes that vehicles follow a given path and therefore only the longitudinal acceleration is calculated. Previous algorithms for that purpose exist; the original work on the IDM [152] as well as Toledo [151] provide an overview. The IDM is used in several applications, e.g. to predict the turning direction of

drivers at an intersection [94]. Also, there exist several modifications to it. Treiber et al. [153, 154] propose the human driver model (HDM), which additionally incorporates human aspects of driving, it e.g. implements a reaction time and takes further vehicles in front of it into consideration. Zhou et al. [174] introduce a modification to the IDM, the cooperative intelligent driver model (CIDM), to also be able to control merging onto highways. There also exist extensions that enable the IDM to solve aspects of driving through intersections: With the intersection intelligent driver model (IIDM) [131] it is possible to merge into a traffic stream at a T-intersection. Kreutz and Eggert [84] present the generalized intelligent driver model (GIDM) with which they successfully control the interaction at an uncontrolled intersection with another vehicle. They also present a version of the GIDM to drive through lane merge scenarios [83].

Extensions like these are not necessary for the algorithm presented in the current work because the decision-making is done by the DES model and is thus separate from setting an appropriate acceleration. Therefore, the original version of the IDM is used with only minimal modifications. The acceleration based on the IDM is calculated as follows [152]:

$$a^{\mathrm{a}}_{\mathrm{IDM}} = a_{\mathrm{m}}\left(1 - \underbrace{\left(\frac{v^{\mathrm{a}}}{v_{\mathrm{t}}}\right)^4}_{\text{free}} - \underbrace{\left(\frac{d^*}{\Delta d}\right)^2}_{\text{follow}}\right) \tag{4.16}$$

$$\text{with } d^* = d_{\min} + t_{\min} v^{\mathrm{a}} + \frac{v^{\mathrm{a}} \Delta v}{2\sqrt{a_{\mathrm{m}} a_{\mathrm{b}}}}\,. \tag{4.17}$$

Its output, the longitudinal acceleration $a^{\mathrm{a}}_{\mathrm{IDM}}$, depends on the maximum positive acceleration $a_{\mathrm{m}} = 2.5\,\mathrm{m\,s^{-2}}$ and the ratio between the A-V's velocity v^{a} and the target velocity v_{t} as well as the ratio between the desired distance to the L-V d^* and the actual distance to the L-V Δd. The desired distance d^* itself further depends on several parameters and variables: The braking acceleration a_{b} is set to the comfort deceleration $a_{\mathrm{c}} = -2.5\,\mathrm{m\,s^{-2}}$, $d_{\min} = 1.5\,\mathrm{m}$ and $t_{\min} = 1.2\,\mathrm{s}$ are the minimum gaps to the L-V in distance and time, respectively. If there is no L-V, Δd is set to infinity and $\Delta v = 0\,\mathrm{m\,s^{-1}}$.

The concept behind the IDM is as follows [83, 84, 152]: There are two terms, the *free* and the *follow* term. The former describes the model's desire to reach the target velocity v_{t}: the smaller the current velocity of

the A-V, the larger the resulting acceleration. The *follow* term regulates the behavior in case the A-V follows another vehicle by braking if the distance to it becomes too small. As both terms are non-negative, the acceleration can at most be a_m. However, the acceleration is not limited in the negative direction, e.g. the free term becomes large if $v^\mathrm{a} \gg v_\mathrm{t}$. This can occur if the target velocity is set to a new and much lower value after the current state has changed. As this could result in unrealistic braking maneuvers, the deceleration is limited to the desired braking acceleration a_b:

$$a^\mathrm{a} = \max\left(a^\mathrm{a}_\mathrm{IDM}, a_\mathrm{b}\right) . \tag{4.18}$$

A similar approach to ensure plausible accelerations is also used by Kreutz and Eggert [83, 84].

To enforce stopping before the LSP, a virtual L-V is used. This is possible because the IDM ensures that the A-V stops d_min behind a stationary vehicle. If the A-V is in state s_{42} or s_{52}, it is required to stop and a virtual stationary vehicle is placed so that it causes the A-V to stop $d_\mathrm{l,o} = 1\,\mathrm{m}$ before the LSP. Figure 4.4 shows the placement of the virtual vehicle and the stopping position behind it. If there is a real L-V between the A-V and the position of the virtual vehicle, the virtual vehicle is not used.

If in these two defensive states s_{42} and s_{52} the usual value of the braking acceleration $a_\mathrm{b} = -2.5\,\mathrm{m\,s^{-2}}$ is not sufficient for a stop at the desired point, i.e. $\Delta d + 0.2\,\mathrm{m} < d^\mathrm{a}_\mathrm{b}\left(v^\mathrm{a}, a_\mathrm{b}\right)$, the braking acceleration a_b is set to a hard deceleration $a_\mathrm{h} = -4.5\,\mathrm{m\,s^{-2}}$ while the placement of the virtual vehicle remains unchanged. If this is still not sufficient to stop, an emergency stop is initiated. For that the braking acceleration is set to the emergency deceleration $a_\mathrm{e} = -7.5\,\mathrm{m\,s^{-2}}$ and the virtual vehicle is placed in such a way that the A-V stops directly at the last stop point ($d_\mathrm{l,o} = 0\,\mathrm{m\,s^{-1}}$). This is always possible as the model only enters these two states if an emergency stop is still possible (event e_3).

4.3 Simulation Framework

In general, there are several methods to test and validate such a decision-making system. Each of them has certain advantages and disadvantages. The first option is a validation by simulation. With that a large number of cases can be tested safely and relatively quickly. It is also cost-effective

as it can be set up comparatively easily. The downside of this approach is that the behavior of the cooperation partners has to be simulated as well, thus evaluating the reaction of the algorithm to real human driving behavior is difficult if not impossible. Several works from literature employ this approach, e.g. the simulation frameworks *CoInCar-Sim* [117] or *SUMO* [100]. Another option is to use a test vehicle for automatic driving, as it was done with the *Bertha* vehicle [177] or the *Junior* vehicle by Stanford University [89]. This approach probably provides the most realistic results in terms of the interaction between the algorithm and human cooperation partners. The disadvantages are the need for such a vehicle, the effort of integrating the algorithm into the infrastructure of the vehicle, or setting the infrastructure up if it does not already exist and the demand for safety drivers and regulatory challenges. All of these aspects increase the cost and effort while limiting the number of drives that can be performed. The final option are test beds that use model-scale vehicles that execute the decisions by the algorithms, e.g. the *Cyber-Physical Mobility Lab* [82, 140], the *Berkley Autonomous Race Car* [76] or the work by Khayatian et al. [79], who test their intersection management protocol with model vehicles. Especially when using an established system, the effort is presumably comparable to a simulation, but with the added benefit that the model is tested using real vehicles, even at a smaller scale. The number of drives through a scenario that can be done is probably smaller than that of simulations, though. When the focus is on motion planning a benchmark such as the *CommonRoad* [7] is helpful to compare different solutions to each other. This benchmark contains different scenarios that are made up of a map and trajectories of other vehicles and a start and end position for the automation to reach. Using this concept for decision-making is difficult as ideally many diverse situations are required to properly test an algorithm. Also, the pre-recorded trajectories of the other vehicles make it difficult to assess the interaction of an algorithm with its cooperation vehicles, which is an important aspect.

Of these approaches, i.e. simulations, real test vehicles and model-scale test beds, this work uses simulations for testing the proposed decision-making algorithm. In the remainder of this section first the decision-making algorithm for the C-Vs is introduced, including its events, transi-

tions and the behavior generation. The focus is especially on those aspects that differ from the main algorithm. This is followed by an introduction to the simulation setup, i.e. the simulation framework, its properties and its possibilities for parameterization.

4.3.1 Decision-Making Algorithm for Cooperation Vehicles

In order to realistically test the decision-making algorithm of the A-V within a simulation, additional vehicles have to be added to act as the C-Vs at the intersection. To ensure that these vehicles drive realistically, they themselves have to be able to react to current traffic and make decisions at the intersection. The only focus of the simulations is on validating the A-V's algorithm. Therefore the C-V's algorithm has access to the position, velocity and drive path of all vehicles regardless of the vehicles' visibility status or their distance to the intersection. To further challenge the A-V's algorithm, the C-Vs do not always behave as expected. Instead, they can show so-called *special behavior*, which is possible in three different ways: The C-V yields while having the right of way, it drives despite having to yield or it drives slower by a factor $c_{\mathrm{sl,c}}$ than it normally would within and after the intersection. The first two variants are only executed if the vehicle that would be affected by this behavior is the A-V. The algorithm for the C-Vs is a simplified version of the main algorithm.

The vehicle that is running the C-V's algorithm is referred to as A-V_C in the following and its relevant vehicles are marked with the subscript C, e.g. the Y-V_C is the vehicle that has to yield from the A-V_C's perspective. The superscripts of the features remain unchanged, e.g. $t^{\mathrm{a}}_{\mathrm{c,p,e}}$ refers to the time to collision zone of the A-V_C to the end of the collision zone with its P-V_C in this section. Note that the P-V_C in this example could be the A-V, i.e. the vehicle running the algorithm under test, or any of the remaining C-Vs.

The decision-making algorithm for the C-V does not require any new features; all features that are used have been introduced in section 4.2.4.

4.3.1.1 Events

Like the main algorithm for the global A-V, the C-Vs' algorithm is also based on distance-dependent zones, but is has only five zones. The model

Table 4.4 Base events (BE) for the C-Vs, which can show special behavior (SB).

BE	condition	description
$e_{b1,c}$	P-V_C exists	P-V_C exists
$e_{b2,c}$	$d_l^p < 0\,m \wedge d_{c,p,e}^p > 0\,m$	P-V_C within intersection
$e_{b3,c}$	P-V_C: global A-V & SB flag is set	show SB towards P-V_C
$e_{b4,c}$	$(t_{c,p,e}^a - t_{c,p,b}^p) > \Delta t_c^p \wedge$ $(d_{c,p,e}^a - d_{c,p,b}^p) > \Delta d_c^p$	yield to P-V_C
$e_{b5,c}$	$(t_{c,p,e}^a - t_{c,p,b}^p) > \Delta t_{s,c}^p \wedge$ $(d_{c,p,e}^a - d_{c,p,b}^p) > \Delta d_{s,c}^p$	yield to P-V_C under SB
$e_{b6,c}$	Y-V_C exists	Y-V_C exists
$e_{b7,c}$	Y-V_C: global A-V, $t_w > 0\,s$ is set & not waited for at least t_w	show SB towards Y-V_C
$e_{b8,c}$	$d_l^y < 0\,m \wedge d_{c,y,e}^y > 0\,m$	Y-V_C within intersection
$e_{b9,c}$	$(t_{c,y,e}^a - t_{c,y,b}^y) > \Delta t_c^y$	yield to Y-V_C
$e_{b10,c}$	$(t_{c,y,e}^a - t_{c,y,b}^y) > \Delta t_{s,c}^y$	yield to Y-V_C under SB
$e_{b11,c}$	$v^y < v_{sl,c} \wedge a^y \leq 0\,m\,s^{-2}$	Y-V_C waits
$e_{b12,c}$	B-V_C exists	B-V_C exists
$e_{b13,c}$	$d_f^b > l_v + d_{min}$	enough space behind B-V_C
$e_{b14,c}$	L-V_C exists	L-V_C exists
$e_{b15,c}$	$d_s^l < 0\,m$	L-V_C left intersection
$e_{b16,c}$	$v^a < v_{s,c} \wedge d_s^a < d_n$	A-V_C stopped
$e_{b17,c}$	$v^p < v_{s,c} \wedge d_s^p < d_n \wedge a^p \leq 0\,\frac{m}{s^2}$	P-V_C stopped
$e_{b18,c}$	$v^y < v_{s,c} \wedge d_s^y < d_n \wedge a^y \leq 0\,\frac{m}{s^2}$	Y-V_C stopped

is shown in figure 4.8 and is introduced in detail in section 4.3.1.2. The A-V_C is in zone 1 as long as it is farther than 30 m from the intersection. In this zone the algorithm does not distinguish between offensive and defensive behavior, instead the behavior is only dependent on the L-V_C and the initial velocity. As soon as the A-V_C enters zone 2, the first prediction is run and the vehicle shows either offensive or defensive behavior. A second prediction is run when entering zone 3 at a distance of 20 m and a final prediction occurs at the beginning of zone 4 at a distance of 10 m. If the third prediction results in the decision to show

Table 4.5 Events of the DES for decision-making for the C-Vs. Some events are a combination of base events.

definition	description
$e_{1,\mathrm{p,c}} = \neg e_{\mathrm{b1,c}} \vee (\neg e_{\mathrm{b2,c}} \wedge \neg e_{\mathrm{b3,c}} \wedge \neg e_{\mathrm{b4,c}}) \vee (\neg e_{\mathrm{b2,c}} \wedge e_{\mathrm{b3,c}} \wedge \neg e_{\mathrm{b5,c}})$	green light by P-$\mathrm{V_C}$
$e_{1,\mathrm{y,c}} = \neg e_{\mathrm{b6,c}} \vee (\neg e_{\mathrm{b7,c}} \wedge \neg e_{\mathrm{b8,c}} \wedge \neg e_{\mathrm{b9,c}}) \vee (\neg e_{\mathrm{b7,c}} \wedge \neg e_{\mathrm{b8,c}} \wedge e_{\mathrm{b11,c}}) \vee (e_{\mathrm{b7,c}} \wedge \neg e_{\mathrm{b8,c}} \wedge \neg e_{\mathrm{b10,c}})$	green light by Y-$\mathrm{V_C}$
$e_{1,\mathrm{b,c}} = \neg e_{\mathrm{b12,c}} \vee e_{\mathrm{b13,c}}$	green light by B-$\mathrm{V_C}$
$e_{1,\mathrm{l,c}} = \neg e_{\mathrm{b14,c}} \vee e_{\mathrm{b15,c}}$	green light by L-$\mathrm{V_C}$
$e_{2,\mathrm{c}}$	entered next zone
$e_{3,\mathrm{c}}$	deadlock is possible
$e_{4,\mathrm{c}} = e_{\mathrm{b16,c}} \wedge e_{\mathrm{b17,c}} \wedge e_{\mathrm{b18,c}}$	deadlock occurred
$e_{5,\mathrm{c}}$	resolve deadlock timer
$e_{\mathrm{g,c}} = e_{1,\mathrm{p,c}} \wedge e_{1,\mathrm{y,c}} \wedge e_{1,\mathrm{b,c}} \wedge e_{1,\mathrm{l,c}}$	green light by rel. C-Vs
$e_{\mathrm{dl,c}} = e_{3,\mathrm{c}} \wedge e_{4,\mathrm{c}} \wedge e_{5,\mathrm{c}} \wedge e_{1,\mathrm{b,c}} \wedge e_{1,\mathrm{l,c}}$	resolve deadlock

offensive behavior, the A-$\mathrm{V_C}$ drives through the intersection. In case of defensive behavior, the vehicle stops before the intersection and waits. The final zone is entered after the A-$\mathrm{V_C}$ leaves the intersection, i.e. at a distance smaller than 0 m.

The events are similar to those of the A-V's algorithm as well, but due to the simplified algorithm and the special behavior there are some differences. Again, only the four relevant C-Vs (from the A-$\mathrm{V_C}$'s perspective) are considered and also the traffic light analogy is used. All events are listed in table 4.5 and are again a combination of base events, which are listed in table 4.4.

The A-$\mathrm{V_C}$ drives before its P-$\mathrm{V_C}$ if it can safely do so or if its P-$\mathrm{V_C}$ is the global A-V and it is supposed to show special behavior. If the A-$\mathrm{V_C}$ does so, it drives before its P-$\mathrm{V_C}$ even when the safety margins are a lot smaller. The condition $e_{1,\mathrm{p,c}}$ is true, i.e. the A-$\mathrm{V_C}$ can drive relative to its P-$\mathrm{V_C}$, if one of the following conditions is fulfilled:

- The P-$\mathrm{V_C}$ does not exist ($\neg e_{\mathrm{b1,c}}$).

- The P-V_C is not inside the intersection ($\neg e_{b2,c}$), no special behavior is to be shown ($\neg e_{b3,c}$) and the A-V_C can leave the collision zone with its P-V_C with enough lead distance and time in case of no special behavior, i.e. it does not have to yield ($\neg e_{b4,c}$).
- The P-V_C is not within the intersection ($\neg e_{b2,c}$), special behavior is desired ($e_{b3,c}$) and the lead is large enough for the case of special behavior ($\neg e_{b5,c}$).

These base events for green light by the P-V_C ($e_{1,p,c}$) are introduced in the following: The P-V_C is considered to be inside the intersection ($e_{b2,c}$) if its distance to the last stop point is negative while the distance to the end of the collision zone is still positive:

$$d_l^p < 0\,\text{m} \wedge d_{c,p,e}^p > 0\,\text{m}. \tag{4.19}$$

Special behavior is shown ($e_{b3,c}$) if the P-V_C of the A-V_C is the global A-V and a corresponding flag has been set for the A-V_C during the initialization of the simulation run. If the lead of A-V_C over its P-V_C in case of no special behavior ($e_{b4,c}$) is not large enough in time and distance ($\Delta t_c^p = -2.5\,\text{s}$ and $\Delta d_c^p = -15\,\text{m}$) it yields to the P-$V_C$:

$$(t_{c,p,e}^a - t_{c,p,b}^p) > \Delta t_c^p \wedge (d_{c,p,e}^a - d_{c,p,b}^p) > \Delta d_c^p. \tag{4.20}$$

The condition in case of special behavior ($e_{b5,c}$) is similar, but the thresholds are chosen in such a way that the A-V_C drives more aggressively, i.e. it even drives if it is predicted to leave after the P-V_C has entered the common collision zone ($\Delta t_{s,c}^p = 3\,\text{s}$ and $\Delta d_{s,c}^p = 7\,\text{m}$) and only yields if

$$(t_{c,p,e}^a - t_{c,p,b}^p) > \Delta t_{s,c}^p \wedge (d_{c,p,e}^a - d_{c,p,b}^p) > \Delta d_{s,c}^p \tag{4.21}$$

is met. With these conditions the event for green light by the P-V_C is thus defined as

$$e_{1,p,c} = \neg e_{b1,c} \vee (\neg e_{b2,c} \wedge \neg e_{b3,c} \wedge \neg e_{b4,c}) \vee (\neg e_{b2,c} \wedge e_{b3,c} \wedge \neg e_{b5,c}). \tag{4.22}$$

The A-V_C can drive before its Y-V_C if this vehicle does not obstruct the A-V_C, i.e. if the A-V_C is able to leave the common collision zone before its Y-V_C enters it. In case it is supposed to show special behavior, it will not

do so but instead stand at the intersection for a predetermined period of time. This time is set during the initialization of a simulation run. The special behavior is only executed if the C-V in question is supposed to do so, the Y-V_C is the global A-V and the A-V_C does not arrive too much earlier.

To determine if the A-V_C can drive relative to its Y-V_C, some base events are needed: Condition $e_{b6,c}$ is true if the Y-V_C exists. The A-V_C shows special behavior ($e_{b7,c}$) if the Y-V_C is the global A-V, a wait time of $t_w > 0$ has been set during initialization and the A-V_C has not yet been stationary for at least t_w. The waiting time is incremented if the A-V_C is currently in state $s_{42,c}$, a wait time has been set ($t_w > 0$) and if its velocity is below $v_m = 0.15\,\mathrm{m\,s^{-1}}$. It follows from this implementation that the A-V_C switches to its regular behavior towards its Y-V_C after its waiting time is over. Base event $e_{b8,c}$ is true if the Y-V_C is beyond its LSP but has not yet passed the end of the collision zone with the A-V_C: $d_l^y < 0\,\mathrm{m} \wedge d_{c,y,e}^y > 0\,\mathrm{m}$. The A-$V_C$ yields to its Y-V_C ($e_{b9,c}$) if it is predicted to leave the collision zone at least $\Delta t_c^y = 3\,\mathrm{s}$ later than when the Y-V_C is predicted to enters it: $(t_{c,y,e}^a - t_{c,y,b}^y) > \Delta t_c^y$. If the A-$V_C$ has to show special behavior, i.e. it yields to its Y-V_C, it does so even if it is predicted to leave the collision zone up to $\Delta t_{s,c}^y = -3\,\mathrm{s}$ earlier than its Y-V_C leaves it: $(t_{c,y,e}^a - t_{c,y,b}^y) > \Delta t_{s,c}^y$. If this condition is met, base event $e_{b10,c}$ is true. The Y-V_C is considered to be waiting ($e_{b11,c}$), if it is slower than $v_{sl,c} = 1\,\mathrm{m\,s^{-1}}$ and if it is not accelerating: $v^y < v_{sl,c} \wedge a^y \leq 0\,\mathrm{m\,s^{-2}}$. The event $e_{1,y,c}$ for the Y-V_C occurs, i.e. the A-V_C drives relative to that vehicle, if one of the following conditions is met:

- The Y-V_C does not exist ($\neg e_{b6,c}$).
- The Y-V_C is outside the intersection ($\neg e_{b8,c}$), special behavior is not shown ($\neg e_{b7,c}$) and the A-V_C does not need to yield to its Y-V_C ($\neg e_{b9,c}$).
- The Y-V_C is outside the intersection ($\neg e_{b8,c}$), special behavior is not shown ($\neg e_{b7,c}$) and the Y-V_C waits ($e_{b11,c}$).
- The Y-V_C is outside the intersection ($\neg e_{b8,c}$), special behavior is shown ($e_{b7,c}$) and the A-V_C does not need to yield to its Y-V_C even under special behavior conditions ($\neg e_{b10,c}$).

The events for the remaining two relevant C-Vs are defined the same as in the case of the algorithm for the A-V. The event for the B-V_C occurs if the vehicle either does not exist ($\neg e_{b12,c}$) or if there is enough space between it and the end of the intersection ($e_{b13,c}$, $d_f^b > l_v + d_{min}$):

$$e_{1,b,c} = \neg e_{b12,c} \vee e_{b13,c}. \tag{4.23}$$

The A-V_C can drive relative to its L-V_C if the L-V_C does not exist ($\neg e_{b14,c}$) or if it has already left the intersection ($e_{b15,c}$, $d_s^l < 0\,\mathrm{m}$):

$$e_{1,l,c} = \neg e_{b14,c} \vee e_{b15,c}. \tag{4.24}$$

The remaining events of the algorithm for the A-V_C are also very similar or equivalent to those of the A-V. Event $e_{2,c}$ is triggered if the zone has changed in the current time step. The event $e_{3,c}$ is true if a deadlock is possible due to the turning directions of the A-V_C, the P-V_C and the Y-V_C. If $e_{4,c}$ is true, a deadlock has been detected, i.e. all three vehicles in question are stationary close to the intersection:

$$e_{4,c} = e_{b16,c} \wedge e_{b17,c} \wedge e_{b18,c}. \tag{4.25}$$

The conditions for a standing vehicle differ from the original algorithm in that a vehicle is already considered as stationary at a velocity below $v_{s,c} = 0.3\,\mathrm{m\,s^{-1}}$ instead of the slower velocity threshold used for the main algorithm ($v_s = 0.15\,\mathrm{m\,s^{-1}}$). This is true for the A-V_C ($e_{b16,c}$), the P-V_C ($e_{b17,c}$) and the Y-V_C ($e_{b18,c}$). The final event $e_{5,c}$ occurs if the A-V_C decides to resolve a deadlock. This is the case if it has been stationary for a certain duration. The duration is again set during the initialization of a simulation run. The corresponding timer is started if a deadlock is possible, has occurred and the A-V_C would not get stuck within the intersection, i.e. the B-V_C and the L-V_C give green light:

$$e_{dl,c} = e_{3,c} \wedge e_{4,c} \wedge e_{1,b,c} \wedge e_{1,l,c}. \tag{4.26}$$

Even though the A-V's algorithm immediately triggers the vehicle to drive after detecting a deadlock the A-V_C can still go first because its model's threshold for considering a vehicle as stationary is at a higher velocity with $v_{s,c} = 0.3\,\mathrm{m\,s^{-1}}$ than the A-V's model ($v_s = 0.15\,\mathrm{m\,s^{-1}}$). If the waiting time is short enough, it is thus able to still drive first. This is implemented to test the A-V's reaction to someone else resolving a deadlock.

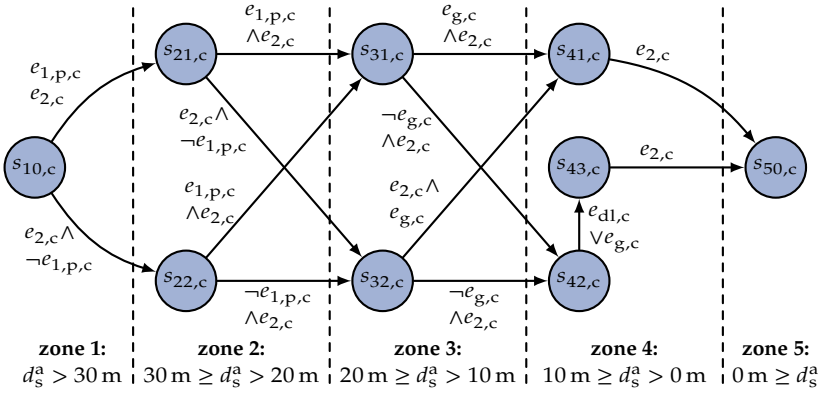


Figure 4.8 Simplified decision-making algorithm for the C-Vs. If none of the events associated with the current state occurs, the system remains in its current state. For better readability these transitions are not shown here.

4.3.1.2 Decision-Making Model

The model for decision-making for the C-V is based on these events and is shown in figure 4.8. While in zone 1 the A-V_C is always in $s_{10,c}$. Directly after entering zone 2 ($e_{2,c}$) a first prediction is run. For that prediction only the P-V_C is considered ($e_{1,p,c}$). If the P-V_C gives green light, the system transitions to the offensive state $s_{21,c}$, otherwise it switches to the defensive state $s_{22,c}$. The same prediction is run when driving into zone 3. The model is in the offensive state $s_{31,c}$ if P-V_C gives green light or progresses to the defensive state $s_{32,c}$ otherwise. In zones 2 and 3 the A-V_C is assigned a target velocity that depends on the current state. The target velocities for all states are listed in table 4.6.

After entering zone 4 the prediction is run again, this time all relevant C-Vs are considered, however. If all vehicles give green light ($e_{g,c}$, see table 4.5 for the definition), the system shifts to state $s_{41,c}$, otherwise state $s_{42,c}$ is assumed. If the A-V_C is in the defensive state $s_{42,c}$, it brakes and stops before the intersection. While in that state, as soon as all traffic lights turn green ($e_{g,c}$) or a deadlock is to be resolved ($e_{dl,c}$, see table 4.1) the vehicle switches to state $s_{43,c}$. This state describes offensive driving after stopping in zone 4. Both offensives states of that zone have in common

Table 4.6 Target velocities $v_{t,c}$ in $m\,s^{-1}$ for the states of the DES for the C-V. The values in state $s_{42,c}$ are set to these high values while a virtual vehicle is present to enforce stopping before the intersection.

state	$s_{21,c}$	$s_{22,c}$	$s_{31,c}$	$s_{32,c}$	$s_{41,c}$, $s_{42,c}$ & $s_{43,c}$
$v_{t,c}$ (straight)	8.3	6.0	7.5	4.0	7.0
$v_{t,c}$ (turning)	8.3	6.0	5.5	4.0	4.0

that the decision to drive through the intersection is not reconsidered but the A-V_C drives through the intersection in any case. After the vehicle enters zone 5 from either state the model transitions to state $s_{50,c}$.

4.3.1.3 Behavior Generation

The behavior, i.e. the longitudinal acceleration, is again calculated using the IDM from (4.16). In state $s_{10,c}$ the initial velocity is used as the target velocity $v_{t,c}$, in the remaining states $v_{t,c}$ is set according to table 4.6. In state $s_{42,c}$ the A-V_C is supposed to stop before the intersection. This is again done by placing a virtual vehicle at the intersection. The vehicle brakes with either a_c, a_h or a_e. In zone 5 the target velocity $v_{t,c}$ is set to the speed limit of $v_{max} = 30\,km\,h^{-1}$.

The final special behavior, i.e. driving slower than usual after the intersection, is implemented by manipulating $v_{t,c}$. For that reason the target velocity in zone 5 is adapted: $\tilde{v}_{t,c} = c_{sl,c} v_{max}$. If the target velocity $v_{t,c}$ in state 4 is larger than the adapted target velocity $\tilde{v}_{t,c}$ in zone 5, it is set to the same value to avoid accelerating followed by decelerating after leaving the intersection: $\tilde{v}_{t,c} = \min\left(v_{t,c}, c_{sl,c} v_{max}\right)$.

4.3.2 Simulation Set-Up

The simulation framework that is used for testing the decision-making model provides all important functionalities for validating the algorithm. First, the current simulation run is initialized. For that purpose the desired number of vehicles is randomly assigned to one of the three entry streets. Further, each vehicle is assigned an initial distance to the intersection and an initial velocity as well as a turning direction. These two

values are set in such a way that the vehicles are placed far enough from each other to avoid collisions before the IDM can adapt their behavior accordingly. If there is not enough space on one of the roads for all vehicles, the vehicles furthest back are removed until there is enough space for all remaining vehicles. The streets in the maps are extended so that this only seldom happens, i.e. if there are too many vehicles placed on the same entry road. Next, one of the vehicles is randomly selected to be the one using the main algorithm from section 4.2. The remaining vehicles use the simplified algorithm described in section 4.3.1. The special behavior towards the A-V described there can be shown by these vehicles. These behaviors are assigned randomly. During a simulation run, at most one vehicle can be assigned the special behavior to drive before the A-V. The remaining C-V can alternatively be set to stop despite having the right of way when their Y-V_C is the A-V. This special behavior can be assigned to multiple C-Vs; each vehicle with that behavior is assigned its random waiting time t_w after which it tries to drive again. The final special behavior makes the C-V drive slowly within and after the intersection. All C-Vs, regardless if they have already been assigned a special behavior, can be assigned to perform that way. If a vehicle is selected for that, the slowing factor $c_{sl,c}$ is randomly drawn. It is important to note that these special behaviors are only seldom assigned; in most simulated runs all C-Vs drive as expected. C-Vs try to resolve a deadlock by driving after a certain waiting period, as introduced in section 4.3.1.2. This waiting period is also randomly assigned during the set-up of a simulation run.

In each time step of the actual simulation, the current state of all vehicles is extracted and given to the decision-making models. For the A-V the vehicles are filtered so that only the data of itself and of those vehicles that are visible in the current time step is provided to the algorithm. Also, the assumption that the turning direction is unknown to the A-V while a C-V is farther than $d_s^x > 10\,\text{m}$ from the intersection is implemented here. The decision-making models are then run for all vehicles and the resulting longitudinal acceleration is integrated twice to obtain the new distance along the drive path. This process is repeated until either the maximum simulation time of 1 min is exceeded or all vehicles have left the intersection. This is the case if the A-V is at least 50 m from the end of the intersection ($d_s^a < -50\,\text{m}$) and all C-Vs are at least 5 m from the end

of the intersection. Finally, a log file containing the states of all vehicles at all time steps is saved.

In a further step the log files are then used to evaluate the runs. This includes the check for collisions and the calculation of the time it took the A-V to pass the intersection t_d, i.e. the time while the A-V was within a distance of $30\,\mathrm{m} \geq d_\mathrm{s}^\mathrm{a} \geq 0\,\mathrm{m}$. As $d_\mathrm{s}^\mathrm{a} = 0\,\mathrm{m}$ holds within the intersection, this time includes both the time during the approach to and within the intersection. Also, the number of C-Vs that interacted with the A-V is counted. A C-V is considered as an interaction partner if it is close to the intersection's beginning ($10\,\mathrm{m} > d_\mathrm{s}^x$) and before the projection of the intersection center onto its path at the same time as this condition was true for the A-V. These C-Vs are then further classified into the categories, P-V, Y-V and others. Finally, some of the log files are used to generate videos of the interaction at the intersection. These are especially helpful for development and additional manual validation.

4.4 Simulation Results

The decision-making model is evaluated using the simulation framework described above. For that purpose several simulations are run. With these the general performance of the algorithm is evaluated using the generic intersection. This is followed by simulations on the maps of the real intersections and finally, the influence of the visibility conditions at the intersections is evaluated.

The main metric by which the simulations are evaluated is the time t_d it takes the A-V to pass the intersection. If the number of simulation runs N_s that are included in a given evaluation exceeds 1, the average time of all these runs is used for the evaluation:

$$\bar{t}_\mathrm{d} = \frac{1}{N_\mathrm{s}} \sum_{i=1}^{N_\mathrm{s}} t_{\mathrm{d},i}, \tag{4.27}$$

where $t_{\mathrm{d},i}$ is the time to pass the intersection in simulation run i.

Table 4.7 Average time to clear the generic intersection $\bar{t}_\mathrm{d}$ in s by the number of interacting C-Vs that are classified as either P-Vs or Y-Vs. The values are averaged over all visibility distances.

	0 P-Vs	1 P-V	2 P-Vs	3 P-Vs	4 P-Vs
0 Y-Vs	9.49	14.02	19.43	22.49	24.26
1 Y-V	9.62	16.79	21.95	25.45	29.87
2 Y-Vs	13.53	19.83	25.28	27.31	36.87
3 Y-Vs	15.87	20.58	24.73	-	-
4 Y-Vs	26.87	15.22	-	-	-

4.4.1 Generic Intersection

The first simulation is run on the map of the generic intersection of figure 4.2. For this simulation the number of C-Vs was set to $N_\mathrm{cv} = \{1, 2, 3, 4, 5, 6\}$, the visibility distance to $d_\mathrm{vs} = \{7\,\mathrm{m}, 14\,\mathrm{m}, 21\,\mathrm{m}\}$ and the simulation is run $N_\mathrm{sc} = 200$ times with different, randomly selected start conditions. This results in 3600 simulation runs. There was no collision between any of the vehicles in all runs. Also, all vehicles were able to clear the intersection during the simulated time span. Averaged over all simulation runs, it took the A-V $\bar{t}_\mathrm{d} = 12.13\,\mathrm{s}$ to clear the intersection with a standard deviation of $\sigma = 6.14\,\mathrm{s}$.

A further important aspect is the number of interaction partners and their influence on the average time to clear the intersection $\bar{t}_\mathrm{d}$. In table 4.7 $\bar{t}_\mathrm{d}$ is calculated separately for each combination of the numbers of interaction C-Vs that are classified as P-Vs and Y-Vs, respectively. The results show that the time to drive through the intersection increases with the number of C-Vs. This increase is more pronounced in the case of the P-Vs.

These findings are plausible because the A-V is required to yield to its P-Vs. The presence of these vehicles thus causes an increase in the waiting time. The more P-Vs are present, the longer the required waiting time. The fact that $\bar{t}_\mathrm{d}$ is also larger in the case of more Y-Vs is probably caused by two factors: Some of the Y-Vs will have driven first due to their special behavior assignment. The second possible aspect is the rather conservative parameterization of the decision-making model. If a Y-V

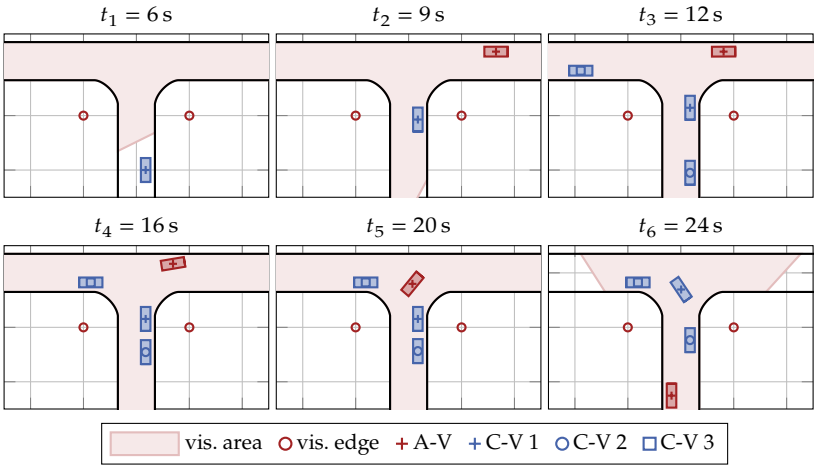


Figure 4.9 Example of a run through the generic intersection. C-V 1 and the A-V arrive first but the situation cannot be resolved before the arrival of C-V 3. The turning directions of the three vehicles (A-V: left, C-V 1: left and C-V 3: straight) result in a deadlock which is resolved by the A-V driving first. After it has passed the intersection, the situation is resolved and the C-Vs have a clear driving order. The grid spacing is 10 m and the visibility distance $d_{\mathrm{vs}} = 7\,\mathrm{m}$.

approaches the intersection too offensively, the A-V's algorithm will often wait until it is certain that the Y-V actually yields, thus increasing the time to pass the intersection.

When interpreting the results from table 4.7 it is important to note that runs with more than three C-Vs the A-V has to interact with seldom occur, $\bar{t}_{\mathrm{d}}$ is thus calculated based on fewer simulation runs than compared to e.g. the value for runs with no C-V and the values are less reliable. The entry for 4 Y-V and no P-V is for example based on only a single run. Also, there are more runs with no P-V (2638 runs) than runs with no Y-V (1626). This is probably caused by the method with which the interaction C-Vs are classified: They are counted as a P-V or Y-V, respectively, if that flag is assigned to them at least once during the simulation run. If the A-V turns into the next street to the right, no vehicle is classified as a P-V, as the A-V does not have to yield to anyone. This is in contrast to the C-V approaching from the next street to the left: In this case it

Table 4.8 Average time $\bar{t}_\mathrm{d}$ to drive through the intersection for all real world intersections, including the standard deviation σ. There are 900 runs through each intersection. For comparison, the average over all runs in which the A-V did not interact with any P-V and Y-V is given by $\bar{t}_\mathrm{d}^0$.

intersection #	1	2	3	4	5	6
$\bar{t}_\mathrm{d}$ in s	13.30	12.00	14.37	14.90	13.35	13.02
σ in s	6.52	6.03	6.81	7.09	6.71	6.53
$\bar{t}_\mathrm{d}^0$ in s	10.65	9.44	11.84	12.47	10.69	10.20

intersection #	7	8	9	10	11	12
$\bar{t}_\mathrm{d}$ in s	12.19	11.08	15.52	15.43	13.69	13.76
σ in s	6.07	5.67	7.39	7.69	6.97	6.83
$\bar{t}_\mathrm{d}^0$ in s	9.33	8.30	13.01	12.77	10.66	10.84

depends on the C-V's turning direction if it is actually a Y-V. As the turning direction is assumed to be unknown at larger distances, many more C-Vs are assigned that flag at some time during the approach as compared to the P-V flag.

In figure 4.9 the positions of the vehicles during one of the simulation runs are shown at selected points in time. Besides the positions of the vehicles at time t_i also the street area that is currently visible to the A-V is shown in light red. Additionally, the corners that determine the visibility are displayed. In this scenario, a deadlock involving the A-V and two of its C-Vs occurs which is solved by the A-V driving first.

4.4.2 Real Intersections in Karlsruhe

For the second simulation the 12 maps of inner-city intersections of the City of Karlsruhe (c.f. figure 4.3) are used. The numbers of cooperation vehicles $N_\mathrm{cv} = \{1, 2, 3, 4, 5, 6\}$ and the visibility distances $d_\mathrm{vs} = \{7\,\mathrm{m}, 14\,\mathrm{m}, 21\,\mathrm{m}\}$ are the same as for the first simulation, but there are only $N_\mathrm{s} = 50$ simulation runs per combination, resulting in a total of 10 800 simulation runs.

All vehicles were able to resolve the situation at the intersections within the simulated time span, thus both algorithms, the one for the A-V and

Table 4.9 Average time to clear the intersection $\bar{t}_{\mathrm{d}}$ in s at intersection 8 by the number of P-Vs and Y-Vs. The values are averaged over all visibility distances. This intersection is an example of an intersection with a small intersection area, cf. Figure 4.3.

	0 P-Vs	1 P-V	2 P-Vs	3 P-Vs	4 P-Vs
0 Y-Vs	8.30	12.47	18.40	21.51	22.93
1 Y-V	8.69	15.29	19.86	24.77	29.77
2 Y-Vs	11.39	21.27	24.35	23.12	24.48
3 Y-Vs	20.02	19.67	-	-	-
4 Y-Vs	-	-	-	-	-

Table 4.10 Average time to clear the intersection $\bar{t}_{\mathrm{d}}$ in s at intersection 9 by the number of P-Vs and Y-Vs. The values are averaged over all visibility distances. This intersection is an example of an intersection with a large intersection area, cf. Figure 4.3.

	0 P-Vs	1 P-V	2 P-Vs	3 P-Vs	4 P-Vs
0 Y-Vs	13.01	17.16	22.99	31.68	31.06
1 Y-V	12.08	20.38	27.18	30.76	29.32
2 Y-Vs	16.95	22.57	22.13	28.87	42.52
3 Y-Vs	18.01	22.32	-	-	-
4 Y-Vs	-	22.13	-	-	-

the one for the C-Vs, were also able to control the vehicles through this more diverse set of intersection maps. In one simulation run, however, the A-V was not able to fully leave the intersection area within the simulation time. This was caused by the relatively large intersection area of intersection 9 and the fact that it had to wait for all C-Vs to pass before. When the simulation time was up the A-V was currently in the process of driving through the intersection area. One similar case occurred at this intersection where a C-V also was unable to cross the intersection in time. The algorithm for the A-V managed to find a collision-free solution in all runs. In case of the algorithm for the C-Vs there are two collisions. Both occur when two C-Vs tried to resolve a deadlock simultaneously. As the decision to drive is not reconsidered by this algorithm, a collision is inevitable in this case. They are ignored as they are caused by the simplifications within the model for the C-Vs and are not an indication

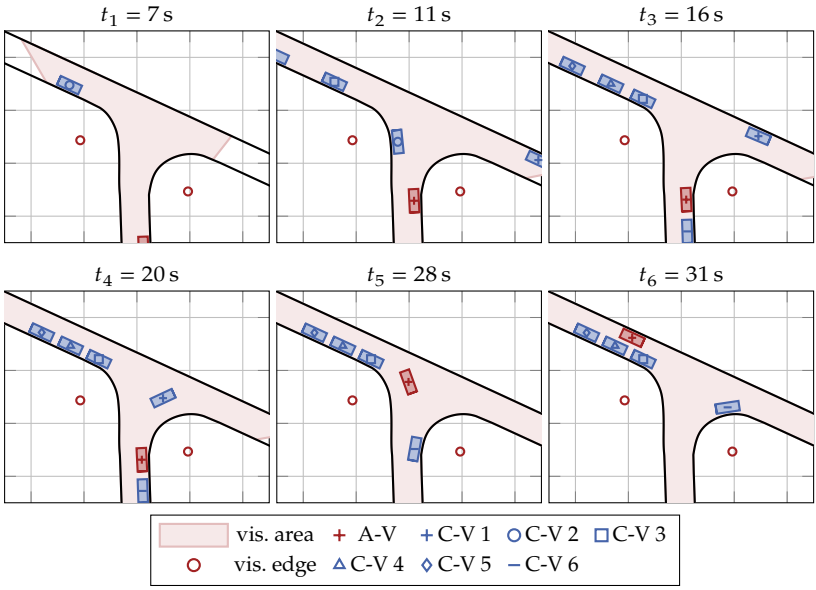


Figure 4.10 Example of a simulation run through intersection 9. C-V 2 passes the intersection before the other vehicles arrive. Then a deadlock occurs between the A-V and C-V 1 (the P-V) and C-V 3 (Y-V). It is resolved by C-V 1 driving first. After that the situation is clearly regulated and the A-V, followed by C-V 6, drives next in accordance with the traffic regulations. The grid spacing is 10 m and the visibility distance is set to $d_{\mathrm{vs}} = 7\,\mathrm{m}$.

of an error within the decision-making model of the A-V. The influence on the average times $\bar{t}_{\mathrm{d}}$ to clear the intersections is also very limited due to their rare occurrence. In one case two C-Vs are set up on top of each other, but due to their different initial velocity they manage to separate from each other far enough from the start of the intersection so that the interaction there is not affected.

In table 4.8 the average time to drive through the intersections is given for this simulation. In that table the average is calculated separately for each intersection, the first rows of the table contain the average time to drive through the intersection for all runs $\bar{t}_{\mathrm{d}}$, the second rows contain the standard deviations for that value. The third rows contain the average time to drive through the intersection where only those runs are

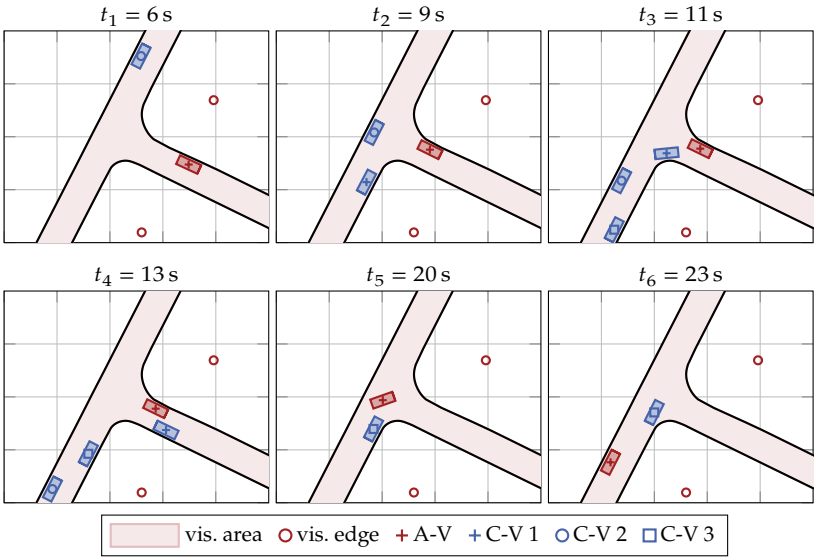


Figure 4.11 Example of a run through intersection 2: Both a P-V and a Y-V exist, but no deadlock occurs due to their turning directions. C-V 2 (the P-V) passes first, followed by the A-V. However, the A-V only passes after it is certain that C-V 3 (the Y-V) actually stops. C-V 1 also passes the intersection in the meantime but as it turns right is not involved in the interaction. The grid spacing is 10 m and the visibility distance is set to $d_{\mathrm{vs}} = 14\,\mathrm{m}$.

considered in which no interacting C-V is present ($\bar{t}_{\mathrm{d}}^{0}$). For comparison, in case of the generic intersection $\bar{t}_{\mathrm{d}}^{0} = 9.49\,\mathrm{s}$ (c.f. table 4.7).

These results show that both $\bar{t}_{\mathrm{d}}$ and $\bar{t}_{\mathrm{d}}^{0}$ differ substantially for the various intersections. The difference between intersection 8, which has the lowest times and intersection 9, where the times are largest, is greater than 4 s for both $\bar{t}_{\mathrm{d}}$ and $\bar{t}_{\mathrm{d}}^{0}$. When comparing the times with the plots of the maps of figure 4.3 a clear pattern is visible: the larger the intersection area, the larger the time to clear it. This is especially apparent for intersections 9 and 10, which have the largest times to clear them. The area where the lanes overlap are relatively large, which results in longer distances and thus longer times through the intersections. This is also true for the opposite direction; intersections 8 and 2 are relatively

compact, consequently $\bar{t}_\mathrm{d}$ and $\bar{t}_\mathrm{d}^0$ at these intersections are the lowest and second lowest, respectively, of all intersections.

In table 4.9 $\bar{t}_\mathrm{d}$ depending on the number of P-Vs and Y-Vs the A-V has to interact with is shown for intersection 8, the intersection with the lowest overall $\bar{t}_\mathrm{d}$. The same table for the intersection with the largest overall $\bar{t}_\mathrm{d}$, intersection 9, is presented in table 4.10. In both cases the same phenomenon as in the case of the generic intersection can be observed: The average time $\bar{t}_\mathrm{d}$ increases with more C-Vs the A-V interacts with and this increase is more pronounced in case of the P-Vs than compared to the Y-Vs. As in table 4.7 the entries of the cases with more C-Vs are less reliable as there are fewer cases. This effect is increased by the lower number of simulated runs per combination ($N_\mathrm{s} = 50$).

The performance of the decision-making algorithm on the maps of real intersections is showcased in figure 4.10 and in figure 4.11. The first scenario shows one of the simulation runs at intersection 9. In it the A-V and six C-Vs approach the intersection where a deadlock between the A-V and two of the C-Vs occurs, which is resolved by one of the C-Vs. The latter example is taken from a simulation run at intersection 2. It shows the interaction of the A-V with two C-Vs. The A-V first lets its P-V pass and as soon as it is certain that the Y-V stops, the A-V drives through the intersection. Additional examples of simulation runs can be found in appendix C.

4.4.3 Influence of Limited Visibility

So far the visibility distance has not been taken into consideration, instead the average time $\bar{t}_\mathrm{d}$ to pass the intersection has been calculated of all available visibility distances in the evaluations for the previous two sections. One could, however, argue that poor visibility conditions lead to other traffic participants being visible later and thus force the model to approach the intersection more cautiously and slower. A reduction in the speed would lead to increased time to clear the intersection t_d. This argument is further supported by the event e_b1 that ensures that the A-V may only drive if there cannot be a P-V due to its own turning direction or if the street the P-V would approach on is sufficiently visible.

To investigate that, $\bar{t}_\mathrm{d}$ is calculated separately for each of the visibility distances $d_\mathrm{vs} = \{7\,\mathrm{m}, 14\,\mathrm{m}, 21\,\mathrm{m}\}$ of the 3600 simulation runs on the

generic map: $\bar{t}_\mathrm{d}\,(d_\mathrm{vs} = 7\,\mathrm{m}) = 12.10\,\mathrm{s}$ with a standard deviation of $\sigma = 6.03\,\mathrm{s}$; $\bar{t}_\mathrm{d}\,(d_\mathrm{vs} = 14\,\mathrm{m}) = 12.14\,\mathrm{s}$, $\sigma = 6.18\,\mathrm{s}$ and $\bar{t}_\mathrm{d}\,(d_\mathrm{vs} = 21\,\mathrm{m}) = 12.16\,\mathrm{s}$, $\sigma = 6.23\,\mathrm{s}$.

These results indicate that there is no dependence on the visibility distance. However, these values are calculated by averaging all runs with the same visibility distance and thus the effects of it could be masked by other, stronger influences. Also, the visibility distances d_vs of the previous simulations may be too large to have any substantial influence on the time as large areas of the intersections are visible even when the A-V is still farther from the intersection (c.f. figures 4.9, 4.10 and 4.11). For further investigation an additional simulation is conducted on the generic map and on intersections 8, 5 and 9 where the visibility distance is set to values in 1 m or 2 m increments: $d_\mathrm{vs} = \{1\,\mathrm{m}, 2\,\mathrm{m}, \ldots, 10\,\mathrm{m}, 12\,\mathrm{m}, \ldots, 22\,\mathrm{m}\}$. These intersections are included because they have the lowest (intersection 8), an intermediate (intersection 5) and the highest (intersection 9) $\bar{t}_\mathrm{d}$ in the simulation of section 4.4.2. The number of C-Vs remains unchanged with $N_\mathrm{cv} = \{1, 2, 3, 4, 5, 6\}$ and the number of simulation runs per combination is set to $N_\mathrm{s} = 50$, which results in 19 200 additional simulation runs. Again, no collisions involving the A-V occurred, but in three runs collisions between C-Vs could be observed, all were caused by attempting to resolve a deadlock simultaneously.

The average time to pass $\bar{t}_\mathrm{d}$ for the different visibility distances d_vs of these simulations is given in figure 4.12 (*full sim.* variant in the plots). At those distance where the A-V is able to pass the intersection, the standard deviations are in a similar range as they are when considering the three original visibility distances combined, as is reported in table 4.8. The results show a pattern similar to the analysis that uses only the three visibility distances: There is little difference between $\bar{t}_\mathrm{d}$ at different distances while the standard deviation is relatively high, which prohibits a clear conclusion regarding a possible influence of d_vs. There are no results shown for very small values of d_vs because the model for the A-V gets stuck at the intersection in some runs when the visibility distance is that small. This effect is discussed in detail below.

To investigate the influence of the visibility distance on the time to clear the intersection without any disturbing factors, a final simulation is performed. For that purpose only the A-V is placed at the intersection.

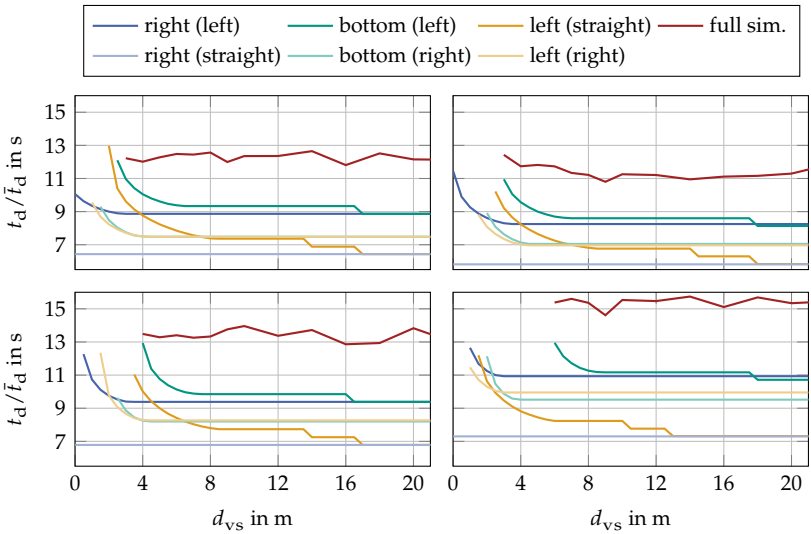


Figure 4.12 Time to pass the intersection t_d over the visibility distance d_{vs} at four different intersections for all three possible entry positions and the two driving directions. The times are recorded by initializing the A-V at a constant distance and with a constant initial velocity before the intersection while there are no additional C-Vs. As this procedure is deterministic, each variant is run only once. Times for a given d_{vs} are only reported if the A-V is able to clear the intersection and does not get stuck due to insufficient visibility over the intersection. The time t_d is measured for all three entry directions. First row: generic intersection (left) and intersection 8. Second row: intersection 5 (left) and intersection 9. Additionally, $\bar{t}_d$ of the full simulation with multiple vehicles and random initial conditions is given.

All runs are initialized at the same distance and with the same velocity. Two variants are simulated for each entry position, one where the A-V turns so that a P-V can exist (drives straight when entering from the left, turns left when entering from the bottom or right) and one where a P-V cannot exist as the driving paths do not intersect (driving straight when entering from the right or driving right when entering from the left or bottom). The visibility distance d_{vs} is set to values between 0.5 m and 21 m in 0.5 m increments. Additionally, $d_{vs} = 0.01$ m is simulated. This value is used as placing the visibility corner directly onto the curb cannot be simulated by the framework in its current form. As these simulations

have deterministic results, only one simulation run is performed per variant.

The results of the simulations are also shown in figure 4.12. The time to pass the intersection t_d for the generic map is presented alongside those for intersections 8, 5 and 9. There is no reported t_d if the A-V gets stuck at the intersection due to insufficient visibility. In this simulation a dependence of the driving behavior on the visibility distance is observable. The general shape of the curves is the same for the four intersections and also holds for the remaining intersections, which are not shown here.

First the results for the variants are presented where the A-V turns so that a P-V can exist: In most cases the A-V is unable to clear the intersection for very small visibility distances d_vs. When it is able to clear it, the time to clear the intersection t_d decreases continuously with increasing d_vs. For larger distances the time to pass is constant, but if the A-V enters the intersection from the left street, t_d drops abruptly twice at certain values of d_vs to a new constant value. The same phenomenon is visible when entering from the bottom, but there is only one drop of t_d in this case. The continuous decrease in t_d exists for all three entry positions; the extent and the distances, however, at which this effect is visible differ. It is further common among all intersections that entering from the right converges to a constant value at the smallest values of d_vs. Of all three entry positions, t_d reaches the smallest values when entering from the left.

No P-V can exist if the A-V turns into the next street to the right. If the A-V enters from the right, in this case its t_d is constant over all visibility distances. In case it enters from one of the other two directions, t_d decreases to a constant value for increasing visibility distance d_vs. This pattern is again common for all intersections, but the values of t_d and d_vs depend on the intersection.

All of these phenomena can be explained: The initial continuous decrease in t_d is caused by the distance to the intersection of the A-V at which it is able to see the reference point (see section 4.2.3) on the streets of the P-V and of the B-V, respectively. If the visibility is poor, which is the case for low values of d_vs, the reference points are not visible when the A-V enters zone 4. The A-V thus remains in the defensive state s_{42} until the reference points become visible, which is when it switches to

offensive behavior as there are no further C-Vs present. The continuous curve results from the fact that in zone 4 the model is re-evaluated at every time step. The reference point on the street of the B-V is only relevant in zone 4; if it is not visible before, it does therefore not have an influence on the driving behavior in the earlier zones. The P-V, in contrast, is also evaluated in the earlier zones and if the reference point associated to it is not visible before, it causes a more defensive driving behavior. As the transitions of the model are only evaluated when entering zones 2 and 3, t_d changes abruptly if the A-V is able to drive offensively in these zones at a certain value of d_vs. The different number of steps for the different entry positions can also be explained: When entering from the right, a P-V would enter from the left and is therefore visible from the start. In this case only the B-V is not directly visible, which only has an influence in zone 4. Because of that there are no steps in t_d. When entering from the left or bottom, the street of the P-V is at an angle and thus the visibility is important. When entering from the left, two steps in t_d can be observed caused by the A-V being able to drive offensively in zones 2 and 3, respectively. Entering from the bottom results in only one step; this is caused by the fact that offensive and defensive behavior in zone 3 have the same target velocity ($v_\mathrm{t} = 6.0\,\mathrm{m\,s^{-1}}$) when the A-V turns.

If there cannot be a P-V, only the reference point on the street of the B-V influences the time to clear the intersection. If the A-V enters from the right, it has to drive straight in order to ensure that there cannot be a P-V. If this is the case, the reference point of the B-V is visible from the start, t_d is therefore constant over all visibility distances d_vs. In case the A-V enters from the other two positions, turning right ensures the non-existence of the P-V. Because of that the reference point is not necessarily visible at large distances of the A-V from the intersection and thus the time to pass the intersection is larger for small visibility distances as the A-V has to get closer to the intersection before the point becomes visible.

The effects of the visibility distance d_vs on the visible area at an intersection are visualized in figure 4.13. There the visible area is plotted at two intersections, the generic one and intersection 9. For both intersections a visibility distance of $d_\mathrm{vs} = 1.5\,\mathrm{m}$ and $d_\mathrm{vs} = 6\,\mathrm{m}$ is shown. As the current visibility conditions also depend on the distance of the A-V to

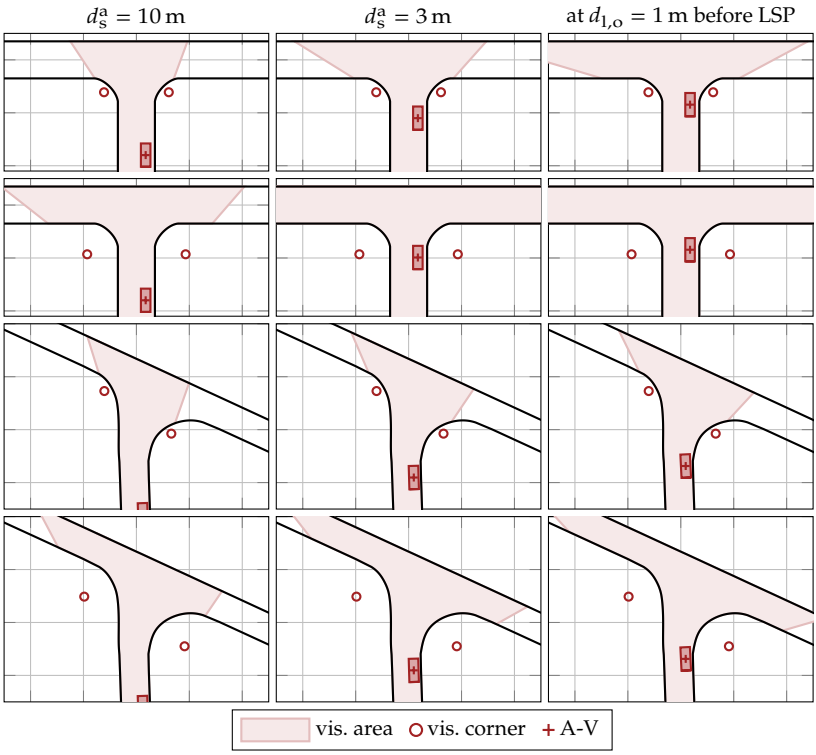


Figure 4.13 Visualization of the visible area at two intersections. In rows 1 and 3 the visibility distance is $d_{vs} = 1.5\,m$, in rows 2 and 4 it is $d_{vs} = 6\,m$. The first two rows show the results from the generic intersection, the third and last row show intersection 9.

the intersection, the situations when the A-V is $d_s^a = 10\,m$ (first column), when it is $d_s^a = 3\,m$ (second column) and when it is $d_{1,o} = 1\,m$ from its last stop point when turning left (last column) are shown for all four variants[6]. $d_s^a = 10\,m$ is important because the A-V transitions to zone 4 at that distance where defensive behavior does not only mean a slightly

[6] As the simulation is run only for discrete time steps, these exact positions of the A-V cannot be generated. Instead, the simulation frame in which the A-V is closest to the desired distance is shown. These deviations are relatively small, the maximum value is approx. 5 cm.

slower target velocity v_t but in that zone the A-V is actually slowed down all the way to stop. If a potential P-V is not visible at this distance, the A-V is slowed down substantially from this point on and thus the travel time will increase rapidly. The position $d_{l,o}$ before the last stop point is important because the A-V is usually stopped there if it shows defensive behavior. $d_s^a = 3\,\text{m}$ is chosen as an intermediate value.

The real-world visibility distances d_{vs} of the intersections used here for simulation vary considerably and range from approximately 4 m, usually in denser parts of the city closer to the center, to more than 10 m in the more spacious areas, typically more towards the outskirts of the city. These values are measured using publicly available online satellite image sites. However, at real intersections there are oftentimes parked vehicles or other items that may block the visibility further than only stationary objects like buildings or vegetation. This makes relatively small values like $d_{vs} = 1.5\,\text{m}$ relevant as well.

As stated above, the A-V is unable to pass the intersection for very small values of d_{vs}. The values at which this is the case depend on the intersection and the entry position, but all these cases are caused by the fact that the reference point on the street of either the P-V or the B-V is not visible during the entire approach, which keeps the A-V in the defensive states of zones 4 and 5 (states s_{42} and s_{52}). This, in turn, forces the A-V to stop $d_{l,o}$ before the last stop point. As the model only continues driving if it is certain that there either is no P-V and B-V respectively, or that it can drive safely despite their existence, it does not recover and it is stuck at the intersection with the current version of the algorithm.

4.4.4 Conclusions

The results of these simulations show that the decision-making algorithm which is introduced in this chapter is able to safely drive through maps of inner-city intersections without causing collisions. Sufficient visibility provided, it is also able to clear the intersection without getting stuck there. The model further shows reasonable behavior; this is supported by the fact that the time to drive through the intersection increases with more C-Vs and especially with more P-Vs. Finally, the algorithm generalizes well to different intersection geometries, which is illustrated by its ability to safely drive through maps of several real-world intersections. The

time to pass these intersections also correlates well with the area of the intersection; larger intersection areas cause longer average travel times. This further indicates a reasonable performance.

From the visibility areas depicted in figure 4.13 one can conclude several aspects: A certain minimum visibility distance is required for the A-V to see far enough into a potential P-V's street; this is especially true for intersections with small angles between their streets like intersection 9. If d_{vs} is large enough for the A-V to eventually pass, the relevant parts of an intersection are visible even if the A-V is still relatively far from the intersection. Thus the A-V decelerates for a comparatively short period only, which further contributes to the limited influence of the visibility conditions on the driving time.

In conclusion one can say that there is an effect of the visibility distance d_{vs} at an intersection on the time to pass it t_{d}. It is, however, also apparent that this effect is rather weak and vanishes for larger values of d_{vs}. Comparing the results of the different simulations indicates that other aspects like the number and type of C-Vs have a far larger influence on the time to pass the intersection. If there actually is a P-V present, the A-V has to wait regardless of the visibility conditions, which increases t_{d} to a greater extent.

4.5 Summary & Outlook

In this chapter an algorithm for decision-making for automatic vehicles at inner-city T-intersections is proposed. The algorithm utilizes a discrete event model, which allows the model to be easily explainable while also having only moderate requirements regarding calculation times. This is evident if one considers the type of calculations that are required for this approach. The decision-making is based on features that can be efficiently calculated from the observations of the C-Vs. These features then trigger events which themselves are responsible for transitioning through the DES for decision-making. All these calculations require very little computational effort.

The model is evaluated and validated using a simulation framework. This framework additionally models vehicles that serve as cooperation partners for the vehicle controlled by the model. These additional vehi-

cles utilize a decision-making algorithm that is a simplified version of the proposed one. It also shows unexpected special behavior to further challenge the main model. The framework can simulate on arbitrary maps of intersections; in this work a generic map and 12 maps of real intersections are used for that purpose. To generate a wide variety of situations, the framework randomly initializes all vehicles for each simulation run.

The simulation results show that the model is able to solve the decision-making task reliably at several different intersections. There is no collision of the A-V with any of its C-Vs; also, the algorithm is always able to pass the intersection during the simulation time, if the intersection is sufficiently visible. It is further able to deal with unexpected behavior, which includes driving despite not having the right of way. The model itself always respects priority regulations and only breaks these in order to solve a deadlock or a situation in which another vehicle yields for some time despite not having to.

The simulation results indicate that the decision-making model works as intended: The more C-Vs are present on the map, the larger are the average travel times through an intersection. As expected, vehicles the A-V has to yield to cause the largest increase in the time to pass the intersection. Also, the visibility conditions influence the driving time as expected; worse visibility causes larger travel times through an intersection. This effect is small, though.

The proposed model is already able to cope with many situations and aspects of inner-city driving. There are, however, some scenarios that can benefit from future improvements: The most important aspect in that regard is certainly the behavior of the A-V in case of insufficient visibility in an intersection. Even though the model is able to drive through all the maps that are part of the simulations at realistic visibility distances, these distances can be reduced in reality.

To solve this, the model should initially approach the intersection all the way to the last stop point, this would probably reduce the risk of getting stuck. But it would also be important to include a mode in which a vehicle that cannot be certain if there is another vehicle approaching would still, very cautiously, slowly enter the intersection to improve the visibility. This is similar to how human drivers behave in this situation. In order for the algorithm to be able to avoid obstacles along the center

line path, it would be beneficial to implement path planning and enable the algorithm to follow paths other than the center line.

Currently, the model only supports T-intersections; adapting the algorithm to work at X-intersections should be possible using the same approach. Similar algorithms for roundabouts and narrow passages could also be developed. This, however, would presumably require more changes to the decision-making model. Finally, the parameterization of the algorithm could be improved by making it adaptable to specific situations or by making some parameters adjustable by the passengers, enabling them to have the vehicle drive according to their preferences, e.g. more defensively.

5 Conclusion

This thesis contributes to the field of autonomous driving in two areas: The driving behavior of human drivers is evaluated and predicted in chapters 2 and 3; this is based on the concept of intersection complexity. Also, a decision-making algorithm for autonomous vehicles at non-signalized intersections is proposed in chapter 4. This chapter first gives a summary of this work, then an outlook on potential future work is given.

5.1 Summary

In chapter 2 the influence of the surroundings at an intersection on the driving behavior is investigated. Based on data of a field study, several features of an intersection are defined and successfully used to predict the driving behavior. The latter is defined by behavior features. The set of intersection features is assumed to constitute the complexity of an intersection. To identify the most important complexity features, several investigations are performed and the free space at an intersection is found to be most relevant for predicting the behavior. Also, the feature set is reduced by using only the most relevant features for prediction and by combining all features into fewer ones with the help of an autoencoder. Both approaches show reduced predictive power compared to the full feature set.

The fact that several complexity features are needed for behavior prediction implies that the concept of complexity is limited in its interpretability. To overcome that challenge, a one-dimensional, or at least low-dimensional, complexity is desirable. This would make statements such as “increased complexity leads to more defensive driving” feasible. This is the goal of chapter 3. In that chapter, based on a second study of pairwise comparisons, complexity scores for intersections are derived.

These are then used for behavior prediction as well. Using these scores on their own is not sufficient for reliable predictions, but combining them with the entry position does allow good predictions, while simultaneously improving one's ability to understand and explain the reasons for the expected driving behavior of human drivers.

The final contribution of this thesis is a decision-making algorithm for autonomous vehicles at non-signalized T-intersections in inner-city traffic. The algorithm does not require any communication between the vehicles but is instead able to make safe decisions only based on the observable states of its cooperation vehicles at the intersection. The algorithm is based on a discrete event system; its decisions are therefore easy to understand, which might contribute to better acceptance. The algorithm is extensively tested in simulations; this includes simulations using maps of real-world intersections.

5.2 Outlook

There are some aspects of this work that could be improved upon in future work. The set of complexity features used in chapters 2 and 3 has the potential to cover even more aspects. For that purpose additional features would have to be considered as well. One aspect that is currently not directly considered are parked vehicles; a feature like the number or the density of parked vehicles at or by the road could be added. A further aspect that future work could be focusing on is the traffic at the intersection. In the current data set most runs were without any additional vehicles; this is the most likely reason why these features have little relevance for prediction. To investigate the influence traffic actually has on the driving behavior, a larger data set would be necessary containing more runs that include interaction with traffic at the intersections.

The fact that using only the complexity scores of chapter 3 for behavior prediction does not yield good results indicates that calculating the intersection's complexity score based on its complexity features should be improved in future work. The better prediction results when using the direct assignment method further supports that assumption. To remedy this, additional complexity features could be added, as noted above, and

a larger video study with additional intersections could help to improve the predictive power of the complexity score.

The decision-making algorithm introduced in chapter 4 is able to make vehicles drive through T-intersections reliably. Extending this algorithm to X-intersections is the next obvious step and should be possible with only minor adaptations. Similarly, this type of algorithm could also be used for roundabouts and narrow passages. Some aspects of the current algorithm could also be improved: As is evident from the visibility investigations, the vehicle controlled by the current algorithm can get stuck if visibility is too poor. To circumvent this, the vehicles using the algorithm should be able to carefully and slowly enter the intersection, as is allowed by the German road regulations[1]. The parameterization of the algorithm could be improved or potentially adapted for different circumstances. Finally, if the vehicle did not always follow the center of its lane but instead follow a planned path, obstacles on the road that block the path along the lane center like parked vehicles would not prevent it from using the algorithm.

[1] § 8 Abs. 2 StVO

Appendix

A Behavior Prediction Results – Complexity Features

In chapter 2 the driving behavior is predicted using features describing intersections. The behavior itself is described by behavior features and regression models are trained to predict them based on the intersection features. Models using several different sets of intersection features are trained and a selection of scatter plots of these models are shown here. The plots of each behavior feature are given in a separate figure; figure A.1 shows those for the commit distance d_c, figure A.2 contains scatter plots when using the minimum velocity v_m and figure A.3 those of the velocity drop v_d. The first three rows of these figures contain plots for the T-intersection data; the bottom three rows result from using the X-intersection data set. The first rows show the scatter plots when using the full feature set (FS); these are already shown in figure 2.3 and figure 2.4, respectively, and are included here for easier comparison. The composition of the full feature set is introduced in section 2.5.1. In section 2.5.3 only a reduced feature set is introduced; the scatter plots that result from that are shown in the second columns of the figures. The third rows contain the plots for those models that use only the direction features ($\{p_e, p_t\}$ for the T-intersections and $\{p_t\}$ for the X-intersections). Finally, the last rows show the scatter plots of the 1D autoencoder models, c.f. section 2.5.4.

The scatter plots support the results described in chapter 2 itself. The models trained with the full feature set have scatter plots where most points are close to the ideal line. A similar effect is visible for the reduced feature set. The direction only feature sets show the characteristic vertical lines; these are caused by the fact that only six (three for the X-intersections) regression values are possible. The autoencoder models result in worse predictions; this causes the point clouds to be more spread out.

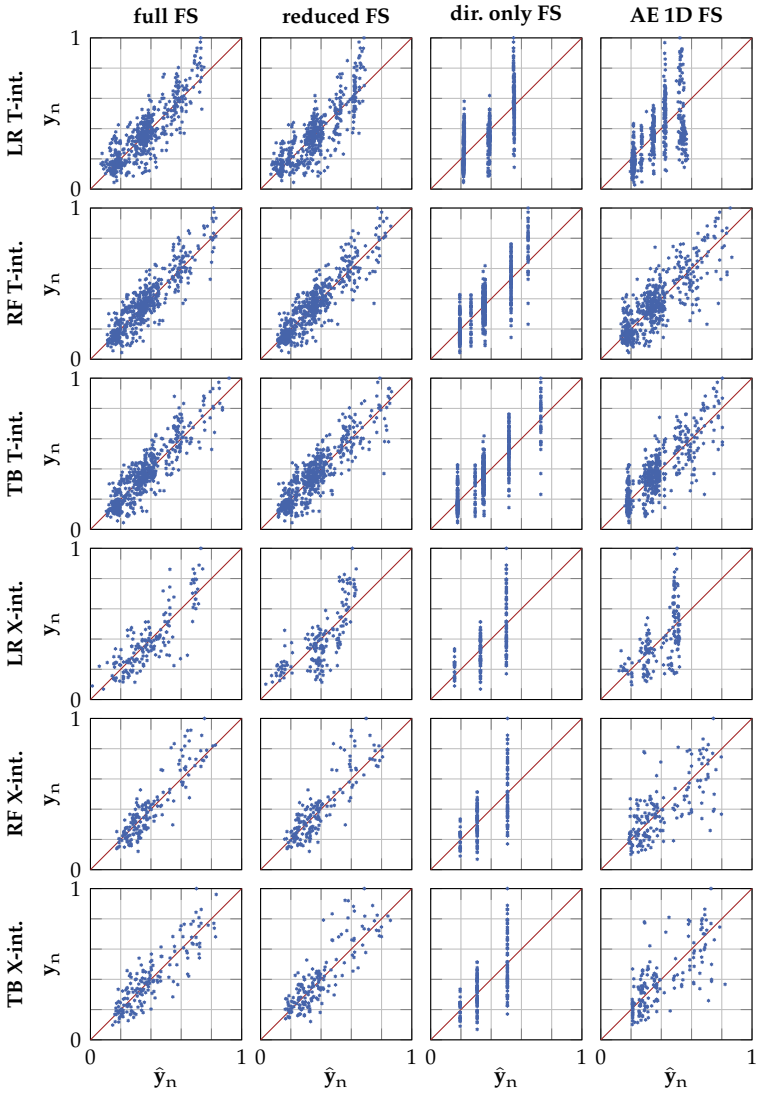


Figure A.1 Scatter plots using different feature sets and algorithms with the commit distance d_c as the label at both the T- and X-intersections.

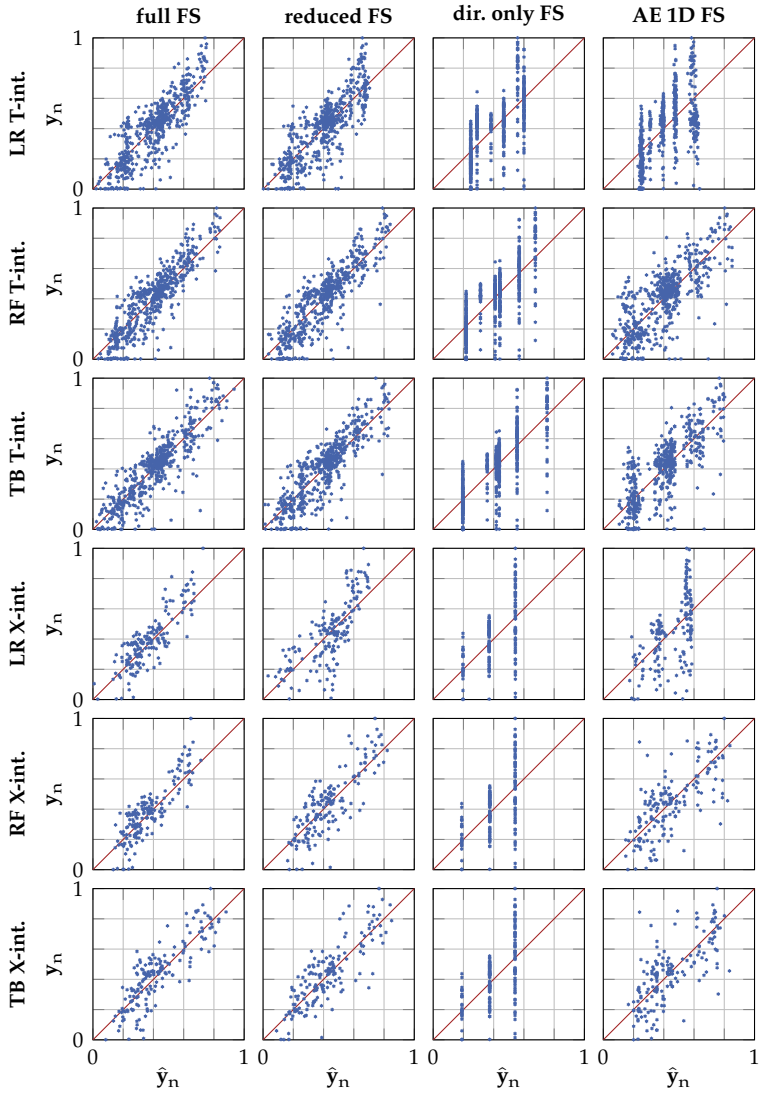


Figure A.2 Scatter plots using different feature sets and algorithms with the minimum velocity v_m as the label at both the T- and X-intersections.

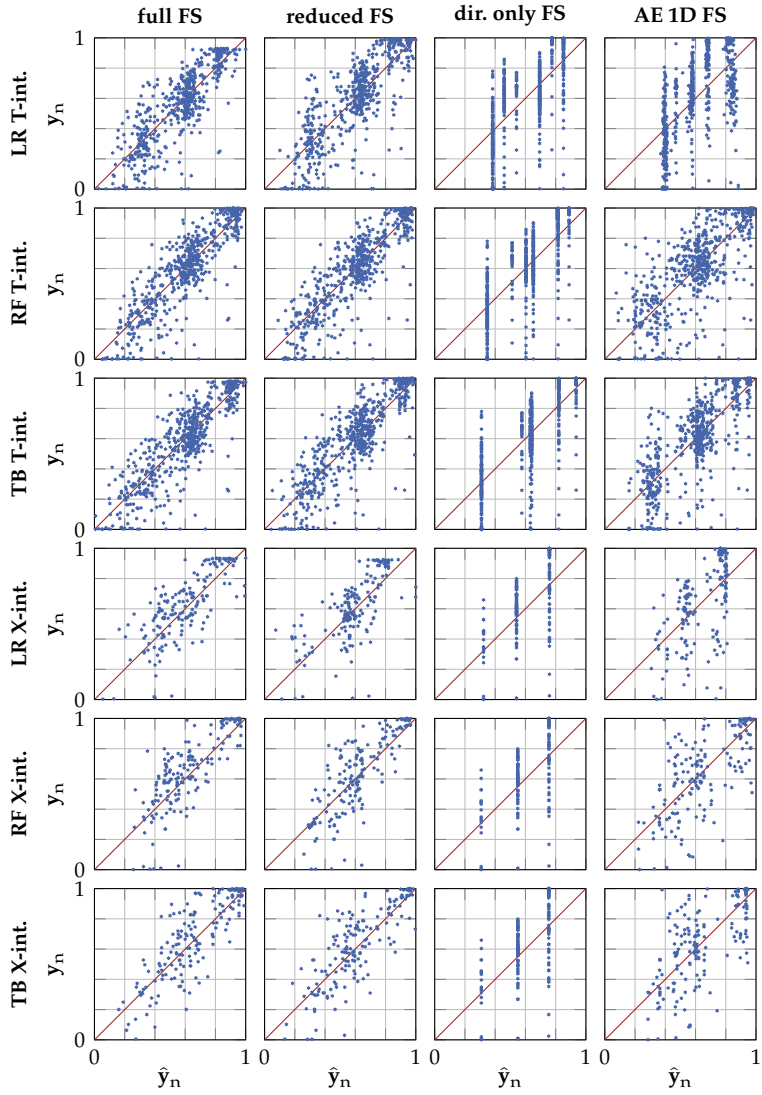


Figure A.3 Scatter plots using different feature sets and algorithms with the velocity drop v_d as the label at both the T- and X-intersections.

B Behavior Prediction Results – Complexity Measure

In section 3.4 the driving behavior is predicted by using complexity scores derived from an online study. In that section only the performances of the models using the commit distance d_c as the behavior feature are reported. Here, the results of the remaining behavior features are given. The selection of scatter plots is the same as is shown in figure 3.2; the corresponding scatter plots for the minimum velocity v_m are given in figure B.1, those for the velocity drop v_d in figure B.2. Additionally, the tables with the prediction errors in RMSE are given in table B.1 for v_m and in table B.2 for v_d. These tables contain both the results of the evaluations on the data set of those runs that are represented in both studies (top

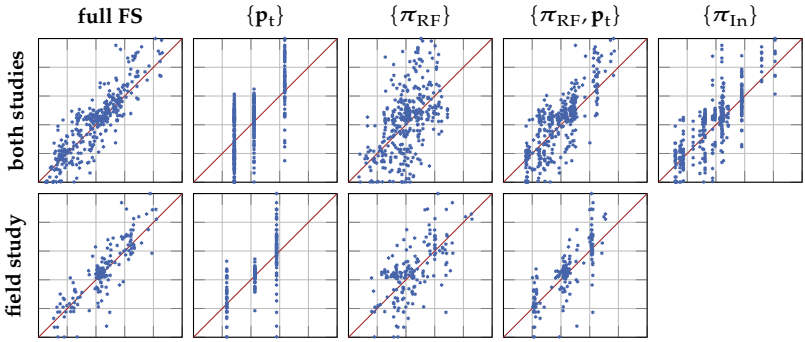


Figure B.1 Scatter plots using RF for prediction and selected feature sets with the minimum velocity v_m as the label using the data sets at the T-intersection/p_e combinations that were in both studies (top row, data set $\mathcal{D}_{iv}^{F}$), and that were in the field study only (bottom row, data set $\mathcal{D}_{ov}^{F}$), respectively. The abscissa shows the predicted normalized labels $\hat{\mathbf{y}}_n$, the ordinate shows the normalized true labels $\mathbf{y}_n$.

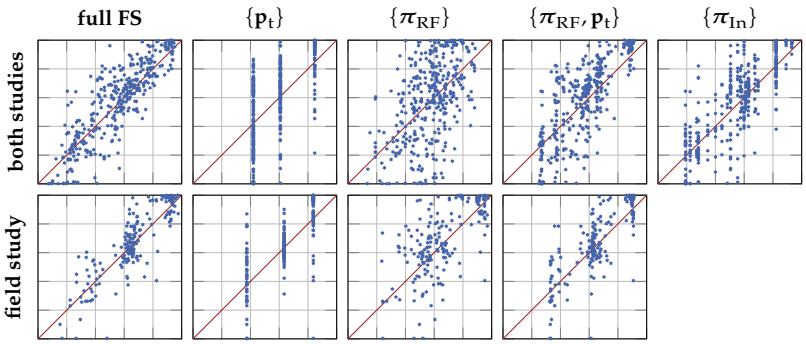


Figure B.2 Scatter plots using RF for prediction and selected feature sets with the velocity drop v_d as the label using the data sets at the T-intersection/p_e combinations that were in both studies (top row, data set $\mathcal{D}_{iv}^{F}$), and that were in the field study only (bottom row, data set $\mathcal{D}_{ov}^{F}$), respectively. The abscissa shows the predicted normalized labels $\hat{\mathbf{y}}_n$, the ordinate shows the normalized true labels $\mathbf{y}_n$.

halves of the tables, data set $\mathcal{D}_{iv}^{F}$) and the data set containing the runs exclusive to the field study (bottom halves, data set $\mathcal{D}_{ov}^{F}$).

The scatter plots for both behavior features, v_m and v_d, in general show the same pattern as those for the commit distance, i.e. the full data set is the one that performs best, using only the turning direction p_t results in characteristic vertical lines. Using π_{RF} as the only feature results in large deviations, adding p_t to it improves the results considerably and using π_{In} on its own results in much better predictions. However, it is apparent that especially the velocity drop v_d feature results in worse predictions (more spread out scatter plots) than the commit distance d_c.

Also, the detailed results of the tables show very similar patterns to those of d_c: The predictions when using the data set of runs exclusive to the field study are generally better than those of the other data set, but some exceptions exist. The complexity scores by direct assignment (π_{In} and E_{In}) are again useful on their own. For these two behavior features the ridge and lasso extensions to the LR algorithm are only superior to the original LR method of complexity score reconstruction in case of the BT score, not the Elo score.

Table B.1 RMSE of behavior prediction models for the minimum velocity v_{m} using different feature sets and regression methods. The data sets $\mathcal{D}_{\mathrm{iv}}^{\mathrm{F}}$ (ref. method: RMSE = 2.015 m s^{-1}) and $\mathcal{D}_{\mathrm{ov}}^{\mathrm{F}}$ (ref. method: RMSE = 1.924 m s^{-1}), respectively, are used.

	feature set	LR	RF	TB	feature set	LR	RF	TB
T-int./p_{e} in field and video study ($\mathcal{D}_{\mathrm{iv}}^{\mathrm{F}}$)	full FS	1.113	1.033	1.008	$\{p_{\mathrm{t}}, p_{\mathrm{e}}\}$	1.510	1.342	1.321
	red. FS	1.191	1.044	1.052	$\{V_{\mathrm{f}}, p_{\mathrm{t}}, p_{\mathrm{e}}\}$	1.341	1.160	1.132
	$\{p_{\mathrm{t}}\}$	1.553	1.547	1.547	AE 1D	1.781	1.405	1.376
	$\{V_{\mathrm{f}}, p_{\mathrm{e}}\}$	1.541	1.385	1.390	AE 2D	1.540	1.217	1.231
	$\{V_{\mathrm{f}}, p_{\mathrm{t}}\}$	1.391	1.306	1.304	AE 3D	1.410	1.182	1.206
	$\{\pi_{\mathrm{In}}\}$	1.861	1.305	1.306	$\{E_{\mathrm{In}}\}$	1.927	1.305	1.306
	$\{\pi_{\mathrm{LR}}\}$	1.736	1.808	1.747	$\{E_{\mathrm{LR}}\}$	1.786	1.783	1.715
	$\{\pi_{\mathrm{LR,r}}\}$	1.758	1.818	1.750	$\{E_{\mathrm{LR,r}}\}$	1.809	1.887	1.798
	$\{\pi_{\mathrm{LR,l}}\}$	1.815	1.826	1.761	$\{E_{\mathrm{LR,l}}\}$	1.783	1.787	1.721
	$\{\pi_{\mathrm{RF}}\}$	1.766	1.827	1.779	$\{E_{\mathrm{RF}}\}$	1.808	1.850	1.780
	$\{\pi_{\mathrm{In}}, p_{\mathrm{t}}\}$	1.391	1.266	1.213	$\{E_{\mathrm{In}}, p_{\mathrm{t}}\}$	1.343	1.266	1.213
	$\{\pi_{\mathrm{LR}}, p_{\mathrm{t}}\}$	1.311	1.282	1.321	$\{E_{\mathrm{LR}}, p_{\mathrm{t}}\}$	1.288	1.269	1.283
	$\{\pi_{\mathrm{LR,r}}, p_{\mathrm{t}}\}$	1.292	1.288	1.289	$\{E_{\mathrm{LR,r}}, p_{\mathrm{t}}\}$	1.275	1.282	1.292
	$\{\pi_{\mathrm{LR,l}}, p_{\mathrm{t}}\}$	1.342	1.272	1.267	$\{E_{\mathrm{LR,l}}, p_{\mathrm{t}}\}$	1.286	1.269	1.282
	$\{\pi_{\mathrm{RF}}, p_{\mathrm{t}}\}$	1.318	1.295	1.291	$\{E_{\mathrm{RF}}, p_{\mathrm{t}}\}$	1.283	1.289	1.307
T-int./p_{e} only in field study ($\mathcal{D}_{\mathrm{ov}}^{\mathrm{F}}$)	full FS	1.027	1.010	0.994	$\{p_{\mathrm{t}}, p_{\mathrm{e}}\}$	1.404	1.270	1.257
	red. FS	1.195	1.013	1.017	$\{V_{\mathrm{f}}, p_{\mathrm{t}}, p_{\mathrm{e}}\}$	1.278	1.126	1.102
	$\{p_{\mathrm{t}}\}$	1.402	1.402	1.402	AE 1D	1.656	1.399	1.389
	$\{V_{\mathrm{f}}, p_{\mathrm{e}}\}$	1.372	1.185	1.202	AE 2D	1.449	1.178	1.198
	$\{V_{\mathrm{f}}, p_{\mathrm{t}}\}$	1.280	1.231	1.264	AE 3D	1.329	1.155	1.216
	$\{\pi_{\mathrm{LR}}\}$	1.484	1.534	1.517	$\{E_{\mathrm{LR}}\}$	1.434	1.480	1.464
	$\{\pi_{\mathrm{LR,r}}\}$	1.468	1.447	1.424	$\{E_{\mathrm{LR,r}}\}$	1.451	1.397	1.379
	$\{\pi_{\mathrm{LR,l}}\}$	1.499	1.464	1.450	$\{E_{\mathrm{LR,l}}\}$	1.438	1.465	1.452
	$\{\pi_{\mathrm{RF}}\}$	1.579	1.509	1.491	$\{E_{\mathrm{RF}}\}$	1.574	1.600	1.599
	$\{\pi_{\mathrm{LR}}, p_{\mathrm{t}}\}$	1.291	1.205	1.252	$\{E_{\mathrm{LR}}, p_{\mathrm{t}}\}$	1.303	1.224	1.285
	$\{\pi_{\mathrm{LR,r}}, p_{\mathrm{t}}\}$	1.307	1.196	1.260	$\{E_{\mathrm{LR,r}}, p_{\mathrm{t}}\}$	1.312	1.180	1.176
	$\{\pi_{\mathrm{LR,l}}, p_{\mathrm{t}}\}$	1.314	1.207	1.236	$\{E_{\mathrm{LR,l}}, p_{\mathrm{t}}\}$	1.302	1.215	1.255
	$\{\pi_{\mathrm{RF}}, p_{\mathrm{t}}\}$	1.305	1.242	1.279	$\{E_{\mathrm{RF}}, p_{\mathrm{t}}\}$	1.331	1.275	1.346

Table B.2 RMSE of behavior prediction models for the velocity drop v_d using different feature sets and regression methods. The data set $\mathcal{D}_{iv}^{F}$ (ref. method: RMSE = 0.280) and $\mathcal{D}_{ov}^{F}$ (ref. method: RMSE = 0.261), respectively, are used.

	feature set	LR	RF	TB	feature set	LR	RF	TB
T-int./p_e in field and video study ($\mathcal{D}_{iv}^{F}$)	full FS	0.166	0.155	0.155	$\{p_t, p_e\}$	0.213	0.195	0.193
	red. FS	0.175	0.157	0.160	$\{V_f, p_t, p_e\}$	0.193	0.175	0.174
	$\{p_t\}$	0.222	0.222	0.222	AE 1D	0.249	0.203	0.197
	$\{V_f, p_e\}$	0.215	0.197	0.199	AE 2D	0.219	0.182	0.184
	$\{V_f, p_t\}$	0.202	0.193	0.193	AE 3D	0.202	0.176	0.180
	$\{\pi_{In}\}$	0.258	0.189	0.189	$\{E_{In}\}$	0.268	0.189	0.189
	$\{\pi_{LR}\}$	0.241	0.251	0.242	$\{E_{LR}\}$	0.248	0.247	0.239
	$\{\pi_{LR,r}\}$	0.245	0.250	0.240	$\{E_{LR,r}\}$	0.253	0.260	0.248
	$\{\pi_{LR,l}\}$	0.252	0.255	0.245	$\{E_{LR,l}\}$	0.248	0.248	0.239
	$\{\pi_{RF}\}$	0.249	0.256	0.248	$\{E_{RF}\}$	0.254	0.256	0.246
	$\{\pi_{LR}, p_t\}$	0.199	0.186	0.178	$\{E_{LR}, p_t\}$	0.194	0.186	0.178
	$\{\pi_{LR}, p_t\}$	0.187	0.188	0.191	$\{E_{LR}, p_t\}$	0.185	0.185	0.184
	$\{\pi_{LR,r}, p_t\}$	0.186	0.188	0.190	$\{E_{LR,r}, p_t\}$	0.186	0.188	0.190
	$\{\pi_{LR,l}, p_t\}$	0.193	0.187	0.187	$\{E_{LR,l}, p_t\}$	0.185	0.185	0.186
	$\{\pi_{RF}, p_t\}$	0.193	0.191	0.191	$\{E_{RF}, p_t\}$	0.188	0.188	0.191
T-int./p_e only in field study ($\mathcal{D}_{ov}^{F}$)	full FS	0.139	0.141	0.138	$\{p_t, p_e\}$	0.183	0.172	0.171
	red. FS	0.153	0.137	0.137	$\{V_f, p_t, p_e\}$	0.162	0.152	0.156
	$\{p_t\}$	0.184	0.184	0.184	AE 1D	0.222	0.189	0.188
	$\{V_f, p_e\}$	0.170	0.157	0.164	AE 2D	0.188	0.164	0.167
	$\{V_f, p_t\}$	0.165	0.157	0.162	AE 3D	0.173	0.159	0.167
	$\{\pi_{LR}\}$	0.192	0.205	0.201	$\{E_{LR}\}$	0.183	0.187	0.184
	$\{\pi_{LR,r}\}$	0.189	0.196	0.193	$\{E_{LR,r}\}$	0.188	0.189	0.187
	$\{\pi_{LR,l}\}$	0.193	0.196	0.194	$\{E_{LR,l}\}$	0.185	0.189	0.187
	$\{\pi_{RF}\}$	0.210	0.199	0.196	$\{E_{RF}\}$	0.208	0.202	0.201
	$\{\pi_{LR}, p_t\}$	0.165	0.163	0.174	$\{E_{LR}, p_t\}$	0.166	0.157	0.164
	$\{\pi_{LR,r}, p_t\}$	0.167	0.160	0.169	$\{E_{LR,r}, p_t\}$	0.169	0.159	0.163
	$\{\pi_{LR,l}, p_t\}$	0.168	0.159	0.164	$\{E_{LR,l}, p_t\}$	0.166	0.157	0.167
	$\{\pi_{RF}, p_t\}$	0.170	0.165	0.169	$\{E_{RF}, p_t\}$	0.173	0.166	0.171

C Examples of Decision-Making

In chapter 4 simulations using the decision-making algorithm introduced there are performed. In addition to the simulation runs presented there, additional ones are shown here. These reveal characteristic properties of the simulation framework and the algorithm's responses to it, namely:

- the Y-V ignores the A-V's right of way (figure C.1);
- interaction between three vehicles without a deadlock (figure C.2);
- the A-V (correctly) drives before its Y-V (figure C.3);
- a complex interaction with two deadlocks (figure C.4);
- the A-V follows its L-V (figure C.5);
- the P-V yields despite having the right of way (figure C.6);
- the A-V passes the intersection without interaction (figure C.7).

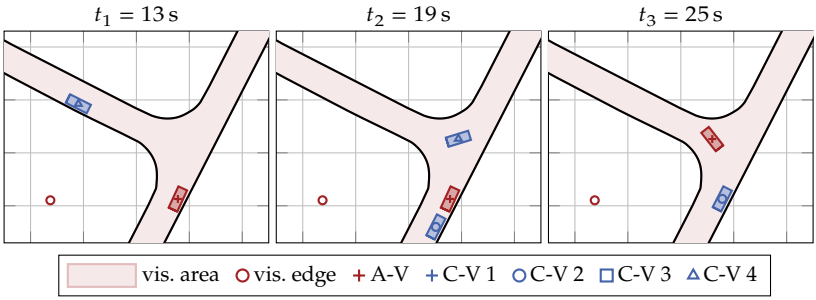


Figure C.1 Simulation run at intersection 5: The A-V has priority over C-V 4 and arrives before it, but C-V 4 does not slow down sufficiently for the A-V's algorithm to drive first. It stops to ensure safety and C-V 4 indeed ignores the A-V's priority. After C-V 4 has passed, it is safe for the A-V to pass itself. C-Vs 1 and 3 are irrelevant here as they passed the intersection before; the grid spacing is 10 m and the visibility distance is set to 21 m.

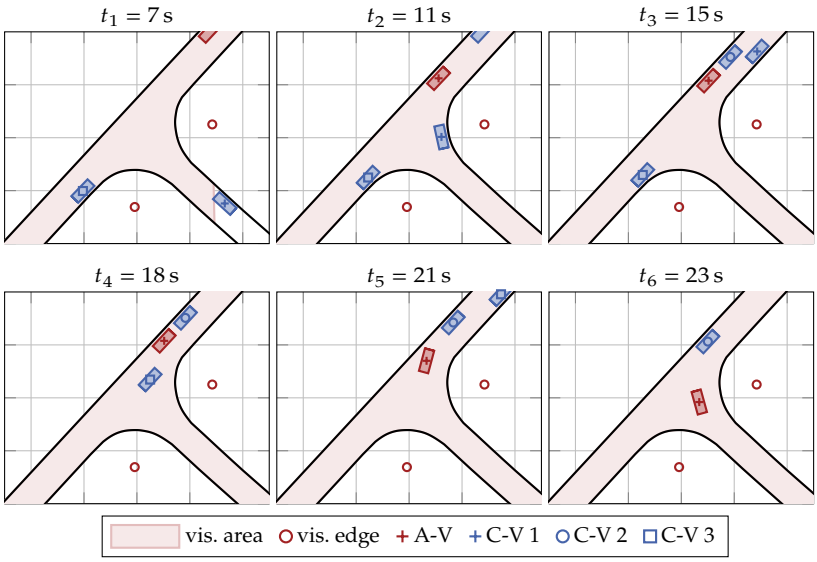


Figure C.2 Simulation run at intersection 3: The A-V, C-Vs 1 and 3 arrive nearly simultaneously, but there is no deadlock as C-V 1 turns right and does not have to yield to any vehicle. It therefore drives first, then C-V 3 followed by the A-V. The grid spacing is 10 m and the visibility distance is set to 7 m.

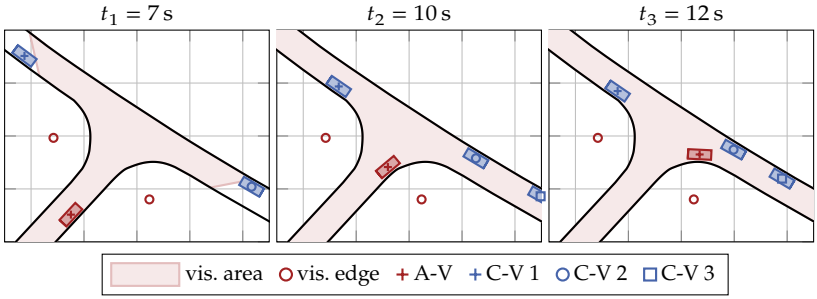


Figure C.3 Simulation run at intersection 4: The A-V only has a Y-V (C-V 1). As it decelerates in time, the A-V passes the intersection before it and remains in its offensive states throughout. The grid spacing is 10 m and the visibility distance is set to 7 m.

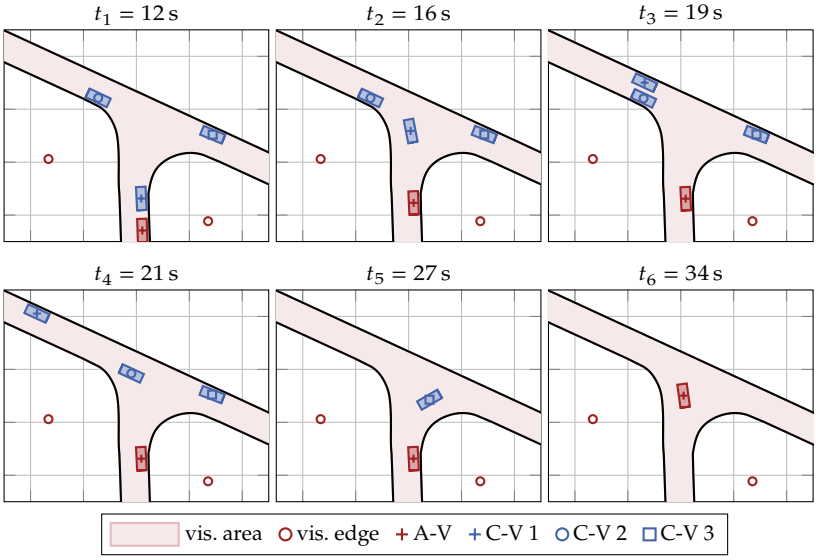


Figure C.4 Simulation run at intersection 9: First a deadlock occurs between C-Vs 1, 2 and 3. It is resolved by C-V 1 driving first. After that, a second deadlock develops with the A-V taking the place of C-V 1. This deadlock is immediately resolved by C-V 2 because this vehicle is set to drive first, which also applies in case of a deadlock. After the resolution of this deadlock, C-V 3, the A-V's P-V, drives, followed by the A-V itself. The grid spacing is 10 m and the visibility distance is set to 14 m.

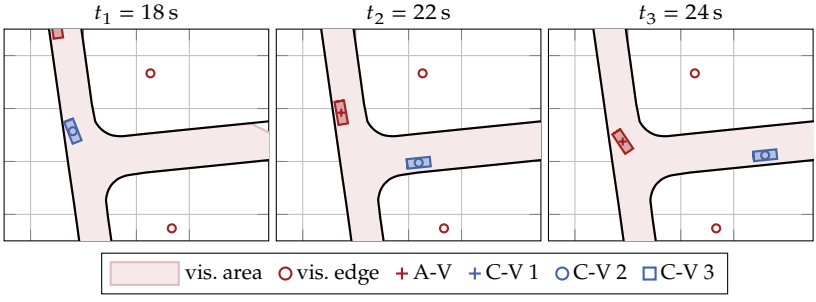


Figure C.5 Simulation run at intersection 7: The A-V follows C-V 2 through the intersection. Before the A-V enters the intersection area it waits until it is able to clear the intersection, i.e. C-V 2 (now both the L-V and the B-V of the A-V) has to have enough space between it and the intersection. C-Vs 1 and 3 passed the intersection before; the grid spacing is 10 m and the visibility distance is set to 14 m.

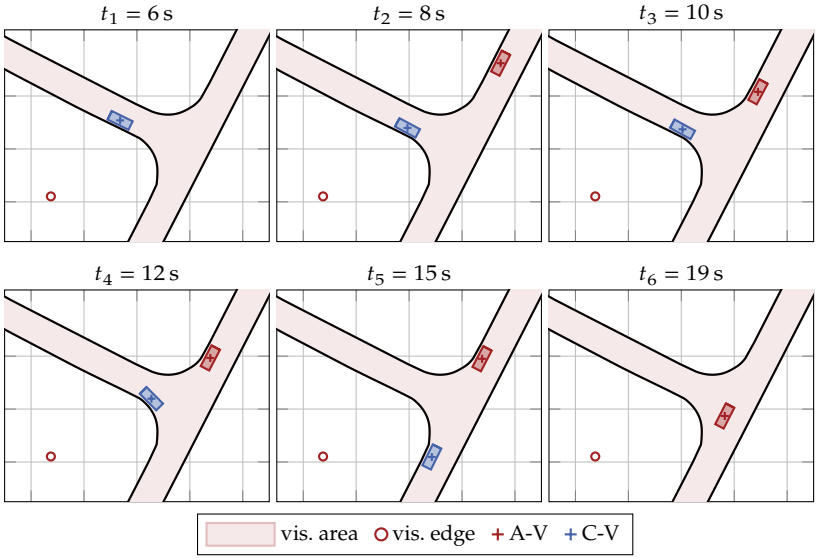


Figure C.6 Simulation run at intersection 5: The C-V has priority over the A-V and arrives before it at the intersection. As the C-V is set to wait nonetheless, it stops at the intersection. The A-V stops as well, as it has to yield. After the waiting period of the C-V is over, it continues and passes the intersection first. The grid spacing is 10 m and the visibility distance is set to 21 m.

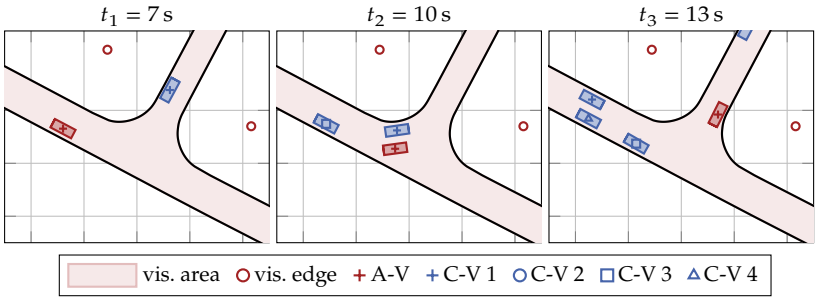


Figure C.7 Simulation run at intersection 6: 10 m before C-V 1 reaches the intersection, the A-V knows that it will turn right and therefore there are no relevant C-Vs at the intersection. The A-V passes it directly. The grid spacing is 10 m and the visibility distance is set to 14 m.

Bibliography

[1] **Statistisches Bundesamt (Destatis)**. *Fehlverhalten der Fahrer bei Unfällen mit Personenschaden*. Available at: https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Verkehrsunfaelle/Tabellen/fehlverhalten-fahrzeugfuehrer.html (In German, accessed: 9 January 2024).

[2] **Statistisches Bundesamt (Destatis)**. *Unfälle und Verunglückte im Straßenverkehr*. Available at: https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Verkehrsunfaelle/Tabellen/unfaelle-verungluecke-.html (In German, accessed: 29 December 2023).

[3] **Statistisches Bundesamt (Destatis)**. *Verkehrsunfälle und Verunglückte im Zeitvergleich (ab 1950)*. Available at: https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Verkehrsunfaelle/Tabellen/liste-strassenverkehrsunfaelle.html#251628 (In German, accessed: 29 Dec. 2023).

[4] **Mourad Ahmane, Abdeljalil Abbas-Turki, Florent Perronnet, Jia Wu, Abdellah El Moudni, Jocelyn Buisson, and Renan Zeo**. *Modeling and controlling an isolated urban intersection based on cooperative vehicles*. In: *Transportation Research Part C: Emerging Technologies* 28 (2013), pp. 44–62.

[5] **Paul CH Albers and Han de Vries**. *Elo-rating as a tool in the sequential estimation of dominance strengths*. In: *Animal Behaviour* (2001), pp. 489–495.

[6] **Javier Alonso, Vicente Milanés, Joshué Pérez, Enrique Onieva, Carlos González, and Teresa de Pedro**. *Autonomous vehicle control systems for safe crossroads*. In: *Transportation Research Part C: Emerging Technologies* 19.6 (2011), pp. 1095–1110.

[7] **Matthias Althoff, Markus Koschi, and Stefanie Manzinger**. *CommonRoad: Composable Benchmarks for Motion Planning on Roads*. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2017, pp. 719–726.

[8] **Shunsuke Aoki and Ragunathan Rajkumar**. *Dynamic Intersections and Self-Driving Vehicles*. In: *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2018.

[9] **Shunsuke Aoki and Ragunathan Rajkumar**. *V2V-based Synchronous Intersection Protocols for Mixed Traffic of Human-Driven and Self-Driving Vehicles*. In: *2019 IEEE 25th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. IEEE, 2019.

[10] **Shunsuke Aoki and Ragunathan Rajkumar**. *Cyber Traffic Light: Safe Cooperation for Autonomous Vehicles at Dynamic Intersections*. In: *IEEE Transactions on Intelligent Transportation Systems* 23.11 (2022), pp. 22519–22534.

[11] **Shunsuke Aoki and Ragunathan (Raj) Rajkumar**. *CSIP: A Synchronous Protocol for Automated Vehicles at Road Intersections*. In: *ACM Transactions on Cyber-Physical Systems* 3.3 (2019), pp. 1–25.

[12] **Shunsuke Aoki and Ragunathan Raj Rajkumar**. *A Configurable Synchronous Intersection Protocol for Self-Driving Vehicles*. In: *2017 IEEE 23rd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. IEEE, 2017.

[13] **Alexandre Armand, David Filliat, and Javier Ibañez-Guzman**. *Ontology-Based Context Awareness for Driving Assistance Systems*. In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014.

[14] **Morteza Asgarzadeh, Santosh Verma, Rania A Mekary, Theodore K Courtney, and David C Christiani**. *The role of intersection and street design on severity of bicycle-motor vehicle crashes*. In: *Injury Prevention* 23.3 (2017), pp. 179–185.

[15] **Winifred D Ashton**. *Gap-Acceptance Problems at a Traffic Intersection*. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 20.2 (1971), pp. 130–138.

[16] **Reza Azimi, Gaurav Bhatia, Ragunathan Raj Rajkumar, and Priyantha Mudalige**. *STIP: Spatio-Temporal Intersection Protocols for Autonomous Vehicles*. In: *2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2014.

[17] **Reza Azimi, Gaurav Bhatia, Ragunathan Rajkumar, and Priyantha Mudalige**. *Ballroom Intersection Protocol: Synchronous Autonomous Driving at Intersections*. In: *2015 IEEE 21st International Conference on Embedded and Real-Time Computing Systems and Applications*. IEEE, 2015.

[18] **Seyed (Reza) Azimi, Gaurav Bhatia, Ragunathan(Raj) Rajkumar, and Priyantha Mudalige**. *Reliable Intersection Protocols Using Vehicular Networks*. In: *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems*. ICCPS '13. ACM, 2013.

[19] **Martin Bach, Daniel Stumper, and Klaus Dietmayer**. *Deep Convolutional Traffic Light Recognition for Automated Driving*. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018.

[20] **Sangjae Bae, Yeojun Kim, Jacopo Guanetti, Francesco Borrelli, and Scott Moura**. *Design and Implementation of Ecological Adaptive Cruise Control for Autonomous Driving with Communication to Traffic Lights*. In: *2019 American Control Conference (ACC)*. IEEE, 2019.

[21] **Rose D Baker and Ian G McHale**. *A dynamic paired comparisons model: Who is the greatest tennis player?* In: *European Journal of Operational Research* 236.2 (2014), pp. 677–684.

[22] **Philipp Bender, Julius Ziegler, and Christoph Stiller**. *Lanelets: Efficient Map Representation for Autonomous Driving*. In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. 2014, pp. 420–425.

[23] **Luis Conde Bento, Ricardo Parafita, Sergio Santos, and Urbano Nunes**. *Intelligent Traffic Management at Intersections: Legacy Mode for Vehicles not Equipped with V2V and V2I Communications*. In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013.

[24] **Maxime Bouton, Alireza Nakhaei, Kikuo Fujimura, and Mykel J. Kochenderfer**. *Scalable Decision Making with Sensor Occlusions for Autonomous Driving*. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 2076–2081.

[25] **Ralph Allan Bradley and Milton E. Terry**. *Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons*. In: *Biometrika* 39.3/4 (1952), pp. 324–345.

[26] **Leo Breiman**. *Random Forests*. In: *Machine Learning* 45.1 (2001), pp. 5–32.

[27] **I.N. Bronstein, K.A. Semendjajew, G. Musiol, and H. Mühlig**. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, 1999.

[28] **Vincent Cantin, Martin Lavallière, Martin Simoneau, and Normand Teasdale**. *Mental workload when driving in a simulator: Effects of age and driving complexity*. In: *Accident Analysis and Prevention* 41.4 (2009), pp. 763–771.

[29] **Manuela Cattelan, Cristiano Varin, and David Firth**. *Dynamic Bradley–Terry modelling of sports tournaments*. In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 62.1 (2012), pp. 135–150.

[30] **Chaoyi Chen, Mengchi Cai, Jiawei Wang, Kai Li, Qing Xu, Jianqiang Wang, and Keqiang Li**. *Cooperation Method of Connected and Automated Vehicles at Unsignalized Intersections: Lane Changing and Arrival Scheduling*. In: *IEEE Transactions on Vehicular Technology* 71.11 (2022), pp. 11351–11366.

[31] **Hoong-Chor Chin and Ser-Tong Quek**. *Measurement of traffic conflicts*. In: *Safety Science* 26.3 (1997), pp. 169–185.

[32] **Andrew P Clark, Kate L Howard, Andy T Woods, Ian S Penton-Voak, and Christof Neumann**. *Why rate when you could compare? Using the "EloChoice" package to assess pairwise comparisons of perceived physical strength*. In: *PloS one* 13.1 (2018), e0190393.

[33] **Elise A. V. Crompvoets, Anton A. Béguin, and Klaas Sijtsma**. *Adaptive Pairwise Comparison for Educational Measurement*. In: *Journal of Educational and Behavioral Statistics* 45.3 (2020), pp. 316–338.

[34] **Alexander G Cunningham, Enric Galceran, Ryan M Eustice, and Edwin Olson**. *MPDM: Multipolicy Decision-Making in Dynamic, Uncertain Environments for Autonomous Driving*. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 1670–1677.

[35] **Herbert Aron David**. *The Method of Paired Comparisons*. Second Edition, Revised. New York: Oxford Univ. Press, 1988.

[36] **Pierre De Beaucorps, Thomas Streubel, Anne Verroust-Blondet, Fawzi Nashashibi, Benazouz Bradai, and Paulo Resende**. *Decision-making for automated vehicles at intersections adapting human-like behavior*. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2017, pp. 212–217.

[37] **Angela Di Febbraro and Davide Giglio**. *On representing signalized urban areas by means of deterministic-timed Petri nets*. In: *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems*. IEEE. 2004, pp. 372–377.

[38] **Philipp Doebler, Mohsen Alavash, and Carsten Giessing**. *Adaptive experiments with a multivariate Elo-type algorithm*. In: *Behavior Research Methods* 47 (2015), pp. 384–394.

[39] **Joshua E Domeyer, John D Lee, Heishiro Toyoda, Bruce Mehler, and Bryan Reimer**. *Interdependence in Vehicle-Pedestrian Encounters and its Implications for Vehicle Automation*. In: *IEEE Transactions on Intelligent Transportation Systems* 23.5 (2022), pp. 4122–4134.

[40] **Chiyu Dong, John M. Dolan, and Bakhtiar Litkouhi**. *Interactive Ramp Merging Planning in Autonomous Driving: Multi-Merging Leading PGM (MML-PGM)*. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017.

[41] **Eric T. Donnell, Richard J. Porter, and Venkataraman N. Shankar**. *A framework for estimating the safety effects of roadway lighting at intersections*. In: *Safety Science* 48.10 (2010), pp. 1436–1444.

[42] **Kurt Dresner and Peter Stone**. *Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism*. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*. AAMAS '04. New York, New York: IEEE Computer Society, 2004, pp. 530–537.

[43] **Kurt Dresner and Peter Stone**. *Multiagent Traffic Management: An Improved Intersection Control Mechanism*. In: *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. AAMAS05. ACM, 2005.

[44] **Kurt Dresner and Peter Stone**. *Sharing the Road: Autonomous Vehicles Meet Human Drivers*. In: *Proceedings of the 20th International Joint Conference on Artifical Intelligence*. Hyderabad, India, 2007, pp. 1263–1268.

[45] **Kurt Dresner and Peter Stone**. *A Multiagent Approach to Autonomous Intersection Management*. In: *Journal of Artificial Intelligence Research* 31 (2008), pp. 591–656.

[46] **Jessica Edquist, Christina M Rudin-Brown, and Michael G Lenné**. *The effects of on-street parking and road environment visual complexity on travel speed and reaction time*. In: *Accident Analysis and Prevention* 45 (2012), pp. 759–765.

[47] **Arpad E Elo**. *The rating of chessplayers, past and present*. New York: Arco Publishing, 1978.

[48] **Fatemeh Enayatollahi, M. A. Amiri Atashgah, Seyed Mohamad-Bagher Malaek, and Parimala Thulasiraman**. *PBN-Based Time-Optimal Terminal Air Traffic Control Using Cellular Automata*. In: *IEEE Transactions on Aerospace and Electronic Systems* 57.3 (2021), pp. 1513–1523.

[49] **Vérane Faure, Régis Lobjois, and Nicolas Benguigui**. *The effects of driving environment complexity and dual tasking on drivers' mental workload and eye blink behavior*. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 40 (2016), pp. 78–90.

[50] **Seyed Alireza Fayazi and Ardalan Vahidi**. *Mixed-Integer Linear Programming for Optimal Scheduling of Autonomous Vehicle Intersection Crossing*. In: *IEEE Transactions on Intelligent Vehicles* 3.3 (2018), pp. 287–299.

[51] **Mathias Franz, Emily McLean, Jenny Tung, Jeanne Altmann, and Susan C Alberts**. *Self-organizing dominance hierarchies in a wild primate population*. In: *Proceedings of the Royal Society B: Biological Sciences* 282.1814 (2015), p. 20151512.

[52] **Enric Galceran, Alexander G Cunningham, Ryan M Eustice, and Edwin Olson**. *Multipolicy Decision-Making for Autonomous Driving via Changepoint-based Behavior Prediction*. In: *Robotics: Science and Systems*. Vol. 1. 2. 2015, p. 6.

[53] **Roberto Gásquez and Vicente Royuela**. *The Determinants of International Football Success: A Panel Data Analysis of the Elo Rating*. In: *Social Science Quarterly* 97.2 (2016), pp. 125–141.

[54] **Thomas Geiger**. *Dieses Auto kann in die Zukunft schauen*. In: *Der Spiegel* (15 July 2021). Available at: https://www.spiegel.de/auto/fahrberichte/s-klasse-mit-drive-pilot-dieses-auto-kann-in-die-zukunft-schauen-a-74c422af-13fe-4c8d-8bcc-04f078becbc9 (In German, accessed: 08 January 2024).

[55] **Mark E Glickman**. *Parameter estimation in large dynamic paired comparison experiments*. In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 48.3 (1999), pp. 377–394.

[56] **Daniel Golson**. *We put our blind faith in Mercedes-Benz's first-of-its-kind autonomous Drive Pilot feature*. In: *The Verge* (27 September 2023). Available at: https://www.theverge.com/2023/9/27/23892154/mercedes-benz-drive-pilot-autonomous-level-3-test (Accessed: 08 January 2024).

[57] **Ian Goodfellow, Yoshua Bengio, and Aaron Courville**. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[58] **Robert Goodspeed**. *Research note: An evaluation of the Elo algorithm for pairwise visual assessment surveys*. In: *Landscape and Urban Planning* 157 (2017), pp. 131–137.

[59] **Jean Grégoire, Silvère Bonnabel, and Arnaud de La Fortelle**. *Priority-based intersection management with kinodynamic constraints*. In: *2014 European Control Conference (ECC)*. IEEE, 2014.

[60] **Wolfgang Gruel and Joseph M. Stanford**. *Assessing the Long-Term Effects of Autonomous Vehicles: a speculative approach*. In: *Transportation Research Procedia* 13 (2016), pp. 18–29.

[61] **Peter A Hancock, Gabriele Wulf, D Thom, and P Fassnacht**. *Driver workload during differing driving maneuvers*. In: *Accident Analysis & Prevention* 22.3 (1990), pp. 281–290.

[62] **Trevor Hastie, Robert Tibshirani, and Jerome Friedman**. *The Elements of Statistical Learning*. Springer New York, 2009.

[63] **Ralf Herbrich, Tom Minka, and Thore Graepel**. *TrueSkill(TM): A Bayesian Skill Rating System*. In: *Advances in Neural Information Processing Systems 20*. MIT Press, 2007, pp. 569–576.

[64] **Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor**. *Real-Time Loop Closure in 2D LIDAR SLAM*. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 1271–1278.

[65] **G. E. Hinton and R. R. Salakhutdinov**. *Reducing the Dimensionality of Data with Neural Networks*. In: *Science* 313.5786 (2006), pp. 504–507.

[66] **Geoffrey Ho, Charles T Scialfa, Jeff K Caird, and Trevor Graw**. *Visual Search for Traffic Signs: The Effects of Clutter, Luminance, and Aging*. In: *Human Factors* 43.2 (2001), pp. 194–207.

[67] **Tim Horberry, Janet Anderson, Michael A Regan, Thomas J Triggs, and John Brown**. *Driver distraction: The effects of concurrent in-vehicle tasks, road environment complexity and age on driving performance*. In: *Accident Analysis and Prevention* 38.1 (2006), pp. 185–191.

[68] **Yi-Sheng Huang, Yi-Shun Weng, and MengChu Zhou**. *Modular Design of Urban Traffic-Light Control Systems Based on Synchronized Timed Petri Nets*. In: *IEEE Transactions on Intelligent Transportation Systems* 15.2 (2013), pp. 530–539.

[69] **Yi-Sheng Huang, Yi-Shun Weng, and MengChu Zhou**. *Design of Traffic Safety Control Systems for Emergency Vehicle Preemption Using Timed Petri Nets*. In: *IEEE Transactions on Intelligent Transportation Systems* 16.4 (2015), pp. 2113–2120.

[70] **Constantin Hubmann, Marvin Becker, Daniel Althoff, David Lenz, and Christoph Stiller**. *Decision Making for Autonomous Driving considering Interaction and Uncertain Prediction of Surrounding Vehicles*. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2017, pp. 1671–1678.

[71] **Michael Hülsen, J. Marius Zöllner, and Christian Weiss**. *Traffic Intersection Situation Description Ontology for Advanced Driver Assistance*. In: *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011.

[72] **Lars Magnus Hvattum and Halvard Arntzen**. *Using ELO ratings for match result prediction in association football*. In: *International Journal of Forecasting* 26.3 (2010), pp. 460–470.

[73] **Jonas Imbsweiler, Renáta Palyafári, Fernando Puente León, and Barbara Deml**. *Untersuchung des Entscheidungsverhaltens in kooperativen Verkehrssituationen am Beispiel einer Engstelle*. In: *at-Automatisierungstechnik* 65.7 (2017), pp. 477–488.

[74] **Georg Jahn, Astrid Oehme, Josef F Krems, and Christhard Gelau**. *Peripheral detection as a workload measure in driving: Effects of traffic complexity and route guidance system use in a driving study*. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 8.3 (2005), pp. 255–275.

[75] **Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani**. *An Introduction to Statistical Learning: with Applications in R*. 2nd ed. 2021. Springer Texts in Statistics. New York, NY: Springer US, 2021.

[76] **E. Jelavic, J. Gonzales, and F. Borrelli**. *Autonomous Drift Parking using a Switched Control Strategy with Onboard Sensors*. In: *IFAC-PapersOnLine* 50.1 (2017), pp. 3714–3719.

[77] **Carl Johnsson, Aliaksei Laureshyn, and Tim De Ceunynck**. *In search of surrogate safety indicators for vulnerable road users: a review of surrogate safety indicators*. In: *Transport Reviews* 38.6 (2018), pp. 765–785.

[78] **Kamal Kant and Steven W. Zucker**. *Toward Efficient Trajectory Planning: The Path-Velocity Decomposition*. In: *The International Journal of Robotics Research* 5.3 (1986), pp. 72–89.

[79] **Mohammad Khayatian, Mohammadreza Mehrabian, and Aviral Shrivastava**. *RIM: Robust Intersection Management for Connected Autonomous Vehicles*. In: *2018 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 2018, pp. 35–44.

[80] **Stefan Klingelschmitt, Florian Damerow, and Julian Eggert**. *Managing the Complexity of Inner-City Scenes: An Efficient Situation Hypotheses Selection Scheme*. In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 1232–1239.

[81] **Maximilian Kloock, Patrick Scheffe, Sascha Marquardt, Janis Maczijewski, Bassam Alrifaee, and Stefan Kowalewski**. *Distributed Model Predictive Intersection Control of Multiple Vehicles*. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 1735–1740.

[82] **Maximilian Kloock, Patrick Scheffe, Janis Maczijewski, Alexandru Kampmann, Armin Mokhtarian, Stefan Kowalewski, and Bassam Alrifaee**. *Cyber-Physical Mobility Lab: An Open-Source Platform For Networked and Autonomous Vehicles*. In: *2021 European Control Conference (ECC)*. IEEE. 2021, pp. 1937–1944.

[83] **Karsten Kreutz and Julian Eggert**. *Analysis of the Generalized Intelligent Driver Model (GIDM) for merging situations*. In: *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2021, pp. 34–41.

[84] **Karsten Kreutz and Julian Eggert**. *Analysis of the Generalized Intelligent Driver Model (GIDM) for Uncontrolled Intersections*. In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE. 2021, pp. 3223–3230.

[85] **Dong-Kyoung Kye, Seong-Woo Kim, and Seung-Woo Seo**. *Decision Making for Automated Driving at Unsignalized Intersection*. In: *2015 15th International Conference on Control, Automation and Systems (ICCAS)*. IEEE. 2015, pp. 522–525.

[86] **Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom**. *PointPillars: Fast Encoders for Object Detection From Point Clouds*. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[87] **Jan Lasek, Zoltán Szlávik, and Sandjai Bhulai**. *The predictive power of ranking systems in association football*. In: *International Journal of Applied Pattern Recognition* 1.1 (2013), pp. 27–46.

[88] **Joyoung Lee and Byungkyu Park**. *Development and Evaluation of a Cooperative Vehicle Intersection Control Algorithm Under the Connected Vehicles Environment*. In: *IEEE Transactions on Intelligent Transportation Systems* 13.1 (2012), pp. 81–90.

[89] **Jesse Levinson *et al.*** *Towards Fully Autonomous Driving: Systems and Algorithms*. In: *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011.

[90] **Jesse Levinson, Jake Askeland, Jennifer Dolson, and Sebastian Thrun**. *Traffic Light Mapping, Localization, and State Detection for Autonomous Vehicles*. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011.

[91] **Guofa Li, Ying Wang, Fangping Zhu, Xiaoxuan Sui, Ning Wang, Xingda Qu, and Paul Green**. *Drivers' visual scanning behavior at signalized and unsignalized intersections: A naturalistic driving study in China*. In: *Journal of Safety Research* 71 (2019), pp. 219–229.

[92] **Guofa Li, Weijian Lai, Xiaoxuan Sui, Xiaohang Li, Xingda Qu, Tingru Zhang, and Yuezhi Li**. *Influence of traffic congestion on driver behavior in post-congestion driving*. In: *Accident Analysis and Prevention* 141 (2020), p. 105508.

[93] **Ziyue Li, Qinghua Zeng, Yuchao Liu, Jianye Liu, and Lin Li**. *An improved traffic lights recognition algorithm for autonomous driving in complex scenarios*. In: *International Journal of Distributed Sensor Networks* 17.5 (2021), p. 155014772110183.

[94] **Martin Liebner, Michael Baumann, Felix Klanner, and Christoph Stiller**. *Driver Intent Inference at Urban Intersections using the Intelligent Driver Model*. In: *2012 IEEE Intelligent Vehicles Symposium*. 2012, pp. 1162–1167.

[95] **Peiqun Lin, Jiahui Liu, Peter J. Jin, and Bin Ran**. *Autonomous Vehicle-Intersection Coordination Method in a Connected Vehicle Environment*. In: *IEEE Intelligent Transportation Systems Magazine* 9.4 (2017), pp. 37–47.

[96] **Yi-Ting Lin, Hsiang Hsu, Shang-Chien Lin, Chung-Wei Lin, Iris Hui-Ru Jiang, and Changliu Liu**. *Graph-Based Modeling, Scheduling, and Verification for Intersection Management of Intelligent Vehicles*. In: *ACM Transactions on Embedded Computing Systems* 18.5s (2019), pp. 1–21.

[97] **Xiao Lin, Jiucai Zhang, Jin Shang, Yi Wang, Hongkai Yu, and Xiaoli Zhang**. *Decision Making through Occluded Intersections for Autonomous Driving*. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE. 2019, pp. 2449–2455.

[98] **Teng Liu, Bing Huang, Zejian Deng, Hong Wang, Xiaolin Tang, Xiao Wang, and Dongpu Cao**. *Heuristics-oriented overtaking decision making for autonomous vehicles using reinforcement learning*. In: *IET Electrical Systems in Transportation* 10.4 (2020), pp. 417–424.

[99] **Wei Liu, Seong-Woo Kim, Scott Pendleton, and Marcelo H Ang**. *Situation-aware Decision Making for Autonomous Driving on Urban Road using Online POMDP*. In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2015, pp. 1126–1133.

[100] **Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner**. *Microscopic Traffic Simulation using SUMO*. In: *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018, pp. 2575–2582.

[101] **Jiliang Luo, Yi-Sheng Huang, and Yi-Shun Weng**. *Design of Variable Traffic Light Control Systems for Preventing Two-Way Grid Network Traffic Jams Using Timed Petri Nets*. In: *IEEE Transactions on Intelligent Transportation Systems* 21.7 (2020), pp. 3117–3127.

[102] **Laleh Makarem and Denis Gillet**. *Model predictive coordination of autonomous vehicles crossing intersections*. In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE. 2013, pp. 1799–1804.

[103] **Udara E Manawadu, Takahiro Kawano, Shingo Murata, Mitsuhiro Kamezaki, Junya Muramatsu, and Shigeki Sugano**. *Multiclass Classification of Driver Perceived Workload Using Long Short-Term*

Memory based Recurrent Neural Network. In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2018, pp. 2009–2014.

[104] **Dan Marinescu, Jan Curn, Marco Slot, Melanie Bouroche, and Vinny Cahill**. *An Active Approach to Guaranteed Arrival Times Based on Traffic Shaping*. In: *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2010, pp. 1711–1717.

[105] **Dan Marinescu, Jan Curn, Melanie Bouroche, and Vinny Cahill**. *On-ramp traffic merging using cooperative intelligent vehicles: A slot-based approach*. In: *2012 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012, pp. 900–906.

[106] **J.N.S. Matthews and K.P. Morris**. *An Application of Bradley-Terry-type Models to the Measurement of Pain*. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 44.2 (1995), pp. 243–255.

[107] **Patrick McGowen and Laura Stanley**. *Alternative Methodology for Determining Gap Acceptance for Two-Way Stop-Controlled Intersections*. In: *Journal of Transportation Engineering* 138.5 (2012), pp. 495–501.

[108] **Ian McHale and Alex Morton**. *A Bradley-Terry type model for forecasting tennis match results*. In: *International Journal of Forecasting* 27.2 (2011), pp. 619–630.

[109] **Joshua E. Menke and Tony R. Martinez**. *A Bradley-Terry artificial neural network model for individual ratings in group competitions*. In: *Neural Computing and Applications* 17 (2008), pp. 175–186.

[110] **Cade Metz and Neal E. Boudette**. *Inside Tesla as Elon Musk Pushed an Unflinching Vision for Self-Driving Cars*. In: *The New York Times* (6 December 2021, updated 22 June 2023). Available at: https://www.nytimes.com/2021/12/06/technology/tesla-autopilot-elon-musk.html (Accessed: 08 January 2024).

[111] **Tripp Mickle, Yiwen Lu, and Mike Isaac**. *'This Experience May Feel Futuristic': Three Rides in Waymo Robot Taxis*. In: *The New York Times* (21 August 2023). Available at: https://www.nytimes.com/2023/08/21/technology/waymo-driverless-cars-san-francisco.html (Accessed: 05 January 2024).

[112] **Michiel M. Minderhoud and Piet H.L. Bovy**. *Extended time-to-collision measures for road traffic safety assessment*. In: *Accident Analysis and Prevention* 33.1 (2001), pp. 89–97.

[113] **Michael Montemerlo *et al.*** *Junior: The Stanford entry in the Urban Challenge*. In: *Journal of Field Robotics* 25.9 (2008), pp. 569–597.

[114] **Imanol Mugarza and Juan Carlos Mugarza**. *A coloured Petri net-and D* Lite-based traffic controller for Automated Guided Vehicles*. In: *Electronics* 10.18 (2021), p. 2235.

[115] **Tadao Murata**. *Petri Nets: Properties, Analysis and Applications*. In: *Proceedings of the IEEE* 77.4 (1989), pp. 541–580.

[116] **Kai Nagel and Michael Schreckenberg**. *A cellular automaton model for freeway traffic*. In: *Journal de Physique I* 2.12 (1992), pp. 2221–2229.

[117] **Maximilian Naumann, Fabian Poggenhans, Martin Lauer, and Christoph Stiller**. *Coincar-sim: An Open-Source Simulation Framework for Cooperatively Interacting Automobiles*. In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2018, pp. 1–6.

[118] **Christof Neumann, Julie Duboscq, Constance Dubuc, Andri Ginting, Ade Maulana Irwan, Muhammad Agil, Anja Widdig, and Antje Engelhardt**. *Assessing dominance hierarchies: validation and advantages of progressive evaluation with Elo-rating*. In: *Animal Behaviour* 82 (2011), pp. 911–921.

[119] **Julia Nilsson, Mattias Brännström, Erik Coelingh, and Jonas Fredriksson**. *Lane Change Maneuvers for Automated Vehicles*. In: *IEEE Transactions on Intelligent Transportation Systems* 18.5 (2017), pp. 1087–1096.

[120] **Samyeul Noh**. *Decision-Making Framework for Autonomous Driving at Road Intersections Safeguarding Against Collision Overly Conservative Behavior and Violation Vehicles*. In: *IEEE Transactions on Industrial Electronics* 66.4 (2018), pp. 3275–3286.

[121] **Shigehiro Oishi, Jungwon Hahn, Ulrich Schimmack, Phanikiran Radhakrishan, Vivian Dzokoto, and Stephen Ahadi**. *The measurement of values across cultures: A pairwise comparison approach*. In: *Journal of Research in Personality* 39.2 (2005), pp. 299–305.

[122] **Oscar Oviedo-Trespalacios, Md Mazharul Haque, Mark King, and Simon Washington**. *Effects of road infrastructure and traffic complexity in speed adaptation behaviour of distracted drivers*. In: *Accident Analysis and Prevention* 101 (2017), pp. 67–77.

[123] **Christopher JD Patten, Albert Kircher, Joakim Östlund, Lena Nilsson, and Ola Svenson**. *Driver experience and cognitive workload in different traffic environments*. In: *Accident Analysis and Prevention* 38.5 (2006), pp. 887–894.

[124] **Radek Pelánek**. *Applications of the Elo rating system in adaptive educational systems*. In: *Computers & Education* 98 (2016), pp. 169–179.

[125] **Andrew S. Phelps, David M. Naeger, Jesse L. Courtier, Jack W. Lambert, Peter A. Marcovici, Javier E. Villanueva-Meyer, and John D. MacKenzie**. *Pairwise Comparison Versus Likert Scale for Biomedical Image Assessment.* In: *AJR. American Journal of Roentgenology* 204.1 (2015), pp. 8–14.

[126] **Derek J. Phillips, Tim A. Wheeler, and Mykel J. Kochenderfer**. *Generalizable Intention Prediction of Human Drivers at Intersections*. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2017, pp. 1665–1670.

[127] **Fabian Poggenhans, Jan-Hendrik Pauls, Johannes Janosovits, Stefan Orf, Maximilian Naumann, Florian Kuhnt, and Matthias Mayr**. *Lanelet2: A high-definition map framework for the future of automated driving*. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2018, pp. 1672–1679.

[128] **Ulrich Pörschmann, Fritz Trillmich, Birte Mueller, and Jochen B. W. Wolf**. *Male reproductive success and its behavioural correlates in a polygynous mammal, the Galápagos sea lion (Zalophus wollebaeki)*. In: *Molecular Ecology* 19.12 (2010), pp. 2574–2586.

[129] **Mahmoud Pourmehrab, Lily Elefteriadou, Sanjay Ranka, and Marilo Martin-Gasulla**. *Optimizing Signalized Intersections Performance Under Conventional and Automated Vehicles Traffic*. In: *IEEE Transactions on Intelligent Transportation Systems* 21.7 (2020), pp. 2864–2873.

[130] **Fernando Puente León and Uwe Kiencke**. *Ereignisdiskrete Systeme: Modellierung und Steuerung verteilter Systeme*. 3. Auflage. München: Oldenbourg, 2013.

[131] **Tim Puphal, Malte Probst, Yiyang Li, Yosuke Sakamoto, and Julian Eggert**. *Optimization of Velocity Ramps with Survival Analysis for Intersection Merge-Ins*. In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2018, pp. 1704–1710.

[132] **Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas**. *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 652–660.

[133] **Liang Qi, MengChu Zhou, and WenJing Luan**. *Modeling and Control of Urban Road Intersections with Incidents via Timed Petri Nets*. In: *2015 IEEE 12th International Conference on Networking, Sensing and Control*. IEEE. 2015, pp. 185–190.

[134] **Bo Qian, Haibo Zhou, Feng Lyu, Jinglin Li, Ting Ma, and Fen Hou**. *Toward Collision-Free and Efficient Coordination for Automated Vehicles at Unsignalized Intersection*. In: *IEEE Internet of Things Journal* 6.6 (2019), pp. 10408–10420.

[135] **Xiangjun Qian, Jean Gregoire, Fabien Moutarde, and Arnaud De La Fortelle**. *Priority-based coordination of autonomous and legacy vehicles at intersection*. In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 1166–1171.

[136] **Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, *et al.*** *ROS: an open-source Robot Operating System*. In: *ICRA workshop on open source software*. Vol. 3. 3.2. Kobe, Japan. 2009.

[137] **Yalda Rahmati and Alireza Talebpour**. *Towards a Collaborative Connected, Automated Driving Environment: A Game Theory Based Decision Framework for Unprotected Left Turn Maneuvers*. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2017, pp. 1316–1321.

[138] **Ralf Regele**. *Using Ontology-Based Traffic Models for More Efficient Decision Making of Autonomous Vehicles*. In: *Fourth International Conference on Autonomic and Autonomous Systems (ICAS'08)*. IEEE, 2008.

[139] **Gabriel Rodrigues de Campos, Paolo Falcone, Robert Hult, Henk Wymeersch, and Jonas Sjoberg**. *Traffic Coordination at Road Intersections: Autonomous Decision-Making Algorithms Using Model-Based Heuristics*. In: *IEEE Intelligent Transportation Systems Magazine* 9.1 (2017), pp. 8–21.

[140] **Patrick Scheffe, Janis Maczijewski, Maximilian Kloock, Alexandru Kampmann, Andreas Derks, Stefan Kowalewski, and Bassam Alrifaee**. *Networked and Autonomous Model-scale Vehicles for Experiments in Research and Education*. In: *IFAC-PapersOnLine* 53.2 (2020), pp. 17332–17337.

[141] **Michael Schreckenberg, Lutz Neubert, and Joachim Wahle**. *Simulation of traffic in large road networks*. In: *Future Generation Computer Systems* 17.5 (2001), pp. 649–657.

[142] **Horst Schulze and Ingo Koßmann**. *The role of safety research in road safety management*. In: *Safety Science* 48.9 (2010), pp. 1160–1166.

[143] **Volkan Sezer, Tirthankar Bandyopadhyay, Daniela Rus, Emilio Frazzoli, and David Hsu**. *Towards Autonomous Navigation of Unsignalized Intersections under Uncertainty of Human Driver Intent*. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 3578–3585.

[144] **Keqi Shu, Huilong Yu, Xingxin Chen, Shen Li, Long Chen, Qi Wang, Li Li, and Dongpu Cao**. *Autonomous Driving at Intersections: A Behavior-Oriented Critical-Turning-Point Approach for Decision Making*. In: *IEEE/ASME Transactions on Mechatronics* (2021).

[145] **Jeffrey T. Steedle and Steve Ferrara**. *Evaluating Comparative Judgment as an Approach to Essay Scoring*. In: *Applied Measurement in Education* 29.3 (2016), pp. 211–223.

[146] **Thomas Streubel and Karl Heinz Hoffmann**. *Prediction of Driver Intended Path at Intersections*. In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE. 2014, pp. 134–139.

[147] **Carolin Strobl, Florian Wickelmaier, and Achim Zeileis**. *Accounting for Individual Differences in Bradley-Terry Models by Means of Recursive Partitioning*. In: *Journal of Educational and Behavioral Statistics* 36.2 (2011), pp. 135–153.

[148] **Evona Teh, Samantha Jamson, Oliver Carsten, and Hamish Jamson**. *Temporal fluctuations in driving demand: The effect of traffic complexity on subjective measures of workload and driving performance*. In: *Transportation Research Part F: traffic psychology and behaviour* 22 (2014), pp. 207–217.

[149] **Trisha Thadani, Faiz Siddiqui, Rachel Lerman, and Jeremy B. Merrill**. *Tesla drivers run Autopilot where it's not intended — with deadly consequences*. In: *The Washington Post* (10 December 2023). Available at: https://www.washingtonpost.com/technology/2023/12/10/tesla-autopilot-crash/ (Accessed: 08 January 2024).

[150] **Sebastian Thrun *et al.*** *Stanley: The Robot that Won the DARPA Grand Challenge*. In: *Journal of Field Robotics* 23.9 (2006), pp. 661–692.

[151] **Tomer Toledo**. *Driving Behaviour: Models and Challenges*. In: *Transport Reviews* 27.1 (2007), pp. 65–84.

[152] **Martin Treiber, Ansgar Hennecke, and Dirk Helbing**. *Congested traffic states in empirical observations and microscopic simulations*. In: *Physical Review E* 62.2 (2000), p. 1805.

[153] **Martin Treiber, Arne Kesting, and Dirk Helbing**. *Delays, inaccuracies and anticipation in microscopic traffic models*. In: *Physica A: Statistical Mechanics and its Applications* 360.1 (2006), pp. 71–88.

[154] **Martin Treiber, Arne Kesting, and Dirk Helbing**. *Influence of Reaction Times and Anticipation on Stability of Vehicular Traffic Flow*. In: *Transportation Research Record: Journal of the Transportation Research Board* 1999.1 (2007), pp. 23–29.

[155] **Stefan Vacek, Tobias Gindele, J. Marius Zollner, and Rudiger Dillmann**. *Using case-based reasoning for autonomous vehicle guidance*. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 4271–4276.

[156] **Jianqiang Wang, Jiaxiang Yan, and Lingxi Li**. *Microscopic Modeling of a Signalized Traffic Intersection Using Timed Petri Nets*. In: *IEEE Transactions on Intelligent Transportation Systems* 17.2 (2015), pp. 305–312.

[157] **Pin Wang and Ching-Yao Chan**. *Formulation of Deep Reinforcement Learning Architecture Toward Autonomous Driving for On-Ramp Merge*. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017.

[158] **Erik Ward and John Folkesson**. *Multi-classification of Driver Intentions in Yielding Scenarios*. In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE. 2015, pp. 678–685.

[159] **Junhua Wei, Anlin Wang, and Nianci Du**. *Study of Self-Organizing Control of Traffic Signals in an Urban Network Based on Cellular Automata*. In: *IEEE Transactions on Vehicular Technology* 54.2 (2005), pp. 744–748.

[160] **Junqing Wei, John M. Dolan, and Bakhtiar Litkouhi**. *Autonomous Vehicle Social Behavior for Highway Entrance Ramp Management*. In: *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013.

[161] **Julia Werneke and Mark Vollrath**. *What does the driver look at? The influence of intersection characteristics on attention allocation and driving behavior*. In: *Accident Analysis and Prevention* 45 (2012), pp. 610–619.

[162] **Martin J. Whiting, Devi M. Stuart-Fox, David O'Connor, David Firth, Nigel C. Bennett, and Simon P. Blomberg**. *Ultraviolet signals ultra-aggression in a lizard*. In: *Animal Behaviour* 72.2 (2006), pp. 353–363.

[163] **Jasper S Wijnands, Haifeng Zhao, Kerry A Nice, Jason Thompson, Katherine Scully, Jingqiu Guo, and Mark Stevenson**. *Identifying safe intersection design through unsupervised feature extraction from satellite imagery*. In: *Computer-Aided Civil and Infrastructure Engineering* (2020), pp. 346–361.

[164] **Leighton Vaughan Williams, Chunping Liu, Lerato Dixon, and Hannah Gerrard**. *How well do Elo-based ratings predict professional tennis matches?* In: *Journal of Quantitative Analysis in Sports* 17.2 (2021), pp. 91–105.

[165] **Canming Xia, Mali Xing, and Shenghuang He**. *Interactive Planning for Autonomous Driving in Intersection Scenarios Without Traffic Signs*. In: *IEEE Transactions on Intelligent Transportation Systems* 23.12 (2022), pp. 24818–24828.

[166] **Fei Yan, Mahjoub Dridi, and Abdellah El Moudni**. *Autonomous Vehicle Sequencing Algorithm at Isolated Intersections*. In: *2009 12th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2009, pp. 621–626.

[167] **Shengchao Yan, Jingwei Zhang, Daniel Buscher, and Wolfram Burgard**. *Efficiency and Equity are Both Essential: A Generalized Traffic Signal Controller with Deep Reinforcement Learning*. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5526–5533.

[168] **Shengchao Yan, Tim Welschehold, Daniel Büscher, and Wolfram Burgard**. *Courteous Behavior of Automated Vehicles at Unsignalized Intersections Via Reinforcement Learning*. In: *IEEE Robotics and Automation Letters* 7.1 (2022), pp. 191–198.

[169] **Xiao Zhang, Wenda Xu, Chiyu Dong, and John M Dolan**. *Efficient L-Shape Fitting for Vehicle Detection Using Laser Scanners*. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2017, pp. 54–59.

[170] **Lihua Zhao, Ryutaro Ichise, Tatsuya Yoshikawa, Takeshi Naito, Toshiaki Kakinami, and Yutaka Sasaki**. *Ontology-based Decision Making on Uncontrolled Intersections and Narrow Roads*. In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 83–88.

[171] **Yanan Zhao, Zhiwei Li, Li Gao, and Jian Xiong**. *Road-Feature-Based Multiparameter Road Complexity Calculation Model of Off-Road Environment*. In: *Mathematical Problems in Engineering* 2018 (2018), pp. 1–12.

[172] **Bowen Zheng, Chung-Wei Lin, Hengyi Liang, Shinichi Shiraishi, Wenchao Li, and Qi Zhu**. *Delay-Aware Design, Analysis and Verification of Intelligent Intersection Management*. In: *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 2017.

[173] **Bowen Zheng, Chung-Wei Lin, Shinichi Shiraishi, and Qi Zhu**. *Design and Analysis of Delay-Tolerant Intelligent Intersection Management*. In: *ACM Transactions on Cyber-Physical Systems* 4.1 (2019), pp. 1–27.

[174] **Mofan Zhou, Xiaobo Qu, and Sheng Jin**. *On the Impact of Cooperative Autonomous Vehicles in Improving Freeway Merging: A Modified Intelligent Driver Model-Based Approach*. In: *IEEE Transactions on Intelligent Transportation Systems* 18.6 (2016), pp. 1422–1428.

[175] **Mofan Zhou, Yang Yu, and Xiaobo Qu**. *Development of an Efficient Driving Strategy for Connected and Automated Vehicles at Signalized Intersections: A Reinforcement Learning Approach*. In: *IEEE Transactions on Intelligent Transportation Systems* 21.1 (2020), pp. 433–443.

[176] **Zhi-Hua Zhou**. *Machine Learning*. 1st ed. Springer eBook Collection. Singapore: Springer Singapore, 2021.

[177] **Julius Ziegler *et al.*** *Making Bertha Drive—An Autonomous Journey on a Historic Route*. In: *IEEE Intelligent Transportation Systems Magazine* 6.2 (2014), pp. 8–20.

[178] **Cesar Zucco Jr, Mariana Batista, and Timothy J. Power**. *Measuring portfolio salience using the Bradley-Terry model: An illustration with data from Brazil*. In: *Research and Politics* 6.1 (2019), pp. 1–8.

[179] **Alex Zyner, Stewart Worrall, James Ward, and Eduardo Nebot**. *Long Short Term Memory for Driver Intent Prediction*. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2017, pp. 1484–1489.

List of publications

[180] **Jonas Imbsweiler, Katrin Linstedt, Renáta Palyafári, Hannes Weinreuter, Fernando Puente León, and Barbara Deml**. *Quasi-experimentelle Untersuchung des Blickverhaltens und der Fahrparameter von Autofahrern in Engstellen*. In: *Zeitschrift für Arbeitswissenschaft* 71.4 (2017), pp. 242–251.

[181] **Jonas Imbsweiler, Maureen Ruesch, Hannes Weinreuter, Fernando Puente León, and Barbara Deml**. *Cooperation behaviour of road users in t-intersections during deadlock situations*. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 58 (2018), pp. 665–677.

[182] **Jonas Imbsweiler, Maureen Ruesch, Tobias Heine, Katrin Linstedt, Hannes Weinreuter, Fernando Puente León, and Barbara Deml**. *Die Rolle der expliziten Kommunikation im Straßenverkehr*. In: *Arbeit(s).Wissen.Schaf(f)t Grundlage für Management & Kompetenzentwicklung : 64. GfA-Frühjahrskongress, 21.-23. Februar 2018, Frankfurt am Main, FOM Hochschulzentrum Frankfurt*. GfA-Press, 2018.

[183] **Thomas Nürnberg, Hannes Weinreuter, and Fernando Puente León**. *Tiefenmessung mithilfe einer Kamera mit programmierbarer Apertur: Depth measurement using a programmable aperture camera*. In: *tm - Technisches Messen* 84.s1 (2017), pp. 52–59.

[184] **Nadine-Rebecca Strelau, Hannes Weinreuter, Michael Heizmann, and Barbara Deml**. *Patt-Situationen im Stadtverkehr - Der Einfluss von Verkehrsaufkommen auf die wahrgenommene Schwierigkeit und das Kooperationsverhalten an T-Kreuzungen*. In: *Arbeit HUMAINE gestalten. 67. Kongress der Gesellschaft für Arbeitswissenschaft. Hrsg.: Gesellschaft für Arbeitswissenschaft, GfA-Press.* 67. GfA-Frühjahrskongress Arbeit HUMAINE gestalten. GfA 2021 (Online, Mar. 3–5, 2021). 2021.

[185] **Nadine-Rebecca Strelau, Hannes Weinreuter, Michael Heizmann, and Barbara Deml**. *Der Einfluss von Fußgängern auf die wahrgenommene Übersichtlichkeit und das Kooperationsverhalten von Autofahrern an T-Kreuzungen*. 68. GfA Frühjahrskongress: Tech-

nologie und Bildung in hybriden Arbeitswelten (2022), Magdeburg, Deutschland, 2.–4. März 2022. 2022.

[186] **Nadine-Rebecca Strelau, Jonas Imbsweiler, Gloria Pöhler, Hannes Weinreuter, Michael Heizmann, and Barbara Deml**. *Cooperation Behavior of Drivers at Inner City Deadlock-Situations*. In: *Cooperatively Interacting Vehicles: Methods and Effects of Automated Cooperation in Traffic*. Ed. by **Christoph Stiller, Matthias Althoff, Christoph Burger, Barbara Deml, Lutz Eckstein, and Frank Flemisch**. Cham: Springer International Publishing, 2024, pp. 545–564.

[187] **Hannes Weinreuter, Jonas Imbsweiler, Nadine-Rebecca Strelau, Barbara Deml, and Fernando Puente León**. *Prediction of human driver intentions at a narrow passage in inner city traffic / Intentionsprädiktion menschlicher Fahrer an einer Engstelle im innerstädtischen Straßenverkehr*. In: *tm - Technisches Messen* 86.s1 (2019), pp. 127–131.

[188] **Hannes Weinreuter, Nadine-Rebecca Strelau, Barbara Deml, and Michael Heizmann**. *Intention prediction of car drivers at inner city junctions*. In: *2020 International Symposium on Electronics and Telecommunications (ISETC)*. IEEE, 2020.

[189] **Hannes Weinreuter, Balázs Szigeti, Nadine-Rebecca Strelau, Barbara Deml, and Michael Heizmann**. *Decision making at unsignalized inner city intersections using discrete events systems*. In: *tm - Technisches Messen* 89.2 (2022), pp. 134–146.

[190] **Hannes Weinreuter, Nadine-Rebecca Strelau, Kevin Qiu, Yancheng Jiang, Barbara Deml, and Michael Heizmann**. *Intersection Complexity and Its Influence on Human Drivers*. In: *IEEE Access* 10 (2022), pp. 74059–74070.

[191] **Hannes Weinreuter, Nadine-Rebecca Strelau, Barbara Deml, and Michael Heizmann**. *Verhaltensentscheidungen für das automatische Fahren an innerstädtischen T-Kreuzungen mittels ereignisdiskreter Systeme*. In: *at - Automatisierungstechnik* 71.4 (2023), pp. 258–269.

[192] **Hannes Weinreuter, Nadine-Rebecca Strelau, Barbara Deml, and Michael Heizmann**. *Analysis and Simulation of Driving Behavior at Inner City Intersections*. In: *Cooperatively Interacting Vehicles: Methods and Effects of Automated Cooperation in Traffic*. Ed. by **Christoph Stiller, Matthias Althoff, Christoph Burger, Barbara Deml, Lutz Eckstein, and Frank Flemisch**. Cham: Springer International Publishing, 2024, pp. 89–110.

List of supervised theses

[193] **Yuqing Cui**. *Investigation and Prediction of Driving Behavior in Cooperative Traffic Situations*. Master's thesis. Karlsruhe Institute of Technology, 2022.

[194] **Yancheng Jiang**. *Automatic Acquisition of Visual Complexity Based on Lidar Data*. Master's thesis. Karlsruhe Institute of Technology, 2020.

[195] **Yichen Jin**. *Implementation and Verification of Sensor Fusion Algorithms for Precise Localization in the Context of Automated Driving*. Master's thesis. Karlsruhe Institute of Technology, 2018.

[196] **Alexander Liebrenz**. *Entwicklung einer Intentionsprädiktion für menschliche Fahrer an gleichrangigen Engstellen anhand der Untersuchung von Trajektorieneigenschaften*. Master's thesis. Karlsruhe Institute of Technology, 2018.

[197] **Long Lin**. *Ereignisdiskrete Modellierung einer gleichrangigen Engstelle im innerstädtischen Straßenverkehr*. Bachelor's thesis. Karlsruhe Institute of Technology, 2018.

[198] **Wanyu Liu**. *Implementation of cooperative situations in a driving simulator in the context of automated driving*. Bachelor's thesis. Karlsruhe Institute of Technology, 2018.

[199] **Kevin Qiu**. *Analysis of driver behavior and complexity at inner city intersections*. Master's thesis. Karlsruhe Institute of Technology, 2021.

[200] **Marian Seeger**. *Objektidentifizierung und -lokalisierung anhand von Lidaraufzeichnungen im Kontext des automatischen Fahrens*. Bachelor's thesis. Karlsruhe Institute of Technology, 2018.

[201] **Sarangan Selvalingam**. *Intentionsprädiktion an innerstädtischen Kreuzungen mit neuronalen Netzen*. Master's thesis. Karlsruhe Institute of Technology, 2021.

[202] **Jiaxin Shen**. *Behavior Modeling at Inner City Intersections for Automatic Driving*. Master's thesis. Karlsruhe Institute of Technology, 2021.

[203] **Balázs Szigeti**. *Entscheidungsfindung bei verschiedenen Szenarien an innerstädtischen Kreuzungen für das automatische Fahren*. Master's thesis. Karlsruhe Institute of Technology, 2021.

[204] **Yuntao Wang**. *Design of behavioral decision making in a driving simulator in the context of automated driving*. Master's thesis. Karlsruhe Institute of Technology, 2019.

[205] **Zusheng Wang**. *Discrete event modeling for decision making using intention prediction at a narrow passage scenario*. Master's thesis. Karlsruhe Institute of Technology, 2019.

[206] **Laura Wegmann**. *Entwicklung einer Verhaltensprädiktion an innerstädtischen Kreuzungen*. Master's thesis. Karlsruhe Institute of Technology, 2019.

[207] **Chenghao Yin**. *Design of a decision-making algorithm at a narrow passage in the context of autonomous driving*. Master's thesis. Karlsruhe Institute of Technology, 2020.

[208] **Yu Zhou**. *Intersection complexity score based on a video study of pairwise comparisons*. Master's thesis. Karlsruhe Institute of Technology, 2022.