



Christoph Taschwer, BSc

Implementation of a Highway Chauffeur in a Virtual Test Environment

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Mechanical Engineering

submitted to

Graz University of Technology

Supervisors

Assoc.Prof. Dipl.-Ing. Dr.techn. Arno Eichberger

Dipl.-Ing. Demin Nalic, BSc

Institute of Automotive Engineering

Member of [FSI]

Dipl.-Ing. (FH) Dr.techn. Stefan Bernsteiner

MAGNA STEYR Fahrzeugtechnik AG & CoKG

Graz, December 2018

Restricted access until <12 2020>

Acknowledgement

My sincere thanks go to Assoc.Prof. Dipl.-Ing. Dr.techn. Arno Eichberger and Dipl.-Ing. Demin Nalic, BSc from the Institute of Automotive Engineering at Technical University of Graz for giving me the opportunity to work on this interesting topic as well as the good working atmosphere.

Moreover, I would like to thank Dipl.-Ing. (FH) Dr.techn. Stefan Bernsteiner from MAGNA STEYR Fahrzeugtechnik AG & CoKG who made this project possible.

I am very grateful that my parents gave me the opportunity to study and for supporting me during this time. I also thank my sister for always being there for me.

Finally, I would like to thank all my friends and my team mates of Graz Gladiators Lacrosse with whom I spent a great time in Graz.

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

.....
Date

.....
Signature

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

.....
Datum

.....
Unterschrift

Abstract

Nowadays, the topic of *Advanced Driver Assistance Systems* (ADAS) as well as the next step *Automated Driving Systems* (ADS) are omnipresent and a lot of development effort is put into this field of automotive technology. Starting with the usage of simple ADAS of low automation levels, intended only to support the driver by providing additional information and warnings, more and more sophisticated systems are developed and integrated into modern vehicles. The aim of utilizing these new technologies is not only to improve safety, but also to increase the comfort for the driver by overtaking step by step different parts of the driving task, starting from lateral and longitudinal guidance up to monitoring the environment and making smart decisions according to specific traffic situations.

However, before such assistance systems can be integrated into vehicles, testing and validating is required. To reduce the amount of expensive and time consuming real drive tests, virtual methods are used. Utilizing such virtual techniques help to assess the product at an early stage of the development phase.

In this thesis a model for a specific ADAS called Highway Chauffeur is implemented in Mathworks[®]' MATLAB/Simulink and tested with IPG's simulation tool CarMaker[®]. The Highway Chauffeur is intended to support the driver on highways by overtaking the lateral and longitudinal guidance of the vehicle. The main part of this thesis deals with the decision making process (when to perform a lane change manoeuvre via the *Lane Change Assist* (LCA)), the target selection for *Adaptive Cruise Control* (ACC) and the trajectory generation needed for a lane change manoeuvre.

Kurzfassung

Das Thema *Advanced Driver Assistance Systems* (ADAS) wie auch die nächste Stufe der *Automated Driving Systems* (ADS) ist heutzutage allgegenwärtig und die Automobilindustrie betreibt einen enormen Entwicklungsaufwand in diesen Bereichen. Angefangen mit einfachen ADAS niedriger Automatisierungsstufen, die nur dafür gedacht sind, den Fahrer in gewissen Situationen mit zusätzlichen Informationen oder Warnungen zu unterstützen, werden immer fortschrittlichere Systeme entwickelt und in moderne Fahrzeuge integriert. Durch die Nutzung solcher Systeme soll aber nicht nur die Sicherheit erhöht sondern auch der Fahrkomfort deutlich verbessert werden. Dabei werden Teile der Fahraufgabe, angefangen von der Längs- und Querverführung bis hin zum Überwachen der Umgebung und selbständigen Treffen von klugen Entscheidungen zufolge bestimmter Verkehrssituationen, vom ADAS übernommen.

Bevor aber solche Systeme in ein Fahrzeug integriert werden können ist umfangreiches Testen und Validieren notwendig. Um die Menge an teuren und zeitintensiven realen Tests zu minimieren, werden virtuelle Testmethoden angewendet. Solche virtuelle Methoden bieten außerdem die Möglichkeit das Produkt frühzeitig im Entwicklungsprozess zu bewerten.

In dieser Arbeit soll ein Modell für ein spezielles ADAS mit der Bezeichnung Highway Chauffeur im Softwarepaket MATLAB/Simulink von MathWorks[®] aufgebaut werden und anschließend im Softwarepaket CarMaker[®] von IPG getestet werden. Der Highway Chauffeur soll den Fahrer auf Autobahnen unterstützen, indem er die longitudinale und laterale Führung des Fahrzeugs übernimmt. Der Fokus dieser Arbeit liegt im Bereich des Entscheidungsprozesses (wann soll ein Fahrstreifenwechsel mittels *Lane Change Assist* (LCA) durchgeführt werden), der Targetauswahl für das *Adaptive Cruise Control* (ACC) sowie der Generierung für den Fahrstreifenwechsel notwendigen Trajektorie.

Contents

Acknowledgement	i
AFFIDAVIT	iii
Abstract	v
Kurzfassung	vii
Contents	xi
Abbreviations	xiii
Symbols	xv
1. Introduction	1
1.1. Motivation	1
1.2. State of the art	3
1.2.1. Three level classification according to Rasmussen and Donges . . .	3
1.2.2. History of driver assistance systems and autonomous driving . . .	4
1.2.3. Levels of automation	4
1.2.4. Different types of Advanced Driver Assistance Systems	6
1.2.4.1. Adaptive Cruise Control (ACC)	6
1.2.4.2. Automatic Emergency Braking (AEB)	7
1.2.4.3. Lane Keeping Assist (LKA)	7
1.2.4.4. Active Lane Change Assist (LCA)	7
1.2.4.5. Highway Chauffeur (HWC)	8
1.2.5. Sensors for environmental recognition	10
1.2.5.1. Performance influencing factors and perturbations	11
1.2.6. Development process on virtual level	13
1.2.6.1. V-Model and In-the-Loop methods	14
1.3. Simulation with IPG CarMaker®	17
1.4. Requirements for the HWC model of this thesis	19
2. Methodology	21
2.1. Virtual road	21
2.1.1. Modification of road data	22

2.2.	Refloating a consisting HWC model	24
2.2.1.	Derivation of the Time To Collision (TTC)	25
2.2.1.1.	Time To Collision without acceleration	26
2.2.1.2.	Time To Collision with acceleration	26
2.2.2.	Lane change manoeuvre	29
2.3.	Creation of a new HWC model and integration of different sensor models	32
2.3.1.	Phenomenological RADAR sensor model	33
2.3.2.	Ideal Sensors	33
2.3.2.1.	Object Sensor	33
2.3.3.	High Fidelity Sensors	35
2.3.3.1.	RADAR Sensor	35
2.4.	Determination of the target position	35
2.4.1.	Method for assigning the detected targets to the according lanes .	36
2.5.	HWC model in Simulink	39
2.5.1.	Subsystems and MATLAB function blocks concerning the front of the ego	40
2.5.1.1.	Data RADAR sensor (front)	40
2.5.1.2.	Assignment target to lane (front)	40
2.5.1.3.	Equation of motion/TTC (front)	42
2.5.1.4.	Smallest values of targets (front)	42
2.5.2.	Subsystems and functions concerning the rear of the ego	43
2.5.2.1.	Data RADAR sensor (rear)	44
2.5.2.2.	Assignment target to lane (rear)	44
2.5.2.3.	Equation of motion/TTC (rear)	45
2.5.2.4.	TTC of nearest targets (rear)	45
2.5.3.	Decision making LC left	46
2.5.3.1.	Smallest TTC on the same lane	49
2.5.3.2.	Smallest TTC on the first lane to the left of the ego . . .	50
2.5.3.3.	Smallest positive TTC on the second lane to the left of the ego	51
2.5.4.	Decision making LC right	51
2.5.5.	Hold state	52
2.5.6.	Time during LC manoeuvre to the left	52
2.5.7.	Lateral offset calculation LC to the left	52
2.5.8.	Time during LC manoeuvre to the right	52
2.5.9.	Lateral offset calculation LC to the right	52
2.6.	Simulation and Virtual Testing of the HWC Model	53
3.	Results and Discussion	55
3.1.	Investigation of the polynomial approach	55
3.2.	Investigation of the approach with the smallest TTC in front	59
4.	Summary	63

List of Figures	I
List of Tables	III
Bibliography	V
A. Appendix	VII
A.1. HWC model in Simulink	VII

Abbreviations

ABS	Anti-lock Braking System
ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance System
ADS	Automated Driving System
AEB	Automatic Emergency Braking System
BASt	Bundesanstalt für Straßenwesen
CIT	CarMaker [®] Interface Toolbox
ESP	Electronic Stability Program
FCW	Forward Collision Warning
FOV	Field of View
GUI	Graphical User Interface
HiFi	High Fidelity
HiL	Hardware in the Loop
HMI	Human Machine Interface
HWC	Highway Chauffeur
LDW	Lane Departure Warning
LIDAR	Light Detection and Ranging
LKA	Lane Keeping Assist
MiL	Model in the Loop
RADAR	Radio Detection and Ranging
SAE	Society of Automotive Engineers
SiL	Software in the Loop
SNR	Signal to Noise Ratio
TJA	Traffic Jam Assist
TTC	Time to Collision
UAQ	User Accessible Quantity
ViL	Vehicle in the Loop
VVE	Virtual Vehicle Environment
XML	Extensible Markup Language

Symbols

Variables and Parameters

a	Acceleration
G	Point on line $P_1 - P_2$
h	maximal lateral displacement
ln	Distance between line $P_1 - P_2$ and R (measured perpendicular)
m	Mass
$marker$	Reference number for lateral position
P	Line point
r	Radius
R	Point detected by the RADAR sensor
s	Distance
t	Time
TTC	Time To Collision
v	Velocity
x	Position in x direction
\dot{x}	Velocity in x direction
\ddot{x}	Acceleration in x direction
y	Position in y direction
\dot{y}	Velocity in y direction
\ddot{y}	Acceleration in y direction
α	Angle between line $P_1 - P_2$ and x_s
Δ	Difference
κ	Curvature
ω	Angular velocity

Vectors

\mathbf{a}	Vector of relative acceleration
\mathbf{s}	Vector of relative distance
\mathbf{TTC}	Vector of TTC
\mathbf{v}	Vector of relative velocity

Indices

0	Initial value
1	First adjacent
1l	First lane to the left of the ego
1r	First lane to the right of the ego
2	Second adjacent
2l	Second lane to the left of the ego
2r	Second lane to the right of the ego
<i>f</i>	Front
<i>G</i>	Point on line $P_1 - P_2$
<i>ll</i>	Lower limit
<i>ul</i>	Upper limit
<i>m</i>	Manoeuvre
<i>min</i>	Smallest value
<i>n</i>	Place holder for $2l, 1l, s, 1r, 2r$
<i>r</i>	Rear
<i>rel</i>	Relative
<i>s</i>	Same lane as the ego
<i>S</i>	Sensor
<i>x</i>	x direction
<i>y</i>	y direction

1. Introduction

1.1. Motivation

Besides alternative powertrain technologies, the development and usage of *Advanced Driver Assistance Systems* (ADAS) is becoming more and more important. The main idea behind the integration of ADAS into vehicles is not only to increase safety, but also to make the whole driving process more comfortable for the driver and to reduce the energy consumption by optimizing the driving strategie. As mentioned before, safety plays a huge role and a very important aspect is the reduction of accidents. In figure 1.1 a chronological sequence of road fatalities in Austria is shown (1983 - 2017). This decreasing trend is caused by the step-by-step introduction of different safety systems such as seatbelt, headrest, safety glass, airbag or driver assistance systems like *Anti-lock Braking System* (ABS) etc. over the years and should be reduced further over the next years.

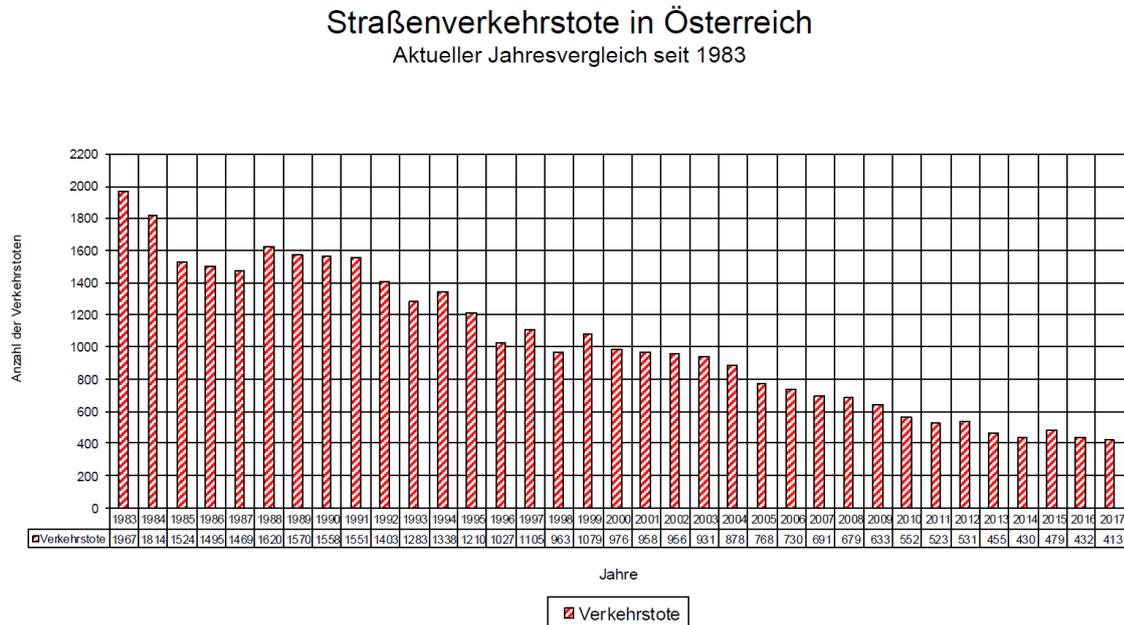


Figure 1.1.: Annual road fatalities in Austria since 1983, [5]

Most of the accidents are caused by human misconduct. Figure 1.2 shows the main causes of accidents. To reduce the influence of the human factor, different tasks, originally performed by the driver, are more and more transferred to driver assistance systems. This can help in critical situations to avoid an accident or to mitigate the effects of an accident for example via an *Automatic Emergency Braking System* (AEB). Other systems like *Adaptive Cruise Control* (ACC) can help to maintain a safe distance between the vehicles. Combined with a *Lane Keeping Assist* (LKA) the comfort for the driver is also increased.

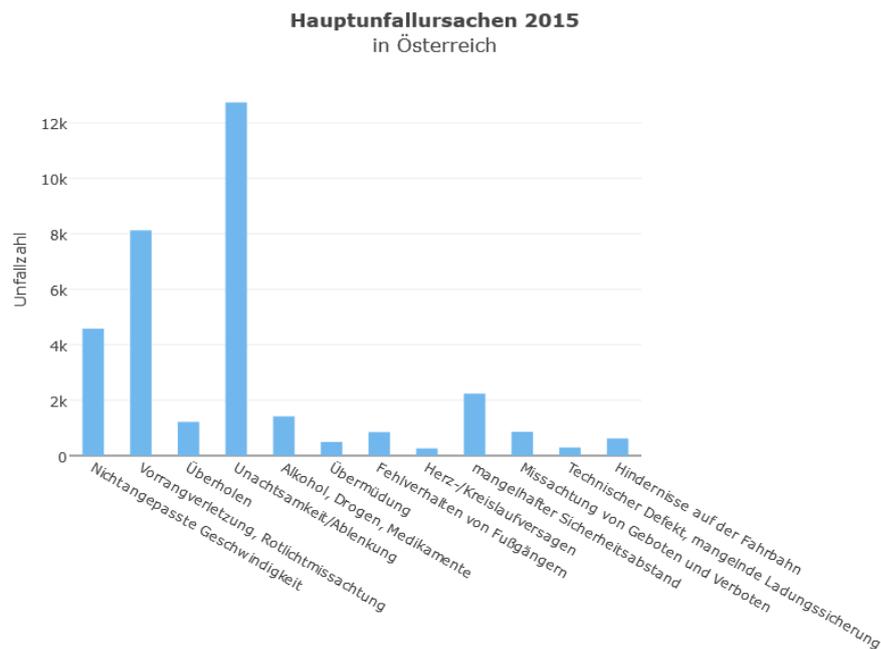


Figure 1.2.: Main causes of traffic accidents with personal injury in 2015, [14]

The development of ADAS, which is the preliminary stage before *Automated Driving Systems* (ADS), requests extensive testing and validation to guarantee error-free functionality before these systems can be introduced. Nowadays, the test and validation procedure is more and more done in an virtual environment to reduce the amount of expensive and time-consuming real drive tests with prototypes.

Further advantages of a virtual approach are:

- reproducibility of the simulated test scenario
- possibility to investigate critical situations, without putting the driver into danger
- no perturbation
- derivation of additional quantities
- easy study of alternatives

It has to be kept in mind, that in a simulation physical effects are approximated and described via mathematical formulas and in a lot of cases simplifications have to be made. For this reason experimental verification is also necessary [1].

1.2. State of the art

This section deals with the state of the art and gives an overview of the topic *Advanced Driver Assistance Systems* and the the virtual development process.

1.2.1. Three level classification according to Rasmussen and Donges

Winner describes in [10] the model for the human behaviour according to Rasmussen and the classification of the driving task according to Donges as well as the interaction between these two models.

Rasmussen generally structured the human behaviour in three layers:

1. Knowledge-based behaviour: based on existing knowledge, it is searched for an appropriate action and a decision is made
2. Rule-based behaviour: according to subjective experience, an action is made
3. Skill-based behaviour: an action is made by reflex

The three components driver, vehicle and environment are forming a system in which the driver is interacting with the environment via the vehicle. According to Donges, the activity of the driver or the driving task can be subdivided into three layers:

1. Navigation: searching for an appropriate route
2. Guidance: generating the desired values of the course and the velocity
3. Stabilization: controlling the vehicle with steering, brake and powertrain

Figure 1.3 shows the interactions between the three layers of human behaviour according to Rasmussen and the three layers of the driving task according to Donges.

According to Bernsteiner [4] and Reif [18] the *Advanced Driver Assitance System* is influencing the interactions between driver, vehicle and environment and supports the motorist with the driving task by

- informing and warning the driver about dangerous or critical situations via visual, acoustic and/or haptic signals
- active intervening (initiated automatically or by the driver) in the vehicle guidance such as *Lane Keeping Assist* (LKA) or stabilization like *Electronic Stability Programm* (ESP)

ADAS can also be categorized into safety and comfort systems.

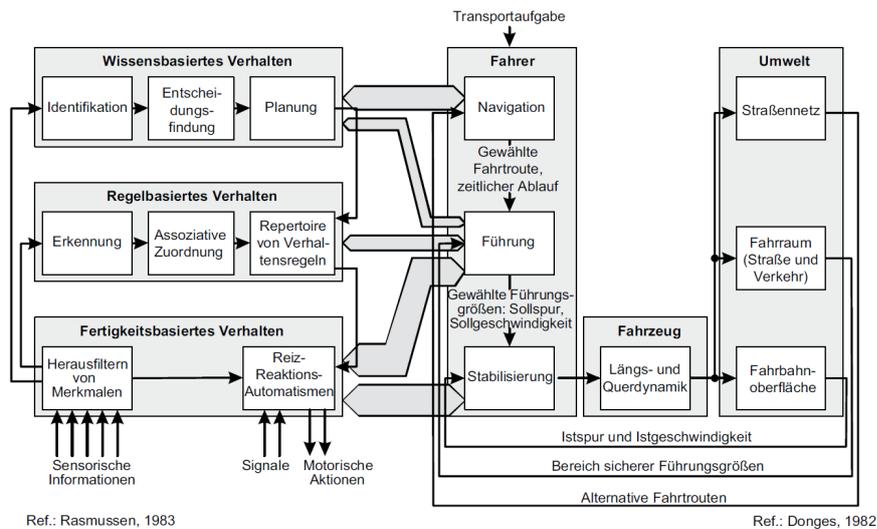


Figure 1.3.: Three layers of human behaviour according to Rasmussen and three layers of the driving task according to Donges, [10]

1.2.2. History of driver assistance systems and autonomous driving

The support of the driver has been one of the main development goals since the beginning of the automotive industry. Over time a lot of systems, which support the driver, were developed and integrated into vehicles. In 1912 Cadillac replaced the starter crank with an electric starter. Six years later, a prototype of a cornering light was integrated in a Cadillac. In 1932 Chrysler introduced the brake booster to reduce the necessary pedal force of the driver. The Chrysler Imperial was the first vehicle with power steering as standard equipment (1951) [18]. In the 1950s General Motors had the idea of automated driving on highways. Due to too low performance of computers and image processing equipment, a combination of infrastructure and vehicle was necessary (magnets were placed in the road which were detected by sensors). In the 1970s and 1980s Japanese scientists worked with image processing devices and cameras to detect objects and lanes. During the 1990s experiments with longitudinal and lateral guided vehicles on highways and highway-like roads were performed in America and Europe. All these tests were supervised by a human driver [10].

1.2.3. Levels of automation

Driver assistance systems can be categorized according to the level of automation. Using the classification scheme of the *Society of Automotive Engineers* (SAE) is one possibility to categorize driver assistance systems according to their level of automation [21]. The following six items give an overview of this classification.

1. Level 0 (No Automation):

The driver has to perform all dynamic driving tasks (longitudinal and lateral guidance) and to monitor the environment at all time. Examples are *Lane Departure Warning* (LDW) or *Forward Collison Warning* (FCW)

2. Level 1 (Driver Assistance):

The driver has to perform either longitudinal or lateral guidance and to monitor the environment at all time, while the other guidance is performed by an assistance system between certain limits. Such systems are only available for specific driving modes. Examples are ACC or LKA.

3. Level 2 (Partial Automation):

A driver assistance system takes care of the longitudinal and the lateral guidance at the same time within certain limits. The driver however has to monitor the environment and has to be able to take control of the vehicle at any time. Human take-over is monitored with hands-on detection. These systems are also only available for specific driving modes. An example for such a system is the *Autopilot*.

4. Level 3 (Conditional Automation):

A driver assistance system takes care of the longitudinal and the lateral guidance. The system also monitors the environment and the driver is allowed to do secondary tasks with eyes-off. In critical situations, the driver has to take control of the vehicle again. Reliable take over is yet an unresolved issue. Level 3 systems are also intended for specific driving modes. An example for such a system is the *Traffic Jam Assist* (TJA).

5. Level 4 (High Automation):

A level 4 system is responsible for both longitudinal and lateral guidance as well as for monitoring the environment, like a level 3 system. The difference however is, that if the driver does not respond to a request to intervene, the driver assistance system is able to handle the situation with suitable fail-safe operation. Similar to level 3 is, that level 4 systems are only available for some driving modes. An advanced *Highway Chauffeur* that can handle complex situations can be named as an example.

6. Level 5 (Full Automation):

Compared to driver assistance systems of the before mentioned level (High Automation), systems of level 5 are working at all driving modes.

Figure 1.4 illustrates the levels of automation according to SAE and also compares these levels to the classification according to *Bundesanstalt für Straßenwesen* (BAST-Prjoektgruppe).

1. Introduction

Level	Name	Narrative definition	Execution of steering and acceleration/deceleration	Monitoring of driving environment	Fallback performance of dynamic driving task	System capability (driving modes)	BASt level	NHTSA level
Human driver monitors the driving environment								
0	No Automation	the full-time performance by the <i>human driver</i> of all aspects of the <i>dynamic driving task</i> , even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a	Driver only	0
1	Driver Assistance	the <i>driving mode</i> -specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the <i>human driver</i> perform all remaining aspects of the <i>dynamic driving task</i>	Human driver and system	Human driver	Human driver	Some driving modes	Assisted	1
2	Partial Automation	the <i>driving mode</i> -specific execution by one or more driver assistance systems of both steering and acceleration/deceleration using information about the driving environment and with the expectation that the <i>human driver</i> perform all remaining aspects of the <i>dynamic driving task</i>	System	Human driver	Human driver	Some driving modes	Partially automated	2
Automated driving system ("system") monitors the driving environment								
3	Conditional Automation	the <i>driving mode</i> -specific performance by an <i>automated driving system</i> of all aspects of the <i>dynamic driving task</i> with the expectation that the <i>human driver</i> will respond appropriately to a <i>request to intervene</i>	System	System	Human driver	Some driving modes	Highly automated	3
4	High Automation	the <i>driving mode</i> -specific performance by an <i>automated driving system</i> of all aspects of the <i>dynamic driving task</i> , even if a <i>human driver</i> does not respond appropriately to a <i>request to intervene</i>	System	System	System	Some driving modes	Fully automated	3/4
5	Full Automation	the full-time performance by an <i>automated driving system</i> of all aspects of the <i>dynamic driving task</i> under all roadway and environmental conditions that can be managed by a <i>human driver</i>	System	System	System	All driving modes	*	3/4

Figure 1.4.: SAE levels of automation, [21]

1.2.4. Different types of Advanced Driver Assistance Systems

In the following sections 1.2.4.1 - 1.2.4.5 ADAS of different areas (lateral/longitudinal guidance) and levels of automation are described.

1.2.4.1. Adaptive Cruise Control (ACC)

Adaptive Cruise Control (ACC) is an ADAS that operates in longitudinal direction and is an extension of the conventional *Cruise Control*. This system uses forward-looking sensors with for example *Radio Detection and Ranging* (RADAR) technology or cameras to monitor the area in front of the so called ego vehicle and adjusts the speed due to the traffic conditions by intervening into the powertrain and/or the braking system. ACC reacts to vehicles, driving on the same lane in front, as well as to vehicles that are cutting into the lane of the ego. If the vehicle in front, the so called target vehicle, is slower as the ego, the ACC system slows the ego down to maintain an adjustable distance between the ego and target. If the target is accelerating again, the ACC controller increases the velocity of the ego by keeping the predefined distance to the target. When the maximum speed, defined by the driver of the ego, is reached, the ACC controller keeps the velocity constant.

The determination of the relevant target to follow is done by detecting the targets in front via the forward-looking sensor and using other sensor signals (for example signals of the *Electronic Stability Programm* (ESP)) in a path prediction approach. It is also

possible to use image processing equipment for assigning the detected targets to the according lanes [18].

1.2.4.2. Automatic Emergency Braking (AEB)

Automatic Emergency Braking (AEB) is used to avoid collisions or to mitigate the effects of an accident. This system initiates an emergency brake if the driver does not respond to a warning signal. Targets are detected via different sensors in front of the vehicle. The RADAR sensor of the ACC is used in this ADAS to detect targets, which are far away. For the short-range area extra sensors like short-range RADAR sensors, *Light Detection and Ranging (LIDAR)* sensors and cameras are used. In critical situations, the brake pressure is increased in the shortest possible time, until the level, at which the ABS is working is, is reached [10].

1.2.4.3. Lane Keeping Assist (LKA)

In contrast to the ACC described in section 1.2.4.1, the *Lane Keeping Assist (LKA)* is a guidance system that operates in lateral direction. It is an extension of the *Lane Departure Warning (LDA)*. LDA is only a warning system that informs the driver via audio, visual and/or haptic signals if the vehicle is going to cross the limiting road markings, which are detected by means of a camera system. LKA however intervenes in the lateral guidance to keep the target on a specific lane. LKA systems can also be subdivided into two systems, which operate in different ways. The first type only supports the driver in keeping the vehicle on the desired lane via active intervention in the steering system. In the situation of driving over the limiting road markings, the motorist is informed via the LDW. In comparison to the first system, the second type tries to keep the vehicle in the centre of the lane. Similar to the first system, the LDW warns the driver in case of leaving the desired lane [10].

1.2.4.4. Active Lane Change Assist (LCA)

According to “Functional Requirements Lane Change Assist” [24] of MAGNA STEYR Fahrzeugtechnik AG & Co KG a *Lane Change Assist Type II (LCA II)* or also known as *active Lane Change Assist* is an ADAS that is used to perform lane change manoeuvres autonomously. LCA II is a partial automated ADAS (SAE level 2) which means, that it is responsible for the lateral and longitudinal driving task. It can only be used in some driving modes (in this case only on highways), while the driver has to monitor the environment and react in situations, which can not be handled by the system. The LCA determines the position and the trajectory of the ego vehicle and the surrounding targets via different sensor types like RADAR, LIDAR, cameras and so on. The use of a LCA II is restricted for highways with straight ahead scenarios, however its functionality can be extended for the use on curvy road sections and non-highway roads. A LCA delivers

manoeuvre recommendation via a *Human Machine Interface* (HMI). The ADAS only performs an autonomous lane change, if the motorist confirms the the recommended manoeuvre via the HMI.

1.2.4.5. Highway Chauffeur (HWC)

In the presentation “The Highway-Chauffeur” [17] of the *PEGASUS Symposium (9th November 2017, Aachen, Germany)* the requirements and conditions of a SAE level 3 (conditional automation) driving function called *Highway Chauffeur* (HWC) are described. According to Figure 1.4, conditional automation means that the longitudinal and lateral driving task is performed by the system at specific driving modes, in this case on the highway, within certain limits. Hence the longitudinal and lateral driving task is performed by the ADAS, it is a combination of the before mentioned systems ACC (see 1.2.4.1), AEB (see 1.2.4.2), LKA (see 1.2.4.3) and LCA (see 1.2.4.4). Figure 1.5 illustrates the area of application of a SAE level 3 HWC. The driver does not need to monitor the system at all times but has to take back the control of vehicle, if he is requested to.

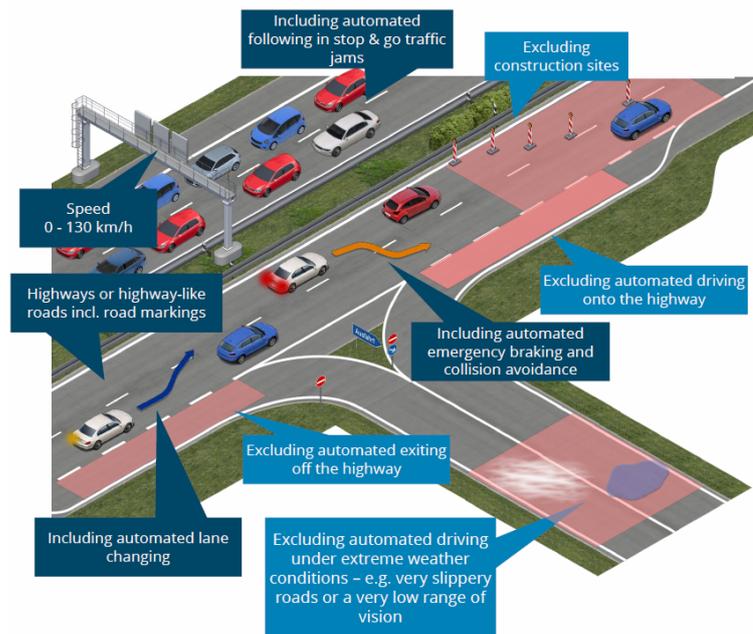


Figure 1.5.: Basic functionality of a Highway Chauffeur (SAE level 3), [17]

The Human driver:

The driver has the ability to decide, if he/she wants to use the driving function and if the activation of the ADAS is reasonable. The motorist has to keep in mind, that the driving task is given back to him/her after a sufficient time margin, if the system is not

capable to handle a situation. In addition to that, the driver can deactivate the driving system at all time.

The ADAS HWC:

The system activation via the driver is only possible under defined conditions. If the HWC is activated, the system monitors the environment and performs the lateral and longitudinal driving task. When the system is going to exceed its limits, the driver has to be informed/warned to take back the control of the driving task and after a sufficient time margin, the system deactivates itself.

Functionality of the HWC

The HWC should be able to adapt speed and distance according to the traffic situation within specific limits. In addition to that, the system should perform lane change and overtaking manoeuvres autonomously. This ADAS should also be able to handle traffic jams.

Requirements for activating the HWC according to PEGASUS [17]

- driving on highways or highway-like roads
- not driving faster than speed limit
- not driving backwards
- not driving in construction sites
- not driving under bad weather or road conditions
- not driving under low visibility conditions
- no coverings on the sensor
- proper state of the vehicle (brakes not overheated, enough fuel...)
- no relevant system components are defect
- driver seat is occupied and seat harness buckle is closed
- doors are closed
- no trailer is used

1.2.5. Sensors for environmental recognition

As described in section 1.2.1, the driver is interacting with the environment via the vehicle. Introducing ADAS into this system as an additional component helps the driver with the driving task and reduces the effort for human execution. For this process, every support system, no matter which degree of automation, needs information concerning the environment. This information is gathered via sensors and sensor combinations of different technologies like RADAR, LIDAR, ultrasound or image processing technology. Figure 1.6 illustrates the different areas (front, side and rear) of the vehicle, which are monitored with various sensor types and also gives an overview of their field of application.

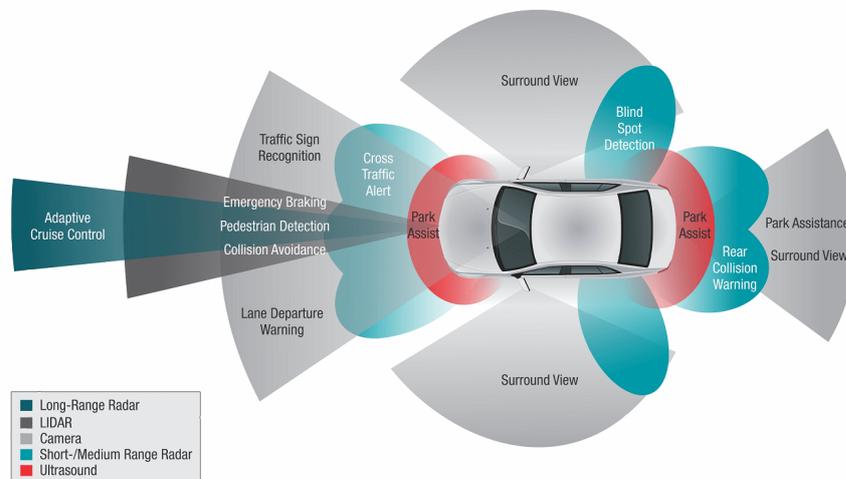


Figure 1.6.: Sensors for environmental recognition, [3]

The monitoring areas can be split in sections according to the range and the *Field of View* (FOV) [18],[22].

1. Extreme short range areas

This area ends at distances of about 2.5 m. Objects in this area are detected via ultrasound sensors. Field of application is for example in *Parking Assistance Systems*.

2. Short-range area

Areas with a range of maximal 60 m - 80 m are called short-range areas. Ultrasound sensors (up to 4.5 m) and RADAR sensors are used to monitor for example the blind spots of the vehicle.

3. Mid-range area

Approximately 160 m is the limit for this area, in which RADAR sensors, LIDAR sensors and cameras are utilized.

4. Long-range area

Distances till 200 m are categorized as long-range areas. RADAR sensors are used to gather information about the environment in this area. Field of application is for example the ACC.

Each sensor collects data of the detected objects or targets in its own FOV. It is possible, that the same object is detected by more than one sensor. For this reason complex sensor fusion algorithms are used to merge the data, gathered by all environmental recognition sensors. This process step makes it possible to generate a consistent and reliable map. This representation of the real environment is then used by the *Advanced Driver Assistance Systems* for supporting the driver or performing the driving task.

1.2.5.1. Performance influencing factors and perturbations

The before mentioned sensor technologies show different sensitivity according to external perturbation. This section gives a compact overview over the performance influencing factors of LIDAR sensors, RADAR sensors, ultrasonic sensors and cameras according to [9], [16], [19].

1. LIDAR sensors

- Atmosphere as an influencing factor

A certain amount of the light, emitted by the the LIDAR sensor is reflected diffuse or absorbed by the atmosphere.

- Target as an influencing factor

- Diffuse reflection

If the emitted light of the LIDAR reaches an object or target, it is then reflected on this target's surface in the shape of a half hemisphere. This means, that the sensor does not receive the amount of light reduced by the amount of diffuse reflection and absorption by the atmosphere. The sensor only receives the light, which is directly reflected. This is often less than 20%.

- Environment as an influencing factor

- Coverings of the the sensor

All kinds of coverings like snow, dirt etc. in front of the sensor cause malfunction.

- Road profile

Extreme changes of the slope can cause total reflection of the beam in a wrong direction.

- Weather as an influencing factor

The weather has an enormous influence on the functionality of the LIDAR sensor. Strong rain, snow or fog are disturbing the performance of this technology.

2. RADAR sensors

- Environment as an influencing factor

- Coverings of the the sensor:

A thick layer of dirt or snow in front of the RADAR sensor reduces the performance or can cause malfunction.

- Road profile

Extreme changes of the slope can also cause, that the transmitted RADAR signal is reflected on the targets surface into a wrong direction. In this case the RADAR sensor can not detect the target.

- Tunnels

RADAR sensors do not work correctly in tunnels. The surrounding surface of the tunnel causes multiple reflections.

- Other signals

RADAR sensor signals of vehicles can be interfered and disrupted by other RADAR signals in the same frequency band.

- Weather as an influencing factor

Compared to LIDAR, the RADAR sensor can operate in foggy conditions. However, heavy rain also disturbs the RADAR signal.

- Driving state as an influencing factor:

Accelerating or decelerating can contribute to large pitch angles. This can lead to not receiving the transmitted and reflected signal.

3. Ultrasonic sensors

- Target as an influencing factor

Objects or targets, which hardly reflect acoustic waves are problematic for this kind of sensors.

- Weather as an influencing factor

Weather conditions are also influencing this technology. Especially rain causes perturbation of the sensor signal.

- Environment as an influencing factor
 - Coverings of the the sensor:

As mentioned in part 2, coverings in front of the sensor reduce signal intensity and can contribute to system failure.
 - Other signals

Interference due to signals of other ultrasonic sensors can lead to malfunction.

4. Cameras

- Weather as an influencing factor

Due to the fact, that a standard camera system is based on image processing, this system is similar to a human eye. If it is raining or snowing heavily, if the sun is blinding, if it is dark or if there are extreme foggy conditions, this system is not working. Night vision cameras or infra-red cameras are exceptions.
- Environment as an influencing factor
 - Coverings of the lens

All different kinds of coverings in front of the lens make a use of this system impossible.

1.2.6. Development process on virtual level

Due to high system complexity of ADAS, the complexity of test scenarios and the enormous amount of needed tests, real drive tests are not economic in the development phase. Furthermore, the interaction between different road users and environmental influencing factors like reflected sunlight on a wet road surface etc. in reality make it nearly impossible to fulfil the requirement for test reproducibility and exact same test conditions. For this reason, tests are carried out in a virtual environment, where functionality of these systems and their components can be simulated.

The main idea of a virtual drive test is to generate a realistic replication of the real drive test in a virtual environment and to use the advantages like flexibility, reproducibility, effort reduction etc. of the simulation in the early development phase. Utilizing appropriate simulation methods contribute to an efficient conceptual design and an efficient development of components and systems. These virtual methods also help to reduce the time gap, till a real prototype is available.

Like a real drive test, a virtual drive test consists of similar components. The centrepiece is a virtual vehicle prototype containing different parts like models, software code or hardware, depending on the progress of the development process. Defined interfaces allow

combining components of different development stages. The virtual vehicle is controlled by a virtual driver model, parametrized via a behaviour model. Open-Loop- and Closed-Loop-Manoeuvres and acting by means distance-, time- or event-based manoeuvre steps are possible.

At the beginning of the development process and the early concept phase all parts of the drive test are virtual, but are replaced step by step by real components during the development process until a real drive test with a human driver and real road users can be done.

The simulation is not only a transformation of the real test into the virtual environment, but also includes evaluation and assessment methods. The main goal of the virtual drive test is to shift the procedure of test manoeuvres and related assessment criteria to the beginning of the concept phase.

With the use of virtual prototypes and virtual drive tests, decisions concerning the design of the whole vehicle can be assessed at an early stage of the development process. Before the first real prototype is built, it can be checked if the specifications of components and systems contribute to the desired goals of the complete vehicle. The V-Model, described in section 1.2.6.1, is a very important instrument in this process.

1.2.6.1. V-Model and In-the-Loop methods

Due to the large amount of software in driver assistance systems, it is obvious to use the V-Model, a common tool in software engineering, or rather a modified approach of the V-model for the development process of such systems. In general, the V-Model is a chronological representation of the development process. An important fact is, that the different steps of the whole development phase are arranged in form of the letter V, consisting of two branches. The left branch describes the requirements, the architecture, the design and the specification of the product. At the bottom, the implementation is done. Integration, testing and validation at different levels (component, system, complete vehicle) are performed on the right branch. Figure 1.7 illustrates the conventional V-model. A disadvantage of the standard V-model is, that checking, if a specification is reached, is only possible after the integration process [10], [8].

To minimize the risk of not fulfilling the customers requirements at the end of the development process, the V-model is modified. This helps to determine the quality of the development at earlier stages.

The integrated methods are:

1. *Model-in-the-Loop* (MiL)
2. *Software-in-the-Loop* (SiL)
3. *Hardware-in-the-Loop* (HiL)
4. *Vehicle-in-the-Loop* (ViL)

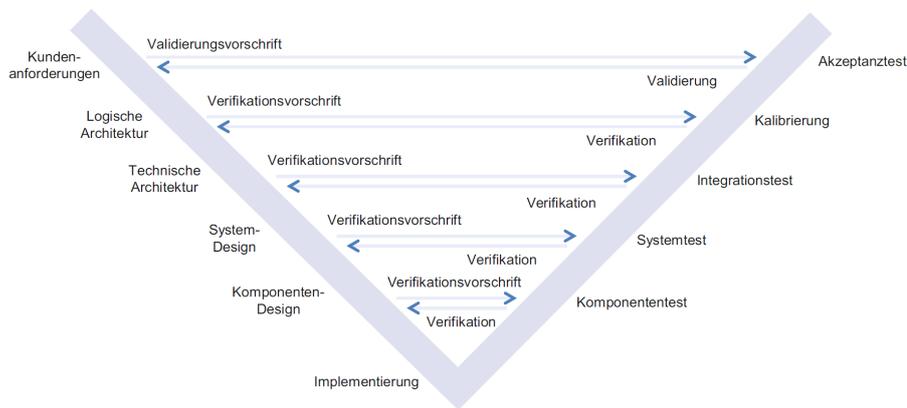


Figure 1.7.: Development process according to V-model [10]

The main idea of the modified approach is to add at a specific stage in the development process available models or real components to a virtual environment to get an assessable complete system. Due to the fact, that there are no real components until the implementation step is reached, the simulation environment must offer the possibility for a virtual integration. Figure 1.8 shows the V-model with the integrated In-the-Loop methods.

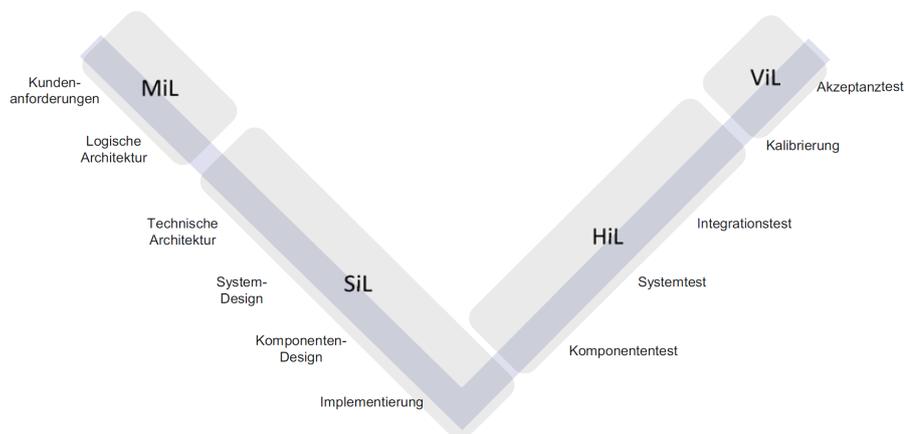


Figure 1.8.: V-Model with integrated In-the-Loop methods [10]

1. Model-in-the-Loop (MiL)

MiL is used at the top area of the left branch in the V-model. In this method, model-based software is used to generate algorithms, which correspond to the functional development goal. At this point, these algorithms do not have any relation to the final hardware. Integrating such models into a simulation environment and using a virtual drive test, consisting of the necessary components like the road, the

environment, sensors, a driver model etc. makes validation at this stage possible. Combining the simulation of this step with a driving simulator can help to assess the functionality. This procedure also helps to determine the acceptance of the customers at a very early stage.

2. Software-in-the-Loop (SiL)

SiL follows after MiL and is utilized until the component design. In SiL the real software code, which is later transferred onto the real hardware, is investigated on a computer. This approach helps to check the specifications of the system components before they are implemented. By means of MiL, SiL and an efficient simulation environment, a virtual integration of the complete system is possible. At the bottom of the left branch, a virtual prototype is available, in which the functionality of each component, as well as the interaction with the complete system and its functionality can be tested and verified. With the help of virtual drive tests, the impact of changes at different levels can be evaluated. The fact, that this is in general an automated process and also faster than real time, makes a virtual prototype to an important and powerful tool. It is also possible to check the effects of misuse or malfunction of specific components at different prototype setups.

3. Hardware-in-the-Loop (HiL)

In this method, the components of the previous step (SiL) are transferred onto real hardware. This is also done in several steps. First of all, testing and verifying is done on component level, where the hardware components are connected to the simulation environment. After that process, the components can be integrated step by step.

4. Vehicle-in-the-Loop (ViL)

The top area of the right branch of the V is called Vehicle-in-the-Loop. At this point, a real vehicle is already existing. ViL provides a method for testing the real vehicle in a virtual environment.

There are two different possibilities:

- Replacing the real sensors

Via an interface between the real vehicle and the virtual environment, simulated sensor signals are used instead of the output of real sensors. This method is used, if the effort for stimulating the real sensors is too big.

- Using the real sensors

In this case, the real sensors are stimulated to generate particular output signals.

A big advantage of ViL is the safe testing process.

Table 1.1.: Transition from a virtual environment to the real world, [10]

	MiL	SiL	ECU-HiL	System-HiL	roller dynamometer	ViL	drive test
Function code	V	R	R	R	R	R	R
ECU	V	V	R	R	R	R	R
System	V	V	V	R	R	R	R
Vehicle	V	V	V	V	R	R	R
Driver	V	V	V	V	V/R	V/R	R
Vehicle dynamics	V	V	V	V	V	R	R
Experience	V	V	V	V	V	R	R
Road	V	V	V	V	V	R	R
Traffic/environment	V	V	V	V	V	V	R

In Table 1.1 the amount of virtual components at different development steps is shown. V stands for virtual component and R signals a real component. Take the first column (MiL) as an example. At this point of the development process, each component (function code till traffic/environment) is only available as a virtual product. In the following In-the-Loop methods, more and more components are real.

Although there are a lot of advantages concerning the use of the virtual development process in the area of driver assistance systems, real drive tests will still be needed for validation and final system proof.

1.3. Simulation with IPG CarMaker®

The simulation software used in this thesis is IPG CarMaker® 6.0.4. This simulation tool consists of two main parts:

1. Virtual Vehicle Environment (VVE)

The VVE includes a virtual vehicle, a computer modelled representation of the real car. Different equations and mathematical formulas, which describe the motion of the vehicle, the kinematics and so on, are used to define a multi-body system. A validated data set can be used to parametrize this model according to the real world counterpart. Models of the powertrain, the brakes, the tires, the chassis etc. are implemented in the virtual vehicle. Also models of controllers for example an ACC controller, can be integrated.

Another important part in the VVE is the virtual road. This part represents the real road or track in the virtual environment. It can either be designed in a sketching editor, called “Scenario Editor”, by combining different road segments like curves, straights etc. or using measured data to generate the road file.

Furthermore, a virtual driver model is used to imitate the driving behaviour of a human. All necessary driving tasks like steering, operating the brake-, gas- or clutch pedal, shifting and so on are performed by this model. A smart driver model, which can also plan ahead or a simple control mechanism are available.

2. CarMaker[®] Interface Toolbox (CIT)

The CIT is a tool, used to manage different parts of the VVE. For example, the manoeuvre definition, starting and stopping of the simulation or displaying the results can be done via the CIT. The available tools can be categorized into four applications:

- Control and Direct Access Tools (starting and stopping the simulation, direct variable access etc.)
- Parametrization Tools (specifying different parameters of the VVE)
- Analysis and Visualization Tools (viewing and analysing data during or after the simulation)
- File Management Tools (creating, saving or modifying files of the simulation)

IPG offers two different software approaches for the development process and the implementation of the VVE:

1. CarMaker[®]/Office

This approach is based on the simulation of the VVE on an ordinary computer. In this case, the simulation can be slower, equal or faster than real time, depending on the computing power. If a software modelled controller or a vehicle subsystem has to be investigated and the integration of hardware is not relevant at this point, the CarMaker/Office approach is used.

2. CarMaker[®]/HiL

If a hardware module for example an ECU should be tested, a real time approach is necessary. CarMaker[®] with Hardware-in-the-Loop offers the possibility to simulate at a timeline that equals the timeline of the real world. In this case the VVE is simulated on a computer with a real time operating system.

In this thesis the function of a HWC with a decision-making-algorithm, implemented in MATLAB/Simulink, should be investigated. Due to the fact, that no hardware is going to be tested, the CarMaker[®]/Office approach is chosen.

CarMaker[®] for Simulink, available in CarMaker[®]/Office, integrates CarMaker[®] into the simulation and modelling environment MATLAB/Simulink of MathWorks[®] and forms a co-simulation environment. Using CarMaker[®] for Simulink extends MATLAB's search path and enables the usage of CarMaker[®] for Simulink blocks. This makes writing, reading or defining of CarMaker[®] quantities in Simulink possible. [11], [13].

1.4. Requirements for the HWC model of this thesis

The requirements and the functionality of the HWC model implemented in this thesis are based on the requirements and the functionality according to PEGASUS, described in section 1.2.4.5.

The main points of this system are summarized below:

1. Field of application

The HWC is only available on highways under good weather and visibility conditions. Automatic entering and exiting the highway as well as driving in construction sites on the highway can not be handled by this ADAS.

2. Functionality

The HWC is intended to take control of the lateral and longitudinal driving task. Information from monitoring the environment is used to make decisions when a lane change manoeuvre is possible or which target is relevant for the ACC. The system is active until the driver deactivates it, for example if he/she wants to exit the highway. The system can only be activated if the conditions are appropriate.

Note: An AEB was not implemented in the HWC of this thesis.

2. Methodology

In this chapter the procedure and the approaches with the necessary derivations needed for the implementation of the HWC model are described. In addition to that, the structure and the functionality of the Simulink model is explained.

2.1. Virtual road

The aim of this thesis is to test the functionality of a HWC model, created in MathWorks[®] software tool MATLAB/Simulink, in a virtual environment. A section of the highway A2 between Graz and Vienna should be used in the simulation. For this reason, measurement data of the road between the transport hub Graz-West and Laßnitzhöhe were used. At the beginning the virtual road was available in a file format called OpenDrive[®]. This file format is based on *Extensible Markup Language* (XML) and was introduced in the year 2006 by VIRES Simulationstechnologie GmbH as an open standard for the logical description of roads and networks. It was developed by VIRES Simulationstechnologie GmbH, Germany, in close cooperation with Daimler Driving Simulator, Sindelfingen, Germany [7]. According to the homepage of VIRES [23] there are several advantages for using OpenDrive[®]:

- vendor-independency
- containing all key features of real road networks
- established format

The main idea of using OpenDrive[®] is to make the exchange of data between users and various creators possible or easier[6].

Unfortunately, IPG CarMaker[®] 6.0.4 is only able to read its own road file format called “Road5”. For this reason IPG provided an executable to convert the original road data in OpenDrive[®] to Road5. A big disadvantage of this conversion is that some information gets lost during the transformation process. Therefore the road file had to be adjusted manually. Several problems occurred due to this conversion and a few of them are described in the next section.

2.1.1. Modification of road data

This section deals with the problems caused by the conversion of the virtual road and the troubleshooting process. However, before the troubleshooting process can be described, it is necessary to understand the basic terms of the road network generation in CarMaker[®]'s *Graphical User Interface* (GUI) called "Scenario Editor".

1. Link

A road network consists of several *Links*. So called *Nodes* limit the start and at the end of a *Link*. The *Reference Line* defines the course of a *Link* and includes a local s-t-coordinate system. The origin of this coordinate system is in the starting *Node*, s is pointing into the course direction and t is orthogonal to s. A *Link* can consist of different *Segments* for example straights, turns, clothoids or point lists.

2. Lane

The lateral extension (in t-direction) of a *Link* is called *Lane*.

3. Lane Sections

Lane Sections are areas of a *Link* with a specific number of *Lanes*. Adjusting the width of a *Lane* within a *Lane Section* makes diverging or merging of *Lanes* possible, shown in figure 2.1.

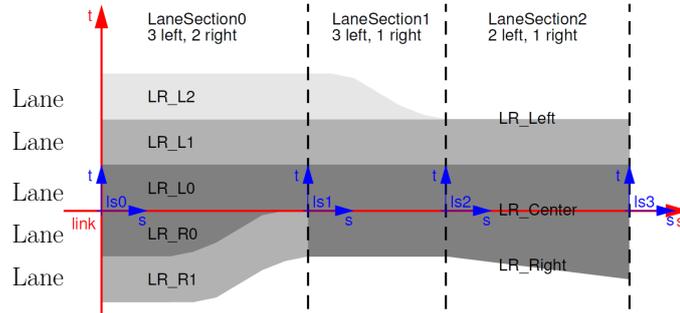


Figure 2.1.: Lane Sections, [13]

4. Junction

Junctions are road elements, used to combine different *Links* with each other. Two different types of *Junctions* are available: *normal Junctions* shown in figure 2.3 and *Ramps* illustrated in figure 2.2. *Ramps* are used for modelling highway entrances and exits.

5. Route

A possible road map for a vehicle is defined by the *Route*. The *Route* is conform to the *Reference Line*. Only at *Junctions* (normal *Junctions* not *Ramps*) the *Route* is calculated via a cubic polynomial. In figure 2.3 the green lines depict the *Routes*.

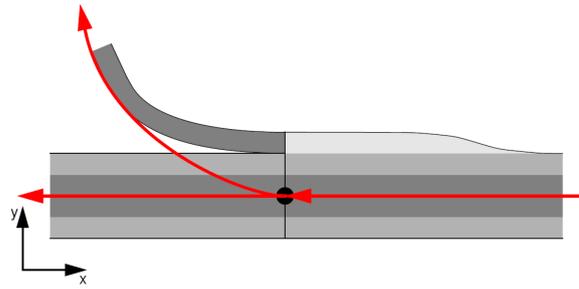


Figure 2.2.: Ramp, [13]

6. Path

If the lateral offset of a Route is adjusted, a individual *Path* can be created. The red lines in figure 2.3 are Paths.

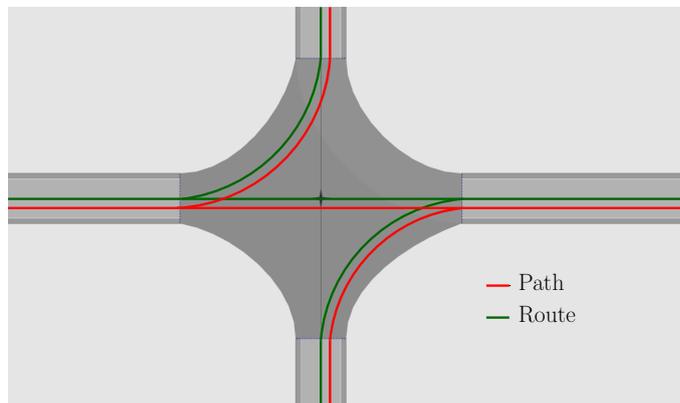


Figure 2.3.: Routes and Paths

In cooperation with the industrial partner MAGNA STEYR Fahrzeugtechnik AG & CoKG adjustments on the converted road were made in the Scenario Editor as well as directly editing the text file of the road. In the following paragraph, the problems associated with the conversion process and how they were solved are discussed.

Most of the problems occurred at entrances and exits of the highway. The entrance or exit of a highway is created by using a junction of the type Ramp. In the drawing process, the link of the exit is first connected to the relevant arm of the Ramp. After that, the necessary lateral offset has to be defined to fit the link into the right and final position. The road converter is not able to calculate the needed offset and places the link at the initial position. For this reason the lateral offset had to be adjusted manually. At very short links it can happen, that the surface of the road gets warped at the final position. In this case, a new and longer link was drawn instead of just adjusting the converted one. A large amount of very small segments in the area of the ramp also cause issues. In this situation, the number of segments was reduced by deleting several of them. In

a lot of cases, the transitions between links were not smooth. Figure 2.4 illustrates this situation. For this reason, elevation profiles at these areas had to be adjusted.



Figure 2.4.: Converted road without adjustments

2.2. Refloating a consisting HWC model

After the process of adjusting the the converted road, a Simulink model of a HWC implemented for IPG CarMaker[®] 4.0.3 had to be refloated. As a main criteria for a possible lane change, a modified approach of the *Time To Collision* (TTC) is used. The trajectory for the lane change manoeuvre is based on a polynomial of fifth order. The following items list the limitations of the refloated Simulink model:

- Works only on straight roads
- No sensor models are used \implies direct addressing the targets to get the relevant data for the calculation (makes the model very inflexible)
- Number of targets is limited due to second point
- Only one lane change manoeuvre to the next left lane is possible
- No lane change manoeuvre to the right lane is possible
- Due to direct addressing each target, the position of each target is known at each time. In this case, no method for the assignment of the targets to their lanes is need and hence not implemented.

The next items repeat the main requirements of the new HWC. This new model

- Should not be limited to driving on straight roads
- Should be independent of the amount of targets and their names or IDs
- Should use sensor models for gathering environmental information
- Should be able to perform multiple lane change manoeuvres to the left and right lane

- Should include a smart decision making algorithm (when to perform a lane change manoeuvre, target selection for the ACC)
- Should be able to assign each detected target to the lane, on which it is driving

According to all these new requirements it was decided to implement a complete new model. Two essential things, the polynomial of fifth order for the trajectory generation and the modified TTC approach, described in the next section, are also used in the new HWC of this thesis.

2.2.1. Derivation of the Time To Collision (TTC)

In this section, the *Time To Collision* (TTC) is derived. The equations (2.1) - (2.8) show the derivation of the travelled distance $x = s_x$ as a function of the time t . The initial position is described by $x_0 = s_{x_0}$, the initial velocity is specified by $\dot{x}_0 = v_{x_0}$ and the acceleration is defined by $\ddot{x} = a_{x_0} = \text{const}$.

$$\frac{d\dot{x}}{dt} = a_{x_0} \quad (2.1)$$

$$d\dot{x} = a_{x_0} dt \quad (2.2)$$

$$\int_{\dot{x}_0=v_{x_0}}^{\dot{x}} d\dot{x} = \int_{t_0=0}^t a_{x_0} dt \quad (2.3)$$

$$\dot{x} - v_{x_0} = a_{x_0} \cdot t \quad (2.4)$$

$$\frac{dx}{dt} = a_{x_0} \cdot t + v_{x_0} \quad (2.5)$$

$$\int_{x_0=s_{x_0}}^{s_x} dx = \int_{t_0=0}^t (a_{x_0} \cdot t + v_{x_0}) dt \quad (2.6)$$

$$s_x - s_{x_0} = a_{x_0} \cdot \frac{t^2}{2} + v_{x_0} \cdot t \quad (2.7)$$

$$s_x = a_{x_0} \cdot \frac{t^2}{2} + v_{x_0} \cdot t + s_{x_0} \quad (2.8)$$

2.2.1.1. Time To Collision without acceleration

As described in the slides of the lecture Vehicle Safety 2 [9], TTC is the time, until a collision would happen, if two vehicles would continue driving with their actual speeds on their present course. Figure 2.5 illustrates the position s_x of the ego and the target depending on the time t for a situation, in which the ego has a higher velocity as the target. In this case, the equation (2.8) is reduced by the acceleration.

$$s_x = v_{x_0} \cdot t + s_{x_0} \quad (2.9)$$

In case of a collision, the position of the ego and the target is equal:

$$s_{x,Ego} = s_{x,Target} \quad (2.10)$$

$$v_{x_0,Ego} \cdot t + s_{x_0,Ego} = v_{x_0,Target} \cdot t + s_{x_0,Target} \quad (2.11)$$

$$(v_{x_0,Target} - v_{x_0,Ego}) \cdot t + s_{x_0,Target} - s_{x_0,Ego} = 0 \quad (2.12)$$

$$\Delta v_{x_0,Sensor} \cdot t + \Delta s_{x_0,Sensor} = 0 \quad (2.13)$$

$$TTC = -\frac{\Delta s_{x_0,Sensor}}{\Delta v_{x_0,Sensor}} \quad (2.14)$$

Equation (2.14) describes the TTC without acceleration. To get a positive TTC, that means the accident will occur at this time in the future, the velocity of the ego has to be bigger than the velocity of the target. In the reversed case, the equation (2.14) would deliver a negative TTC value, which means, no accident will occur.

2.2.1.2. Time To Collision with acceleration

Note: In this section, acceleration is used as a general term. Negative acceleration would be deceleration.

It is possible to modify the standard TTC approach of section 2.2.1.1 by using all mathematical terms of the equation (2.8). To determine the TTC for this case, the position of the ego and the target has to be the same.

$$s_{x,Ego} = s_{x,Target} \quad (2.15)$$

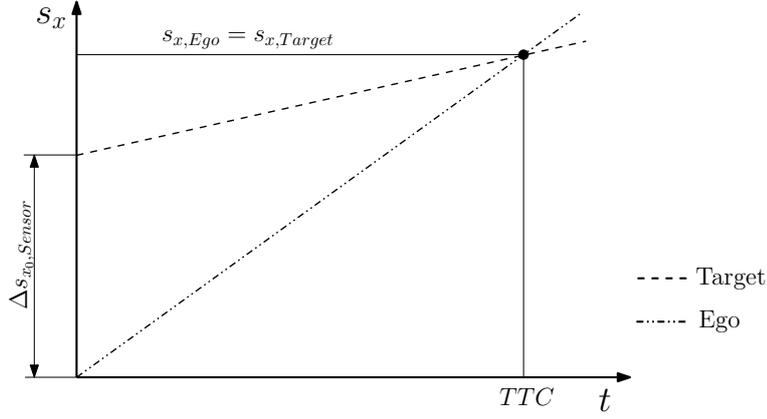


Figure 2.5.: TTC without acceleration

$$a_{x_0,Ego} \cdot \frac{t^2}{2} + v_{x_0,Ego} \cdot t + s_{x_0,Ego} = a_{x_0,Target} \cdot \frac{t^2}{2} + v_{x_0,Target} \cdot t + s_{x_0,Target} \quad (2.16)$$

$$(a_{x_0,Target} - a_{x_0,Ego}) \cdot \frac{t^2}{2} + (v_{x_0,Target} - v_{x_0,Ego}) \cdot t + s_{x_0,Target} - s_{x_0,Ego} = 0 \quad (2.17)$$

$$\Delta a_{x_0,Sensor} \cdot \frac{t^2}{2} + \Delta v_{x_0,Sensor} \cdot t + \Delta s_{x_0,Sensor} = 0 \quad (2.18)$$

$$TTC = \frac{-\Delta v_{x_0,Sensor} \pm \sqrt{\Delta v_{x_0,Sensor}^2 - 4 \cdot \frac{1}{2} \cdot \Delta a_{x_0,Sensor} \cdot \Delta s_{x_0,Sensor}}}{2 \cdot \frac{1}{2} \cdot \Delta a_{x_0,Sensor}} \quad (2.19)$$

with

$$D = \Delta v_{x_0,Sensor}^2 - 2 \cdot \Delta a_{x_0,Sensor} \cdot \Delta s_{x_0,Sensor} \quad (2.20)$$

The solutions of the equation (2.19) are depending on the result under the square root, named as D and described by equation (2.20):

1. Case 1: $D > 0$

Equation (2.18) has two real zeros, with different values

2. Case 2: $D = 0$

Equation (2.18) has two real zeros, with the same value

2. Methodology

3. Case 3: $D < 0$

Equation (2.18) has no real zeros

In Figures 2.6 - 2.8, the course s_x of the ego and the target, calculated with the equation of motion (2.8), including the longitudinal acceleration a_{x_0} , is illustrated. Both vehicles are decelerating. In figure 2.6 it can be seen, that both lines are intersecting each other. The modified TTC approach delivers two different real TTC values (Case 1), which are bigger than zero. The smaller TTC value is the the relevant one. In figure 2.7 however, both lines are just touching each other. This equals Case 2 with two real zeros with the same value. Figure 2.8 illustrates Case 3, where no contact or intersection of both lines occur.

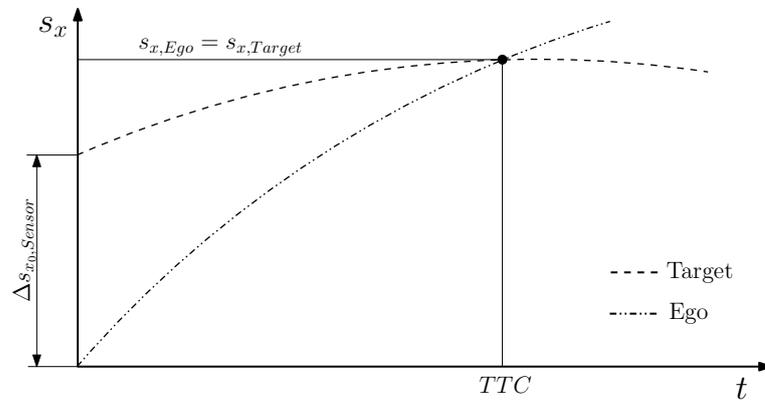


Figure 2.6.: TTC with modified approach: Case 1

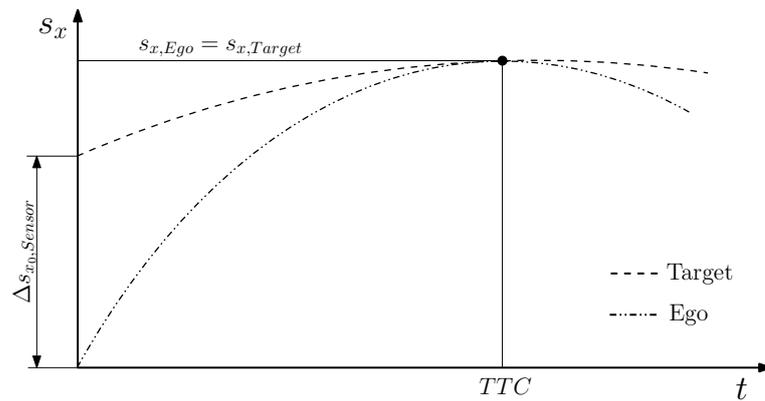


Figure 2.7.: TTC with modified approach: Case 2

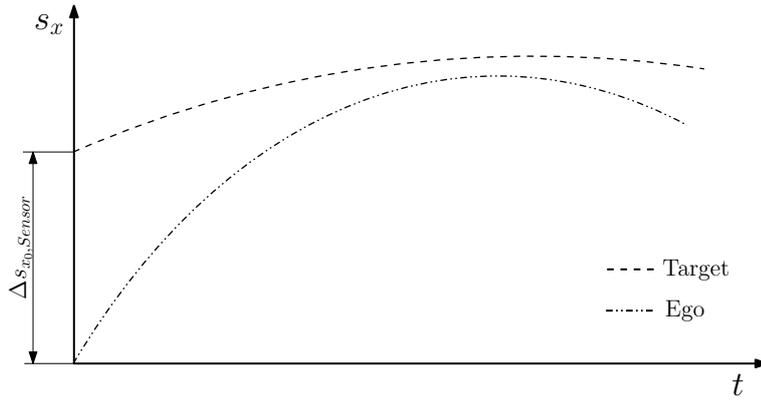


Figure 2.8.: TTC with modified approach: Case 3

The next section shows, how the trajectory for a lane change manoeuvre is calculated.

2.2.2. Lane change manoeuvre

If the decision for a lane change manoeuvre is made, a trajectory has to be planned. Samiee describes in [20] an approach for trajectory generation of a vehicle, that is driving on a straight highway without intersections, by means of a polynomial of fifth order, see equation (2.21). In this equation for the lateral displacement $y(t)$ the variable t defines the time and a to f represent the polynomial factors.

$$y(t) = a \cdot t^5 + b \cdot t^4 + c \cdot t^3 + d \cdot t^2 + e \cdot t + f \quad (2.21)$$

Following assumptions were made in this paper to define the polynomial factors a to f for a lane change manoeuvre to the next right lane :

- lateral displacement $y(t)$ at the beginning of the lane change manoeuvre ($t = 0$) is zero

$$y(t = 0) = 0 \quad (2.22)$$

- lateral displacement $y(t)$ at the end of the lane change manoeuvre ($t = t_m$ with t_m as duration for the manoeuvre) has a value that equals the width of the lane
 - lane change manoeuvre to the next right lane would mean: $-h$
 - lane change manoeuvre to the next left lane would mean: h

$$y(t = t_m) = -h \quad (2.23)$$

- lateral velocity $\dot{y}(t)$ at the beginning of the lane change manoeuvre ($t = 0$) is zero

$$\dot{y}(t = 0) = 0 \quad (2.24)$$

- lateral velocity $\dot{y}(t)$ at the end of the lane change manoeuvre ($t = t_m$) is zero

$$\dot{y}(t = t_m) = 0 \quad (2.25)$$

- lateral acceleration $\ddot{y}(t)$ at the beginning of the lane change manoeuvre ($t = 0$) is zero

$$\ddot{y}(t = 0) = 0 \quad (2.26)$$

- lateral acceleration $\ddot{y}(t)$ at the end of the lane change manoeuvre ($t = t_m$) is zero

$$\ddot{y}(t = t_m) = 0 \quad (2.27)$$

In addition to these points, Samiee also assumes, that the vehicle's longitudinal velocity is constant during the lane change manoeuvre.

Using equation (2.21) and applying the conditions (2.22) to (2.27) delivers the equation (2.28) for the lateral displacement of a lane change manoeuvre to the next right lane.

$$y(t) = -\frac{6 \cdot h}{t_m^5} \cdot t^5 + \frac{15 \cdot h}{t_m^4} \cdot t^4 - \frac{10 \cdot h}{t_m^3} \cdot t^3 \quad (2.28)$$

Changing the condition (2.23) from $-h$ to h and using the remaining conditions as they are, delivers the equation (2.29) for the lateral displacement of a lane change manoeuvre to the next left lane.

$$y(t) = -\left(-\frac{6 \cdot h}{t_m^5} \cdot t^5 + \frac{15 \cdot h}{t_m^4} \cdot t^4 - \frac{10 \cdot h}{t_m^3} \cdot t^3\right) \quad (2.29)$$

Figure 2.9 illustrates the shape of the polynomial for a lane change manoeuvre to the next left lane. It can be seen, that only the lateral displacement from $0 < t < t_m$ of the polynomial is relevant. In this figure a manoeuvre duration of $t_m = 4$ s is chosen.

Although there are some limitations of the method explained before, this approach is going to be used in the HWC, implemented in this thesis, and the suitability of this method for curvy roads is going to be investigated.

For the continuous use of this approach in the Simulink model, the lateral displacement calculated via the equations (2.28) and (2.29) has to be added to the initial position of the ego (before the lane change manoeuvre) on the road. After the lane change manoeuvre

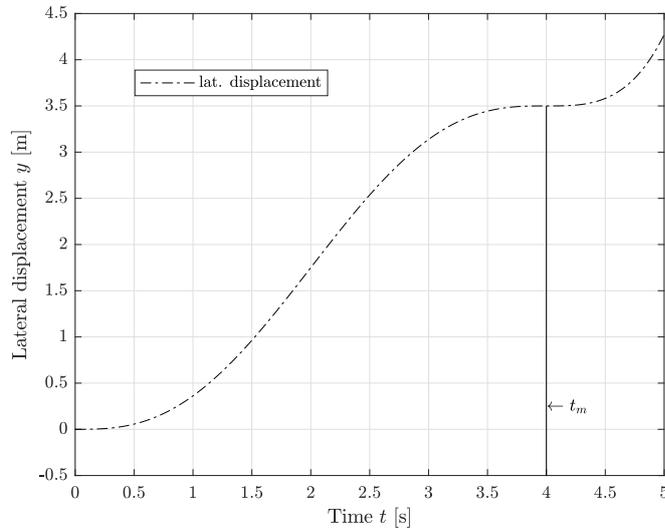


Figure 2.9.: Shape of the polynomial

is finished, this means if $t = t_m$ and the maximum lateral displacement of h or $-h$ was added to the initial position (in this case $y(t = 0) = 0$), this value has to be kept until a new lane change manoeuvre is initiated. Figure 2.10 depicts this situation.

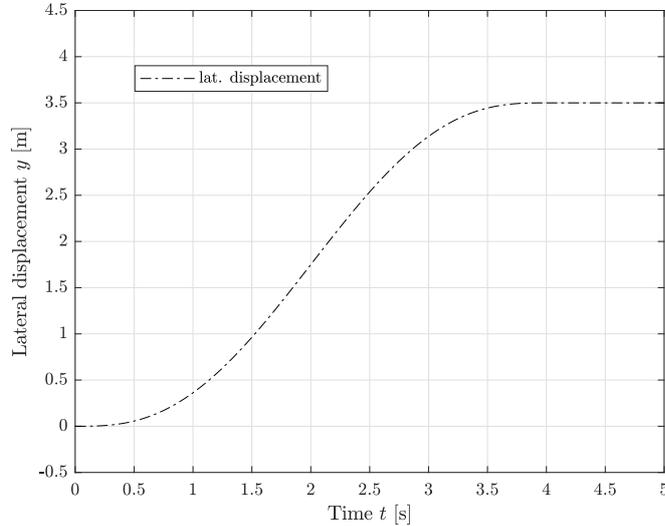


Figure 2.10.: Lateral displacement for a lane change manoeuvre to the next left lane

The following equations (2.30) and (2.31) show, how the current lateral position during a lane change manoeuvre is calculated.

1. Lane change manoeuvre from a lateral starting position $y(t = 0) = y_0$ to the next left lane

$$y(t) = \begin{cases} y_0 & \text{for } t = 0 \\ y_0 - \left(-\frac{6 \cdot h}{t_m^5} \cdot t^5 + \frac{15 \cdot h}{t_m^4} \cdot t^4 - \frac{10 \cdot h}{t_m^3} \cdot t^3 \right) & \text{for } 0 < t \leq t_m \\ y_0 + h & \text{for } t_m < t \end{cases} \quad (2.30)$$

2. Lane change manoeuvre from a lateral starting position $y(t = 0) = y_0$ to the next right lane

$$y(t) = \begin{cases} y_0 & \text{for } t = 0 \\ y_0 + \left(-\frac{6 \cdot h}{t_m^5} \cdot t^5 + \frac{15 \cdot h}{t_m^4} \cdot t^4 - \frac{10 \cdot h}{t_m^3} \cdot t^3 \right) & \text{for } 0 < t \leq t_m \\ y_0 - h & \text{for } t_m < t \end{cases} \quad (2.31)$$

The HWC should be a comfort system. In this thesis a fixed polynomial is used to generate the trajectory, needed for performing a lane change manoeuvre. As shown with the equations (2.30) and (2.31), the shape of this polynomial is defined by a specific manoeuvre duration t_m and the maximum possible lateral displacement h , which equals the width of the lane. Hence the width of the lane is predefined, the generated trajectory can only be influenced by the parameter of the manoeuvre duration. The shape of the trajectory also defines the occurring maximum lateral acceleration.

In [20] Samiee shows a summary of different studies regarding lane change durations. According to this summary an average lane change duration of $t_m = 4$ s was chosen. With this configuration, a maximum lateral displacement of $|h| = 3.5$ m and a maximum longitudinal velocity of $v_x = 130$ km/h, a maximum lateral acceleration of $a_y < 1.2$ m/s² is achieved, if the lane change manoeuvre is performed on a straight road.

At lower speeds, smaller values of t_m would also deliver good results, but a variation of the the parameter t_m was not considered during the implementation of the HWC.

2.3. Creation of a new HWC model and integration of different sensor models

In this section, the creation process of the new HWC model with the integration of different sensor types and their suitability for the model are described.

2.3.1. Phenomenological RADAR sensor model

The first idea was to use a consisting phenomenological RADAR sensor model which was implemented in Simulink by Zoltan Magosi during his Master Thesis [15]. Besides the usage of geometric parameters, also physical influencing properties (propagation of electromagnetic waves) are considered in this model. This sensor model is able to detect objects and to measure their kinematic quantities in real time. It consists of three main parts. The first subsystem is the Geometric Model which deals with geometric quantities. In the second subsystem, Real Performance, the physical properties of real RADAR sensors are taken into account and the *Signal to Noise Ratio* (SNR) is replicated. Weather as an influencing factor is modelled by using signal interruptions. Then a noisy sensor signal is generated and filtered by means of a Kalman - Filter. In the last subsystem a method for path prediction is used to find the relevant target for ACC.

Using such phenomenological sensor models in the early concept phase has a many advantages. For example, they are more time efficient than physical sensor models and they are very easy to parametrize, while also considering real effects like signal losses, time delays or noisy signals etc. [2].

In Magosi's phenomenological model, the needed data for the internal sensor calculation was also gathered by directly addressing each vehicle. A disadvantage of directly accessing the data of the targets is, that each vehicle in the simulation has to be addressed with its name or ID, which makes the model inflexible. This sensor model was implemented to work with a maximum of four targets. Despite of this disadvantage, a few approaches were made with this sensor model, but the result was, that this method is not suitable for simulations with a large amount of targets.

To avoid the before mentioned problem, different sensor models of IPG CarMaker[®] were investigated according to their usability and suitability for the HWC model. At this point two different sensor types were used, described in the following sections 2.3.2 and 2.3.3.

2.3.2. Ideal Sensors

CarMaker[®]'s Ideal Sensors are simple sensor models, used for object tracking and environment detection. The information provided by this sensor type is independent from the real life sensor technology, which means no physical properties are taken into account. In general, Ideal Sensors are used in an early stage of function development [13].

2.3.2.1. Object Sensor

CarMaker[®]'s Object Sensor is one of the Ideal Sensors. The following items show the available quantities in the object list according to the Reference Manual [12]; figure 2.11 illustrates a few of these quantities.

2. Methodology

- Object ID
- Bearing (reference point and nearest point)
- Relative distance and velocity (reference point and nearest point)
- Relative orientation z-y-x (reference point)
- Distance in x, y, z of sensor frame (reference point and nearest point)
- Velocity in x, y, z of sensor frame (reference point and nearest point)
- Flag: object is detected (in sensor viewing area)
- Flag: object is in observation area
- Incidence angles between the sensor beam and the object
- Width, height, length of object and height above ground (only in C-code)
- Object name (only in C-code)

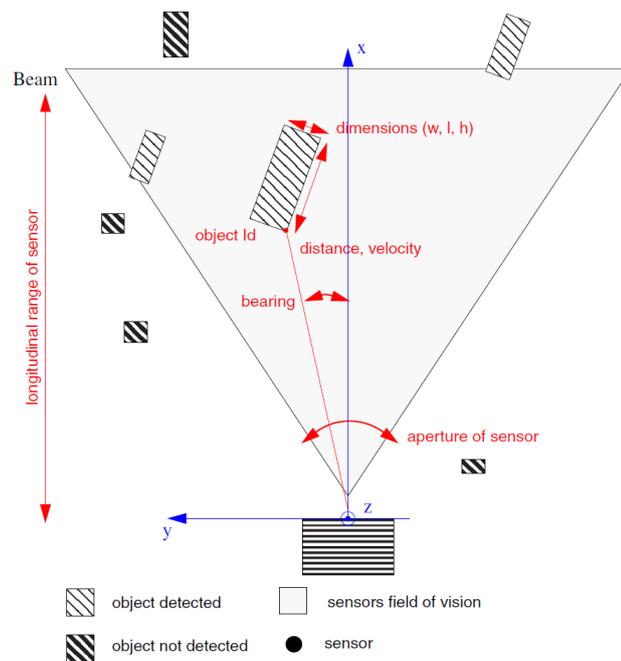


Figure 2.11.: Quantities of CarMaker[®]'s Object Sensor [12]

With the calculation class setting “Nearest Point” and the target selection mode “Nearest” the Object Sensor model provides the quantities of the the nearest detected object point, depicted in figure 2.11 . There is also a target selection mode “Nearest in Path” available. This method is based on estimating the trajectory of the ego to find the nearest target on the same lane, relevant for ACC.

Different approaches and tests were made with this type of sensor. A big advantage compared to the phenomenological sensor of section 2.3.1 is, that the amount of targets is not limited. However, one disadvantage of this sensor is, that no quantity for the acceleration is available, which means that the modified approach of the TTC calculation can not be used. In addition to that, it is not possible to get the data of all detected targets and search for the relevant one with an own algorithm. The conclusion was, that without a C-code modification, this sensor model is not suitable for the use in the HWC. Before modifying this type of sensor, a High Fidelity Sensor model was investigated, described in the following section

2.3.3. High Fidelity Sensors

High Fidelity (HiFi) Sensors are more advanced than the Ideal Sensors. Basic technology specific effects are considered in this sensor type.

2.3.3.1. RADAR Sensor

IPG CarMaker[®]'s RADAR HiFi Sensor tries to replicate the behaviour of a real RADAR sensor. The RADAR HiFi model provides an object list including quantities regarding detection, kinematics, classification and ground truth information of each detected target. The kinematic quantities of the detected targets are relevant and needed for the implementation of the HWC. The following items give an overview of these quantities according to the Reference Manual [12]:

- Position (relative distance, relative longitudinal distance, relative lateral distance, relative vertical distance)
- Velocity (relative speed, relative longitudinal speed, relative lateral speed)
- Acceleration (relative longitudinal acceleration)
- Orientation (relative course angle calculated from the measured velocity)

Due to the fact, that all quantities of each detected target are available and the modified TTC approach is possible, it was decided to use this sensor model instead of using the simple Object Sensor with a C-code modification. After the determination of a suitable sensor model for target detection, a method for assigning the detected targets to the related lanes was needed. The following section 2.4 explains an approach, used to assign the targets to the lanes.

2.4. Determination of the target position

In this section, an appropriate method for the determination of the target positions on the road relative to ego vehicle is searched. ACC for example uses available sensor data

(data of ESP related sensors, data of RADAR sensor etc.) to predict the trajectory of the ego and find the relevant target to follow.

Winner et al. describes in [10] several path prediction methods. These approaches use the information of the road curvature. The curvature κ equals the reciprocal of the radius R ($\kappa = 1/R$), the distance between the instantaneous centre of rotation and the center of gravity of the vehicle. To avoid calculation problems if driving on a straight road (in this case the radius would run towards infinity), the curvature is preferred to the radius.

According to Winner [10] there are four possible ways to determine the curvature:

1. curvature determination by means of the steering angle
2. curvature determination by means of the yaw rate
3. curvature determination by means of the lateral acceleration
4. curvature determination by means of the wheel speed

The results of these curvature determination methods depend on different factors like wind from the side, the slope of the road, the speed range and so on. The calculation of the curvature via the yaw rate delivers in most of the cases good results. A big disadvantage of these methods is, that they are only suitable, if the ego remains on one lane. This means, for ADAS like ACC they can be used. During a lane change manoeuvre the measured values like lateral acceleration, yaw rate etc. are increased due to the lateral driving manoeuvre and a wrong path is predicted. Hence a HWC combines different ADAS including a LCA, a different approach which is also suitable for lane change manoeuvres has to be used.

One possibility would be using an image processing system for the detection of road markings. IPG CarMaker[®] provides an Ideal Sensor model called Line Sensor. With this idealized camera, traffic barriers or road markings can be detected. Unfortunately the coordinates of the detected line points are not included in the standard output quantities of this sensor model. By means of a C-code interface new *User Accessible Quantities* (UAQ) for a desired amount of points can be created.

In figure 2.12 two vehicles, the ego and a target are depicted. The ego is equipped with two sensors facing forwards. The RADAR sensor (RADAR HiFi) determines the relative longitudinal and lateral distance between the ego and the target and the camera (Line Sensor) is used to detect the first road marking on the left and the right side of the ego.

2.4.1. Method for assigning the detected targets to the according lanes

Hence the standard output quantities of the Line Sensor does not include information about the position of the detected lines in the sensor coordinate system, new UAQ for the detected lines were generated by using the C-code interface. The idea is to approximate the actual course of the detected road markings by using straight line segments. For

this reason, each detected line is divided into a point list. For each point at a specific s coordinate (the s coordinate points into longitudinal direction of the Link and is located on the Reference Line of the Link) UQA for the position in the sensor coordinate system x_s, y_s , shown in figure 2.12, were created.

Figure 2.12 illustrates a situation, where a target is driving on the left lane in front of the ego. The target is detected by the RADAR HiFi sensor. The road markings, depicted as dashed lines, are detected via IPG CarMaker[®]'s Line Sensor, a idealized camera model. The dots on the dashed lines are symbolizing the generated UAQ of the line points, needed for the linear approximation of the road marking. The FOV of both sensors are indicated by triangles.

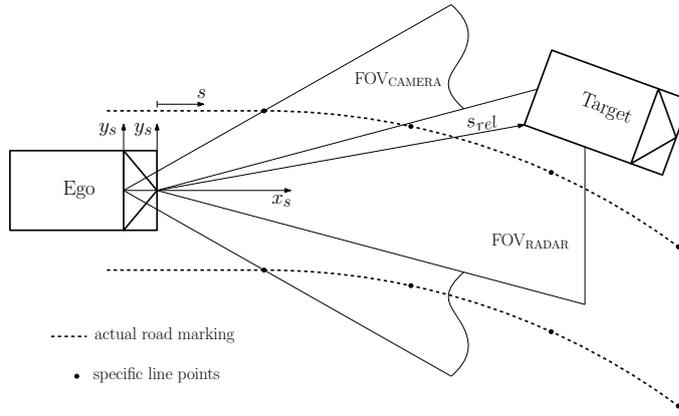


Figure 2.12.: Assignment of detected target to the according lane

Zooming into the area of the target provides the detailed depiction 2.13. In this figure, the rear of the target and the approximated road marking is illustrated.

The following items describe the variables and parameters used in figure 2.13:

- Axes x_s, y_s
are the axes of the sensor coordinate system
- Point R
is the nearest point of the target, with the coordinates $s_{x,rel}$ and $s_{y,rel}$ (Note: according to CarMaker[®]'s Reference Manual [12], the reference point detected by the RADAR HiFi sensor is located in the target's bounding box centre)
- Point P_1
is the first adjacent line point with the coordinates x_1 and y_1 , relevant for the calculation

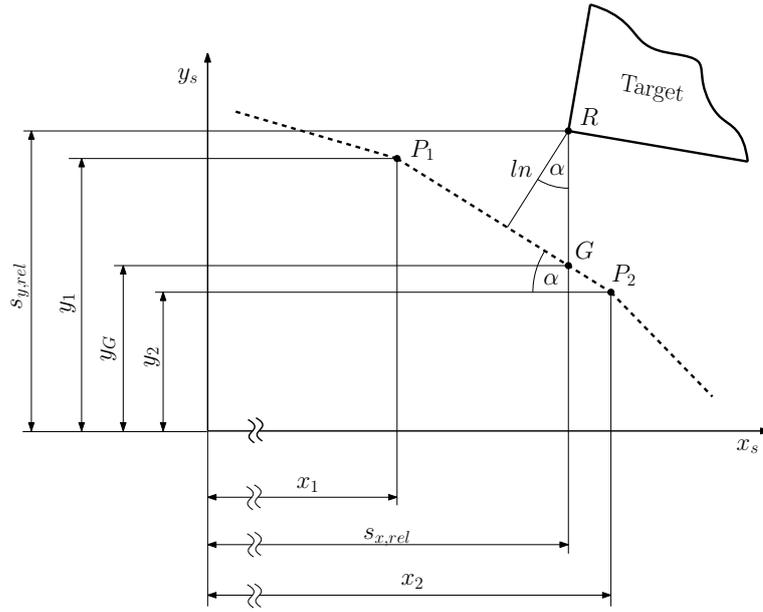


Figure 2.13.: Assignment of detected target to the according lane (detail)

- Point P_2
is the second adjacent line point with the coordinates x_2 and y_2 , relevant for the calculation
- Point G
is a point, positioned on the connecting line between P_1 and P_2 . Point G has the same x coordinate as the point R ($s_{x,rel}$) and the y coordinate y_G
- Distance ln
is the distance between R and the connecting line of P_1 and P_2 , measured perpendicular
- Angle α
is the angle between the connecting line of P_1 and P_2 and x_s

By means of similar triangles, the y position y_G of the limit value (equation (2.33)), the angle α (equation (2.35)) and the perpendicular distance ln (equation (2.36)) can be calculated.

$$\frac{y_1 - y_2}{x_2 - x_1} = \frac{y_G - y_2}{x_2 - s_{x,rel}} \quad (2.32)$$

$$y_G = \frac{y_1 - y_2}{x_2 - x_1} \cdot (x_2 - s_{x,rel}) + y_2 \quad (2.33)$$

$$\tan \alpha = \frac{y_1 - y_2}{x_2 - x_1} \quad (2.34)$$

$$\alpha = \arctan \frac{y_1 - y_2}{x_2 - x_1} \quad (2.35)$$

$$\cos \alpha = \frac{ln}{s_{y,rel} - y_G} \quad (2.36)$$

$$ln = \cos \alpha \cdot (s_{y,rel} - y_G) \quad (2.37)$$

By means of the lengths $s_{y,rel}$, y_G and ln , the position of the targets on the road can be determined.

2.5. HWC model in Simulink

In this section, the structure of the HWC model implemented in MATLAB/Simulink is described. The main parts of the subsystem HWC are:

1. Subsystems and MATLAB function blocks concerning the front of the ego
 - a) Data RADAR sensor (front)
 - b) Assignment target to lane (front)
 - c) Equation of motion / TTC (front)
 - d) Smallest values of targets (front)
2. Subsystems and MATLAB function blocks concerning the rear of the ego
 - a) Data RADAR sensor (rear)
 - b) Assignment target to lane (rear)
 - c) Equation of motion / TTC (rear)
 - d) TTC of nearest targets (rear)
3. Decision making LC left
4. Decision making LC right
5. Hold state (Stateflow Chart)
6. Time during LC manoeuvre to the left
7. Lateral offset calculation LC to the left
8. Time during LC manoeuvre to the right
9. Lateral offset calculation LC to the right

2.5.1. Subsystems and MATLAB function blocks concerning the front of the ego

This section contains a detailed description of the functionality of the subsystems concerning the front area of the ego.

2.5.1.1. Data RADAR sensor (front)

This subsystem is used to gather the relative longitudinal distance, relative longitudinal speed and relative longitudinal acceleration of the targets in front of the ego, detected by a RADAR HiFi sensor. Same quantities are written into a vector. The amount n of blocks of each quantity determines the amount of objects handled in the RADAR sensor's object list. If more than n objects are positioned in the RADAR sensor's FOV, the closest n objects are used to fill the object list.

2.5.1.2. Assignment target to lane (front)

This subsystem consists of three more subsystems and a MATLAB function block:

1. Line sensor front: line points from left road marking

includes the UAQ (x and y coordinates in the line sensor coordinate system) of the line points from the closest detected road marking on the left side (positive y direction in sensor coordinate system) in front of the ego; created via the C-code interface.

Note: consider an offset in x direction if the coordinate systems of the line sensor and the RADAR sensor do not have the same origin.

2. Line sensor front: line points from right road marking

includes the UAQ (x and y coordinates in the line sensor coordinate system) of the line points from the closest detected road marking on the right side (negative y direction in sensor coordinate system) in front of the ego; created via the C-code interface.

Note: consider an offset in x direction if the coordinate systems of the line sensor and the RADAR sensor do not have the same origin.

3. RADAR sensor front: components of relative distance

includes the x and y components of the relative distance of each target in the object list of the RADAR sensor, mounted at the ego's front.

4. Assignment target to lane front

includes the calculation procedure to determine on which lane the targets, detected in front of the ego, are driving. The figures 2.12 and 2.13 visualize this process. In

the MATLAB function block the two adjacent line points P_1 and P_2 of the detected target point R are searched. This is done by comparing the RADAR sensor's x component of the relative distance to the x coordinates of the line points. With the x and y coordinates of the relevant line points P_1 and P_2 the equations (2.33) and (2.37) can be used to calculate the the values y_G and the distances ln related to the left and right line segment of the road marking. Each detected target is labelled with a specific number according to the length of ln . With this *marker*, the detected target can be assigned to a specific lane relative to the ego, depicted in figure 2.14.

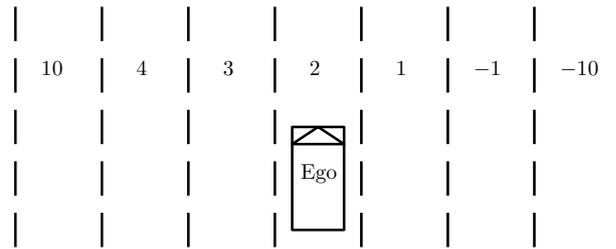


Figure 2.14.: Markers for assigning the detected targets to the according lanes

The following points explain the meaning of the used numbers:

- $marker = 10 \implies$ detected target is positioned on the left side of the ego, but is more than two lane widths away
- $marker = 4 \implies$ detected target is driving on the second lane to the left of the ego
- $marker = 3 \implies$ detected target is driving on the first lane to the left of the ego
- $marker = 2 \implies$ detected target is driving on same lane as the ego
- $marker = 1 \implies$ detected target is driving on the first lane to the right of the ego
- $marker = 0 \implies$ no target detected
- $marker = -1 \implies$ detected target is driving on the second lane to the right of the ego
- $marker = -10 \implies$ detected target is positioned on the right side of the ego, but is more than two lane widths away

The calculated *marker* values are written into a vector **marker** according to the order of the object list.

2.5.1.3. Equation of motion/TTC (front)

In this subsystem the TTC of all targets of the object list are calculated using the equations (2.14) and (2.19) described in section 2.2.1. In this version of the HWC, no TTC is calculated during a lane change manoeuvre. The TTC to each target is permanently determined until the conditions for a lane change manoeuvre are appropriate and the manoeuvre is initiated. In the modified TTC approach with two real solutions (result under the square root ≥ 0 , see equation (2.20)) the smallest positive value is the relevant one. In case of two negative TTC values, the bigger one is chosen. All calculated TTC values are written into a vector $\mathbf{TTC}_{\text{all}}$ according to the object list.

2.5.1.4. Smallest values of targets (front)

If more than one target is detected on the same lane, normally the quantities of the closest target are seen as relevant. The idea however was to test a different approach in this thesis, where also the quantities of the other detected targets on the same lane are considered. Figure 2.15 is used to explain the concept.

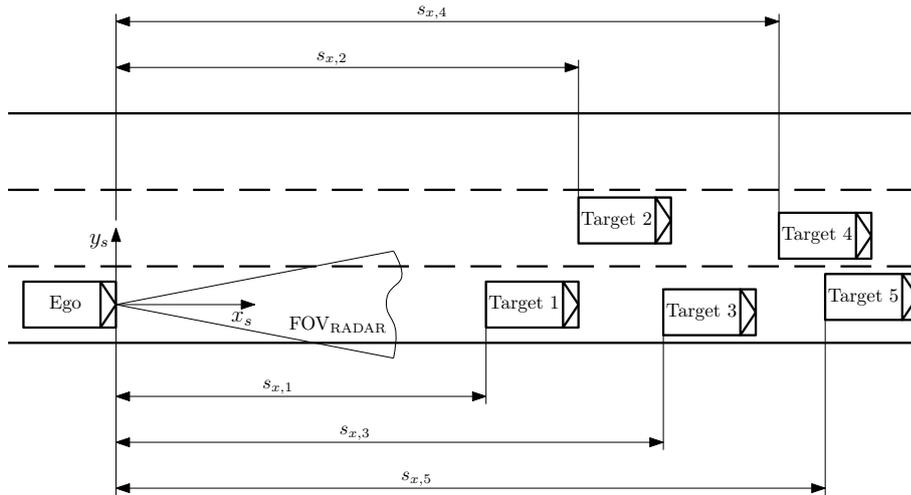


Figure 2.15.: Finding the smallest values of the detected targets (front)

In figure 2.15, a case is illustrated, where the ego detects five targets in front via a RADAR sensor. Due to multipath propagation, real high frequency RADAR sensors are able to detect even objects where the sight is obstructed. Target 1, target 3 and target 5 are driving on the same lane as the ego. Target 2 and target 4 are positioned on the next left lane. In this thesis the smallest TTC value of each lane is considered as the relevant one and used for the decision making. In other words, if the TTC of target 5 is the smallest of the three calculated values on rightmost lane, the TTC of target 5 is considered as the relevant one of this lane.

Only focusing on the closest targets could lead to dangerous situations, described in the following scenario. Target 2 and target 4 are overtaking all vehicles of the rightmost lane. The ego and target 1 are driving with the same speed, so a lane change manoeuvre of the ego is not necessary. Target 1 is approaching target 3 with full speed. Target 3 is driving much slower or is even decelerating due to target 5. If target 1 does not decelerate due to target 3 (too slow reaction of the driver or system failure of target 1) and maintains the actual speed, the ego would keep following target 1 until target 1 crashes into the rear of target 3. Depending on the distance between ego and target 1 it can happen, that at this point a safe lane change or breaking manoeuvre without crashing is not possible.

Using the method of this thesis and choosing the smallest TTC of each lane as the relevant criteria for decision making could help to react earlier to critical situations. The subsystem “smallest values of targets (front)” is used to determine the smallest TTC of each lane.

At the beginning, all TTC values stored in one vector ($\mathbf{TTC}_{\text{all}}$) are separated according to their marker values and written into new vectors $\mathbf{TTC}_{\mathbf{n}}$ for each lane. “ \mathbf{n} ” is a place holder for the different indices used to describe the related lane.

The vector $\mathbf{s}_{\mathbf{x},\text{all}}$ contains the x components of all relative distances and is also split into several vectors $\mathbf{s}_{\mathbf{x},\mathbf{n}}$. In addition to that the x components of all relative velocities stored in $\mathbf{v}_{\mathbf{x},\text{all}}$ are separated according to the marker values as describe before and written into the new vectors $\mathbf{v}_{\mathbf{x},\mathbf{n}}$. A detailed description of these vectors is attached to the Appendix A.

In front only TTC values bigger or equal to zero make sense and are relevant. A negative TTC in front indicates, that the target is driving faster as the ego or that due to a high acceleration no accident will occur. For this reason, the smallest positive TTC on each lane is searched. If no target is detected on a lane or if no positive TTC value is calculated a $TTC = 0$ is used as a relevant value for this lane.

In the Simulink model the smallest TTC value of each lane is written into a vector $\mathbf{TTC}_{\text{min}}$. The Appendix A includes an explanation where the time value of each lane is placed in this vector.

In addition to that, also the x component of the relative distance and the related relative longitudinal velocity of the closest target on each lane is searched and written into the vectors $\mathbf{s}_{\mathbf{x},\text{min}}$ and $\mathbf{v}_{\mathbf{x},\text{min}}$ (description attached to Appendix A). These quantities are needed for the ACC controller.

2.5.2. Subsystems and functions concerning the rear of the ego

Although the subsystems concerning the rear area of the ego are quite similar to those of section 2.5.1, they are described in this section for the sake of completeness.

2.5.2.1. Data RADAR sensor (rear)

This subsystem is used to obtain the x components of relative longitudinal distance, relative longitudinal speed and relative longitudinal acceleration of the targets positioned in the rear area of the ego. A HiFi RADAR sensor is used for target detection. As described in 2.5.1.1 the amount of blocks of each quantity defines the number of objects in the object list.

2.5.2.2. Assignment target to lane (rear)

This subsystem includes three more subsystems and a MATLAB function block:

1. Line sensor rear: line points from left road marking

includes the UAQ (x and y coordinates in the line sensor coordinate system) of the line points from the closest detected road marking on the left side (positive y direction in sensor coordinate system) at the ego's rear; created via the C-code interface.

Note: consider an offset in x direction if the coordinate systems of the line sensor and the RADAR sensor do not have the same origin.

2. Line sensor rear: line points from right road marking

includes the UAQ (x and y coordinates in the line sensor coordinate system) of the line points from the closest detected road marking on the right side (negative y direction in sensor coordinate system) at the ego's rear; created via the C-code interface.

Note: consider an offset in x direction if the coordinate systems of the line sensor and the RADAR sensor do not have the same origin.

3. RADAR sensor rear: components of relative distance

includes the x and y components of the relative distance of each target in the object list of the RADAR sensor mounted at the ego's rear.

4. Assignment target to lane rear

includes the calculation procedure to determine on which lane the targets, detected in the rear area of the ego, are driving. The used variables are depicted in the figures 2.12 and 2.13. The calculation procedure is the same as in front, see section 2.5.1.2.

Note: The rear sensors are rotated by 180° . This means, that left (positive y direction) and right (negative y direction) in the rear sensors' coordinate system are reversed to the left and right in the front sensors' coordinate system. This is already considered in the assignment method. The marker values correspond to the coordinate system pointing in driving direction. Figure 2.16 depicts this situation.

The calculated marker values are also written into a vector **marker** according to the order of the object list.

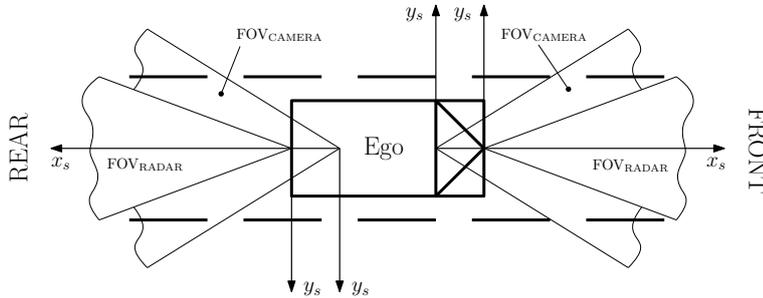


Figure 2.16.: Ego with front and rear sensors

2.5.2.3. Equation of motion/TTC (rear)

In this subsystem concerning the rear of the ego, the TTC of all targets of the object list are calculated using the equations (2.14) and (2.19) described in section 2.2.1. In this version of the HWC, no TTC is calculated during a lane change manoeuvre. The TTC to each target is permanently determined until the conditions for a lane change manoeuvre are appropriate and the manoeuvre is initiated. In the modified TTC approach with two real solutions (result under the square root ≥ 0 , see equation (2.20)) the smallest positive value is the relevant one. In some cases also negative TTC values are considered. A negative TTC means, that an accident would have happened at the calculated time in the past and is also used as a criteria at the backside of the ego for decision making; for example: if the ego is driving faster and overtaking a target, the negative TTC at the rear of the ego corresponds to a distance, which the ego is in front of the target. If both solutions of the modified TTC approach are real but smaller than zero, the bigger value is considered. The TTC values of all targets in the object list are stored in a vector named **TTC_{all}**.

2.5.2.4. TTC of nearest targets (rear)

Compared to the method for finding and selecting the relevant Target in the front area of the ego, described in section 2.5.1.4, the selection technique at the ego's rear is different. In this case the conventional approach is used, where the TTC of the closest target (smallest x component of relative distance detected by the RADAR sensor, mounted at the ego's rear side) of each lane are considered as relevant. Similar to section 2.5.1.4, the TTC values of all targets stored in one vector have to be separated according to the marker values. Hence only the targets on the left and the right lane behind the ego are considered in the decision making algorithm, only two vectors **TTC_n** for the TTC values and two vectors for the x component of the relative distances **s_{x,n}** are needed. As

explained before, “**n**” is a place holder. The vector \mathbf{TTC}_{\min} includes the TTC values of the nearest target of each lane. A detailed description can be found in the Appendix A.

2.5.3. Decision making LC left

This MATLAB function block is the the first part of the decision making algorithm and includes the logic of the HWC concerning when to perform a lane change manoeuvre to the next left lane as well as the selection of a target to follow for the ACC controller, if no lateral driving manoeuvre is possible. Targets in front of the ego on the same lane as well as on the first and second lane to the left are considered in the decision making process. This means, targets with marker values of 2, 3 and 4 are included. Due to complexity reasons, targets positioned on the left side of the ego but more than two lane widths away (marker value of 10) are not considered. In figure 2.17 a scenario with eight targets in front of the ego is illustrated. Table 2.1 gives information of which targets are considered in decision making algorithm for a lane change manoeuvre to the next left lane.

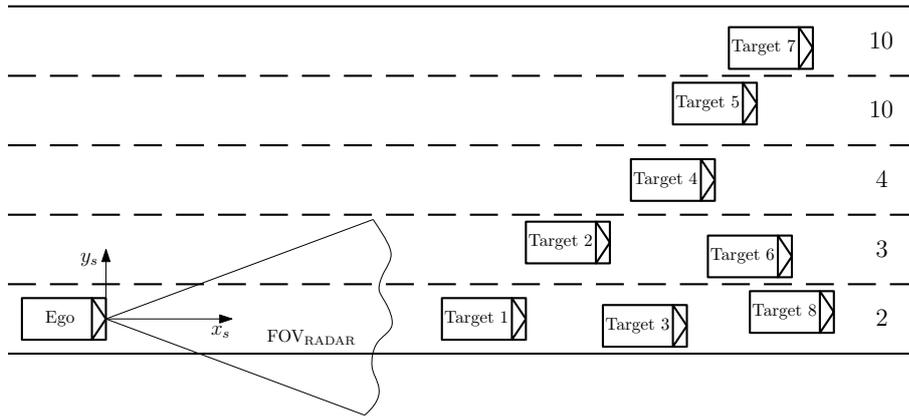


Figure 2.17.: Target consideration in the decision making algorithm for a lane change manoeuvre to the next left lane

Table 2.1.: Considered targets in the scenario depicted in figure 2.17

Target 1	Target 2	Target 3	Target 4	Target 5	Target 6	Target 7	Target 8
✓	✓	✓	✓	x	✓	x	✓

Note: The amount of considered targets with marker values 2, 3 and 4 depends on the length of the RADAR sensor’s object list.

The idea of this decision making algorithm is to initiate a lane change manoeuvre to the next left lane, if a slower or decelerating target is driving in front of the ego on the same lane ($marker = 2$) or on the next lane to the left of the ego ($marker = 3$). If a overtaking

manoeuvre with full speed is not possible, the speed of the ego should be reduced until a lane change manoeuvre can be initiated. In case of a slower or decelerating target on the second lane to the left of the ego ($marker = 4$), this target should be used as the target to follow for ACC to prevent overtaking on the right side of the road.

The main criteria in this MATLAB function block is the calculated TTC value in front of the ego. The classification of the decisions is based on the comparison of the smallest TTC of each lane. If a case is determined via these TTC values, also information from other sensors is used before an action is performed.

Note 1: The vector $\mathbf{TTC}_{\min,f}$ contains the smallest TTC values of each lane in front of the ego (equals the vector \mathbf{TTC}_{\min} calculated in the subsystem “smallest values of targets (front)” described in section section 2.5.1.4).

Note 2: The vector $\mathbf{TTC}_{\min,r}$ contains the TTC values of the nearest targets of the left and right lane behind the ego (equals the vector \mathbf{TTC}_{\min} calculated in the subsystem “smallest values of targets (rear)” described in section 2.5.2.4).

In figure 2.18 the ego with the relevant sensors for the decision making process is illustrated. On the left and the right side of the ego IPG CarMaker[®]'s Object Sensor is used to check, if there is a target in the FOV or if the area beside the ego is free.

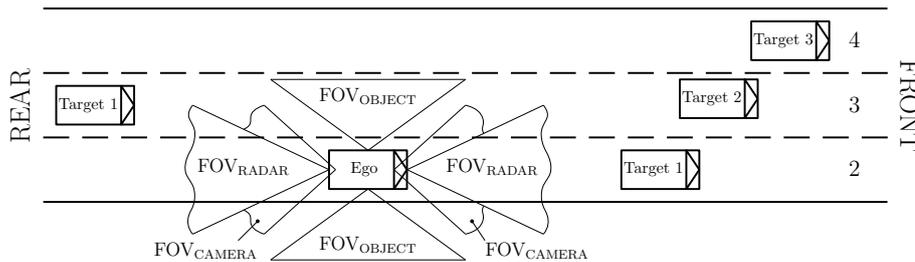


Figure 2.18.: Sensors used for decision making

After a situation is classified by comparing the smallest TTC values of the three relevant lanes, it is checked, if the calculated TTC values are over a specific upper time limit (TTC_{ul}), between this upper time limit and a specific lower time limit (TTC_l) or if the values are below the lower limit. The upper time limit indicates the earliest possible time, the lower time limit defines the latest possible time for a lane change manoeuvre.

The following items describe the meaning of these time limits:

- $TTC > TTC_{ul}$ no lane change manoeuvre is necessary at this point
- $TTC_l < TTC < TTC_{ul}$ lane change manoeuvre is possible
- $TTC < TTC_l$ no safe lane change manoeuvre is possible

The lower time limit depends on the manoeuvre duration t_m which is a configurable parameter and on the situation (lane change manoeuvre due to target in front and on same lane or due to target in front and on first lane to the left of the ego). At the end of

the lane change manoeuvre ($t = t_m$, see equation 2.30 in section 2.2.2) the polynomial provides the maximum lateral displacement. This means, that this point has to be reached before the x component of the relative distance is zero.

In case of a lane change manoeuvre due to a target on the same lane, the lower time limit has to be bigger than than the manoeuvre duration t_m to perform a safe lane change manoeuvre. Figure 2.19 illustrates this situation.

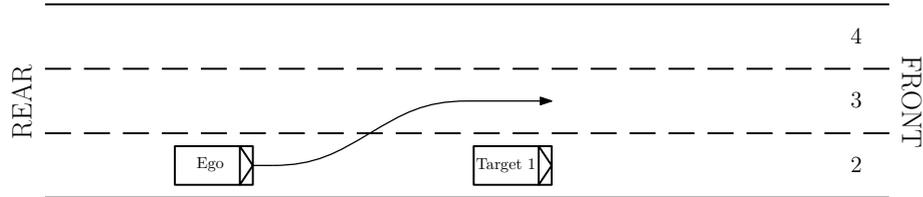


Figure 2.19.: Lane change manoeuvre due to target on the same lane

However, if a lane change manoeuvre is initiated due to a target on the first lane to the left of the ego, the lower time limit has to be bigger than in the first case. Due to the fact, that in this situation actually two lane change manoeuvres have to be done in a row, the lower time limit has to be at least twice as high as the lower time limit of a single lane change manoeuvre. Figure 2.20 illustrates a lane change manoeuvre due to a target on the first lane to the left of the ego. This lateral driving manoeuvre is separated in two subsequent lane changes. It can be seen, that the trajectory remains constant on the first left lane for a certain time. This indicates, that the first lateral driving manoeuvre was finished a certain time before the upper time limit regarding the target on the same lane was reached. Theoretically, it is possible to modify the lateral driving manoeuvre consisting of two separate lane changes into a single manoeuvre, however this was not part of this thesis.

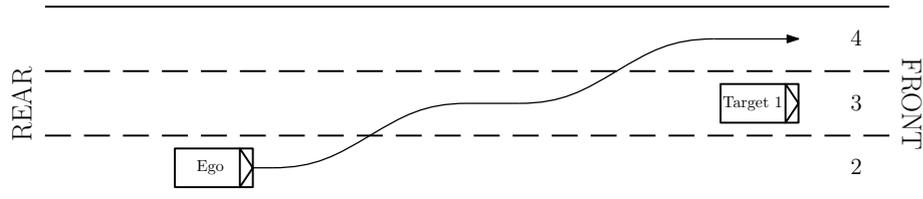


Figure 2.20.: Lane change manoeuvre due to target on first lane to the left of the ego

The upper time limit is freely selectable in both cases, but has to be bigger than the lower time limits.

Note 1: If a TTC value is compared to a time limit, always the time limit of the related lane is meant. For example: if $TTC_{min,f,s}$ (smallest TTC in front on the same lane) is compared to a time limit, the time limit of the same lane as the ego is driving on is used. However if $TTC_{min,f,1l}$ (smallest TTC in front on the first lane to the left) is compared to a time limit, the time limit of the first lane to the left of the ego is meant.

In case of driving on the leftmost lane and a target is detected in front, this target is used as the target to follow for ACC. If the ego is not positioned on the leftmost lane, decisions when to initiate a lane change manoeuvre or when to activate ACC are made. The following sections describe how decisions are made.

2.5.3.1. Smallest TTC on the same lane

In this situation is $TTC_{min,f,s} < TTC_{min,f,1l}$.

1. $TTC_{min,f,s} = 0$

\implies no target is detected on the same lane or no positive TTC is calculated on this lane. In this case the target on the first lane to the left of the ego is the relevant one for the lane change manoeuvre.

Case 1: $TTC_{min,f,1l} > TTC_{ul}$

In this case no lane change manoeuvre has to be performed.

Case 2: $TTC_{ll} < TTC_{min,f,1l} < TTC_{ul}$

If also the left area of the ego is free (nothing detected by the Object Sensor) and if there is a safe distance to the nearest target on the first left lane behind the ego ($TTC_{min,r,1l}$ is outside the time limits: upper time limit has to be equal or bigger than the upper time limit in front on the first left lane, lower time limit has to be smaller or equal to zero meaning no target behind or no accident will occur) a lane change manoeuvre to the next left lane can be initiated.

Case 3: Conditions of Case 1 or 2 are not fulfilled

In a situation where the the conditions of Case 1 or Case 2 can not be fulfilled, the ACC is activated with the closest target on the first left lane as the target to follow until a lane change manoeuvre can be done.

2. $TTC_{min,f,s} > 0$

\implies positive TTC value is calculated on the same lane as the ego is driving on.

In this case it is checked, if the $TTC_{min,f,s}$ is bigger than the upper time limit, smaller than the upper and bigger than the lower time limit or smaller than the lower time limit of the same lane.

Case 1: $TTC_{min,f,s} > TTC_{ul}$

In this situation the smallest TTC value is on the same lane, but no lane change manoeuvre due to his target is necessary. For this reason, the target on the first lane to the left of the ego has to be investigated. If also $TTC_{min,f,1l} > TTC_{ul}$, no lane change manoeuvre is necessary. However, if the the conditions for a lane change manoeuvre are fulfilled ($TTC_{min,f,1l}$ between the time limits, no target is detected by the Object Sensor on the left side of the ego and there is a safe distance

to the target behind the ego on the first left lane ($TTC_{min,r,1l}$ outside the time limits) a lane change manoeuvre to the next left lane is possible. In case, one of these conditions is not reached, the closest target on the left side is chosen as relevant target for the ACC until a lane change manoeuvre can be initiated.

Case 2: $TTC_{ll} < TTC_{min,f,s} < TTC_{ul}$

A lane change manoeuvre is only possible if there is also safe distance to the targets in front and rear on the first lane to the left of the ego ($TTC_{min,f,1l} > TTC_{ul}$; $TTC_{min,r,1l}$ outside the rear time limits) and no target is detected via the Object Sensor on the left side of the ego.

Case 3: Conditions of Case 1 or 2 are not fulfilled

In this situation the target on the same lane in front is chosen as relevant target for the ACC until a lane change manoeuvre is possible.

2.5.3.2. Smallest TTC on the first lane to the left of the ego

In this situation is $TTC_{min,f,s} > TTC_{min,f,1l}$.

1. $TTC_{min,f,1l} = 0$

\implies no target is detected on the first lane to the left of the ego or no positive TTC is calculated on this lane. In this situation the target on the same lane in front of the ego is the relevant one for the lane change manoeuvre.

Case 1: $TTC_{min,f,s} > TTC_{ul}$

At this point, no lane change manoeuvre due to the target on the same lane is necessary.

Case 2: $TTC_{ll} < TTC_{min,f,s} < TTC_{ul}$

If also the left area of the ego is free (nothing detected by the Object Sensor) and if there is a safe distance to the nearest target on the first left lane behind the ego ($TTC_{min,r,1l}$ is outside the time limits) a lane change manoeuvre to the next left lane can be performed.

Case 3: Conditions of Case 1 or 2 are not fulfilled

In this situation the target on the same lane in front is chosen as relevant target for the ACC until a lane change manoeuvre is possible.

2. $TTC_{min,f,1l} > 0$

\implies positive TTC value is calculated on the first lane to the left of the ego.

Case 1: $TTC_{min,f,1l} > TTC_{ul}$

In this situation no lane change due to the target on the first lane to the left of the ego is necessary.

Case 2: $TTC_{ll} < TTC_{min,f,1l} < TTC_{ul}$

If also the left area of the ego is free (nothing detected by the Object Sensor) and if there is a safe distance to the nearest target on the first left lane behind the ego ($TTC_{min,r,1l}$ is outside the time limits) a lane change manoeuvre to the next left lane is possible.

Case 3: Conditions of Case 1 or 2 are not fulfilled

In this situation the target on the first lane to the left of the ego is chosen as relevant target for the ACC until a lane change manoeuvre is possible.

2.5.3.3. Smallest positive TTC on the second lane to the left of the ego

If the smallest positive TTC value in front is located on the second lane to the left of the ego, the closest target on this lane is chosen as relevant target for the ACC to prevent overtaking on the right side.

Depending on the decision made in this MATLAB function block, different output quantities are available:

- $trigger_l$: used as a trigger in the Stateflow Chart described in section 2.5.5; can be bigger or equal to zero
- $relvTgt$: used for the logic of the ACC controller; can be smaller or bigger than zero
- $relvTgt_{dsx}$: relevant relative distance (x component) used for the ACC controller
- $relvTgt_{dvx}$: relevant relative velocity (x component) used for the ACC controller

2.5.4. Decision making LC right

In this MATLAB function block the second part of the decision making algorithm is implemented and includes the logic when to initiate a lane change manoeuvre to the next right lane. A lane change manoeuvre to the right is not possible, if the ego is driving on the rightmost lane. In all other cases, a lane change manoeuvre to the right is only initiated if the ego is driving with a velocity near the desired cruising speed and if the following conditions are fulfilled:

- No target is detected on same lane in front of the ego or the smallest TTC on this lane ($TTC_{min,f,s}$) is bigger than specific time limit or equal to zero.
- No target is detected on first lane to the right and in front of the ego or the smallest TTC on this lane ($TTC_{min,f,s}$) is bigger than specific time limit.
- No target is detected on the first lane to the right and behind of the ego or if the smallest TTC on this lane ($TTC_{min,r,1r}$) is bigger than an upper time limit or smaller than a lower time limit.

Note: The lower time limit has a negative value. This means that the accident would have occurred at this time in the past and corresponds to a distance which the ego is in front of a slower target.

- No target is detected via the Object Sensor on the right side of the ego.

Note: If a lane change manoeuvre to the left was done with a reduced velocity, the ego remains on this lane until the desired cruising speed is reached or if another lane change manoeuvre to the left is initiated.

Depending on the decision, a trigger value ($trigger_r$) bigger or equal to zero is generated which is used in the Stateflow Chart “hold state” described in the following section 2.5.5.

2.5.5. Hold state

“Hold state” is a Stateflow Chart and used to change between the states LC_l (lane change manoeuvre to the left), LC_r (lane change manoeuvre to the right) or $NoLC$ (no lane change manoeuvre). This chart is triggered by a decision of the decision making algorithm described in the sections 2.5.3 and 2.5.4. After a decision is made and the state is changed from $NoLC$ to one of the two states LC_l or LC_r , this state is hold for a duration that equals the the manoeuvre duration t_m , described in section 2.2.2.

2.5.6. Time during LC manoeuvre to the left

This subsystem is used to determine the time t , the variable in the polynomial for the lane change manoeuvre described in section 2.2.2.

2.5.7. Lateral offset calculation LC to the left

In this MATLAB function block the lateral displacement for a lane change manoeuvre to the left is calculated according to equation (2.30) described in section 2.2.2. The road is divided into lateral sections with the width of a lane. For each section the lateral offset can be calculated via the polynomial.

2.5.8. Time during LC manoeuvre to the right

The function of this subsystem is the same as described in section 2.5.6.

2.5.9. Lateral offset calculation LC to the right

Similar to the explanation in section 2.5.7, the lateral offset for a lane change manoeuvre to the right is calculated in this MATLAB function block. In this case, equation (2.31) of section 2.2.2 is used to generate the trajectory.

2.6. Simulation and Virtual Testing of the HWC Model

IPG's CarMaker[®] for Simulink is used to test the functionality of the HWC model, implemented in Simulink, in a virtual environment. At the beginning of the simulation the ego is placed on the rightmost lane of the virtual road using an initial lateral offset. The path or static course, from which the actual position (lateral offset) on the road is calculated, is located on the leftmost lane. In figure 2.21 the lateral offset ΔL from the static course is depicted. The lateral control of the ego concerning a defined course is performed by a model of CarMaker[®] called IPGDriver. This controller adjusts the steering wheel according to the lateral displacement, calculated in the Simulink model, from the static course and the current state of the vehicle. The gas and brake pedal positions are adjusted by using CarMaker[®]'s VehicleControl Model AccelCtrl with an ACC controller.

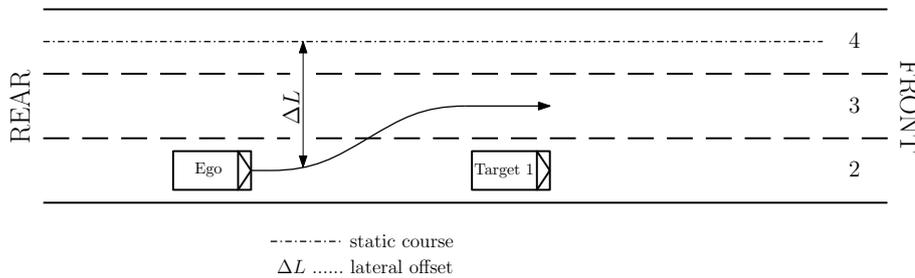


Figure 2.21.: Lateral offset from static course

At the beginning, simulations were made on simple roads (straights and curves) created in CarMaker[®]'s Scenario Editor. During these tests some limitation of the Line Sensor model were noticed.

The order of the Line Sensor's point list depends on the direction of a Link. There are two different possibilities:

1. The order of the point list is from the nearest to the farthest point
 - The Line Sensor points into the direction of the Link and the vehicle is also driving into this direction.
 - The Line Sensor points into the opposite direction of the Link and the vehicle is also driving into this direction.
2. The order of the point list is from the farthest to the nearest point
 - The Line Sensor points into the opposite direction of the Link, but the vehicle is driving into the direction of the Link.
 - The Line Sensor points into the direction of the Link, but the vehicle is driving into the opposite direction of the Link.

Line points are only available on a continuous line

1. Lane Sections

Road Markings are separated by Lane Sections. This means only line points of the Lane Section, in which the ego is positioned are available.

2. Ramps

Road markings are also separated by Ramps.

One detected issue in IPG CarMaker[®] 6.0.4 is, that it is not possible to draw continuous road markings across Lane Sections or Ramps. Due to the fact, that the converted and adjusted virtual road, described in section 2.1, has a lot of such Lane Sections and Ramps the method for the assignment of the detected targets to the according lanes does not work properly. According to IPG's technical support, this problem should be solved with a higher release.

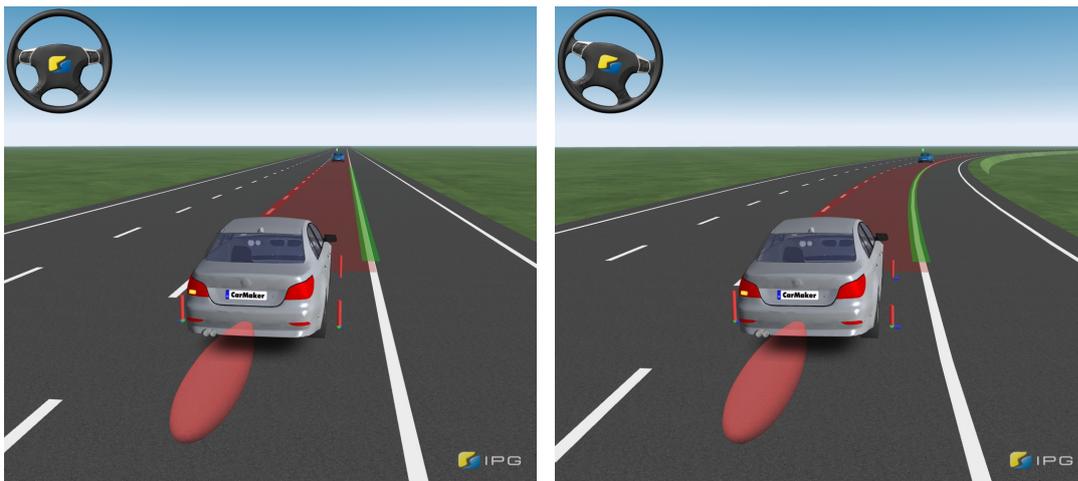
For this reason, the converted road file was not used for simulation in this thesis. Instead, simple roads with continuous road markings were created in CarMaker[®]'s Scenario Editor and used for testing the model.

3. Results and Discussion

3.1. Investigation of the polynomial approach

A simple overtaking manoeuvre (single lane change manoeuvre to the next left lane due to a target in front on the same lane and a lane change back to the right lane) is used to investigate, if the approach with the polynomial of fifth order described in section 2.2.2, originally intended for straight roads, can also be applied on curvy roads.

In figure 3.1 the same manoeuvre on two different roads (straight and curve) is illustrated. In both cases, the ego is driving with 130 km/h. The starting condition of the target is on the same lane 300 m in front of the ego. The target is driving with a constant speed of 80 km/h.



(a) Lane change manoeuvre on straight road

(b) Lane change manoeuvre on curvy road

Figure 3.1.: Comparison of a lane change manoeuvre on two different roads

Figure 3.2 shows the lateral displacement y according to the travelled distance s_x on a straight road at different velocities. Increasing the speed stretches the shape of the trajectory into longitudinal direction. It can be seen, that at a speed of 130 km/h the lane change manoeuvre requires 145 m until the maximum of the lateral displacement (equals the lane width) is reached and the manoeuvre is finished.

In figure 3.3 the longitudinal and lateral acceleration during the test manoeuvre on

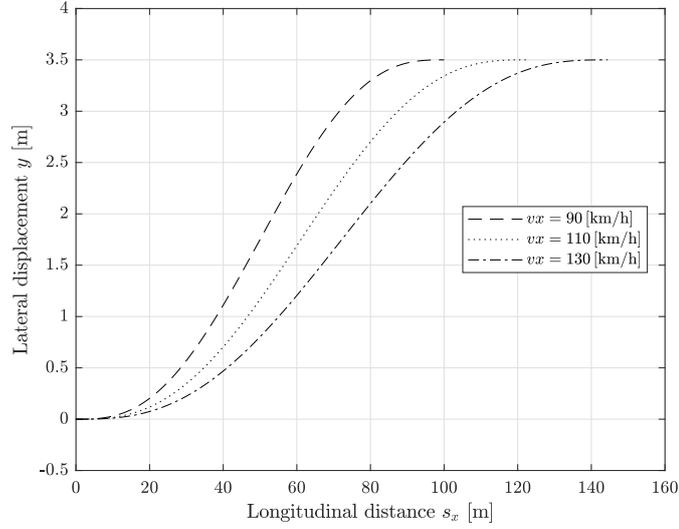


Figure 3.2.: Lateral displacement at different velocities (straight road)

the straight road is depicted. At the beginning of the simulation, the system requires approximately 2 s to reach a steady state. Then the ego is driving with a constant speed for the first 16 s until the calculated TTC is smaller than the upper time limit and a lane change manoeuvre to the next left lane is initiated.

After the lane change manoeuvre is finished ($t_m = 4$ s) and the maximum lateral displacement is reached, a certain time is needed until a steady state is reached again. This is the reason, why a path prediction approach based on the lateral acceleration or yaw rate can not be used. In addition to that, figure 3.3 illustrates that the ego is changing back to the right lane. This manoeuvre starts at approximately 24 s. It can also be seen, that the absolute value of the lateral acceleration is kept under 1.2 m/s^2 as required in section 2.2.2.

Figure 3.4 depicts the occurring lateral and longitudinal acceleration of the ego during the overtaking manoeuvre on the curvy road. Due to the fact, that the simulation already starts on the curve, more time (approximately 6 s) is needed to reach a steady state compared to the test on the straight road. At the beginning of the simulation, the target is also 300 m in front of the ego, which means there is enough space until the the RADAR sensor in front detects the target. A constant lateral acceleration is caused by the curvature of the road. From 6 s to 15 s the ego is driving on the rightmost lane behind the target with a lateral acceleration of about -2.6 m/s^2 . The mathematical sign of the acceleration value describes the direction of the curve and is in line with ISO 8855. In this case the minus indicates, that the ego is driving on a right-hand bend. After approximately 16 s, the upper time limit of the TTC is reached and a lane change manoeuvre to the next left lane is initiated. Normally, after the lane change manoeuvre to the left is finished and a steady state is reached again, the lateral acceleration would only

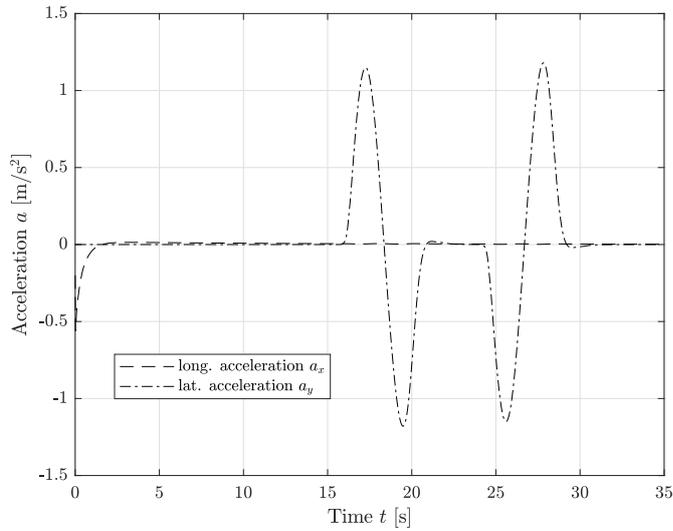


Figure 3.3.: Lane change manoeuvre to next left lane (straight road)

be a bit different compared to the value before the lateral driving manoeuvre (\Rightarrow radius is increased or decreased by the lane width depending on the direction of the curve). The difference of about 1 m/s^2 however indicates that the ego is driving on a road segment with a much bigger radius than before the lateral driving manoeuvre. At approximately 24s the lane change manoeuvre back to the rightmost lane is initiated.

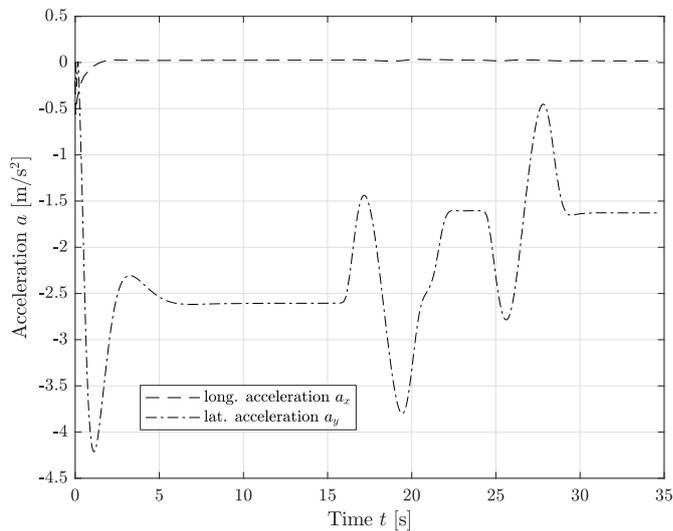


Figure 3.4.: Lane change manoeuvre to next left lane (curvy road)

Figure 3.5 shows the lateral offset, calculated from the static course located on the leftmost lane, as a function of time. In this diagram the lateral displacement of both simulations (straight and curve) are depicted and compared. In the detail it can be seen, that the calculated lateral displacement on the curvy road is slightly larger than the value determined during the lane change manoeuvre on the straight road.

This is caused by the calculation procedure of the TTC. Due to the fact, that the RADAR HiFi sensor only delivers the x component of the relative acceleration, the modified TTC approach is applied in x direction of the sensor coordinate system. At the time, when the target is detected via the RADAR sensor, it is actually more than the maximum range of the sensor in front of the ego. The relative distance delivered by the RADAR sensor is shorter than the arc length. Using the x component of the relative distance also means a reduction of the actual distance. However this is not critical. The calculated TTC value on a curvy road is smaller than the value on a straight road. This means that a lane change manoeuvre on a curvy road is initiated a bit earlier than on a straight road.

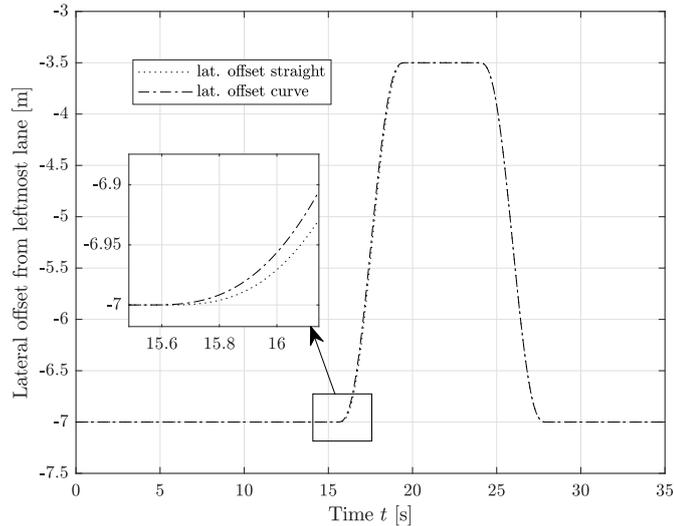


Figure 3.5.: Lateral offset - straight vs. curve

Using the approach of the polynomial of fifth order on a curve instead of a straight means that the shape of the polynomial is a bit deformed depending on the value of the radius, however it is possible to apply this method on highways. Furthermore it has to be kept in mind, that the assumption of a lateral acceleration $a_y = 0$ at the end of the lane change manoeuvre ($t = t_m$), used as a condition to determine the polynomial factors a to f , is not completely true. After the lateral driving manoeuvre is finished ($t = t_m$), this means that the vehicle is already positioned on the adjacent lane, a certain time is needed, until a steady state is reached again. This is independent of the shape of the road (curve or straight) and is illustrated in the figures 3.3 and 3.4. However, due to

simplicity reasons, the assumptions of section 2.2.2 are used to determine the polynomial factors, which influence the shape of the polynomial.

During a lane change manoeuvre on a curvy road, the lateral acceleration due to the curvature of the road is increased or decreased by the share of the lateral acceleration caused by the lateral driving manoeuvre. Equation (3.4) is used to describe how the lateral acceleration is influenced by the longitudinal cruising speed and the curvature of the road.

$$m \cdot a_y = m \cdot r \cdot \omega^2 \quad (3.1)$$

$$v_x = r \cdot \omega \quad (3.2)$$

$$a_y = r \cdot \left(\frac{v_x}{r}\right)^2 \quad (3.3)$$

$$a_y = \frac{v_x^2}{r} \quad (3.4)$$

The following points describe the variables used in the equations (3.1) - (3.4):

- m Mass of the vehicle [kg]
- r Radius of the curve [m]
- ω Angular velocity [1/s]
- a_y Lateral acceleration of the vehicle [m/s²]
- v_x Longitudinal velocity of the vehicle [m/s]

Equation (3.4) and the lateral acceleration values (at the steady state) of figure 3.4 can be used to determine the radii of the curvy road. At a cruising speed of $v_x = 130$ km/h and the lateral acceleration values $a_{y1} \approx -2.6$ m/s² and $a_{y2} \approx -1.6$ m/s² the radii $r_1 \approx 501$ m and $r_2 \approx 815$ m are calculated.

Note: The maximum cruising speed is adjusted according to the curvature of the road to keep the maximum lateral acceleration below a limit.

3.2. Investigation of the approach with the smallest TTC in front

A simple manoeuvre is used to investigate, if the approach of using the smallest TTC on each lane as the relevant criteria in the decision making process contributes to the

predicted advantages. A single lane change manoeuvre to the next left lane with two targets on the same lane in front of the ego is chosen. This scenario is tested on a straight road, depicted in figure 3.6.

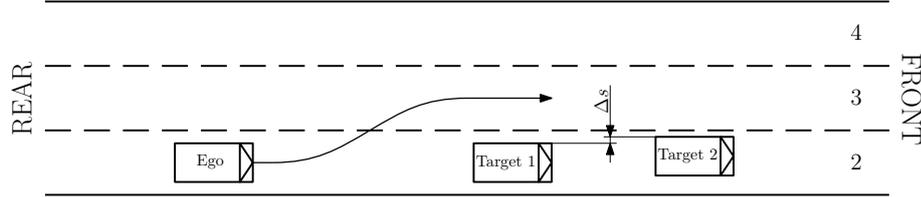


Figure 3.6.: Lane change manoeuvre to the next left lane (straight road)

In this simulation, the ego is driving with $v_x = 130$ km/h behind target 1 ($v_x = 135$ km/h). At the beginning of the simulation, the target 1 is positioned approximately 40 m in front of the ego. Target 2 is placed approximately 400 m in front of the ego, with a cruising speed of $v_x = 30$ km/h. In addition to that, target 2 is positioned a bit beside the centre line of the lane (Δs) to avoid, that target 1 is covering target 2 completely.

In the ideal case, the RADAR sensor should detect target 2 until it is within the FOV. After the conditions for a lateral driving manoeuvre are fulfilled, a lane change manoeuvre to the next left lane should be initiated. The simulation however shows, that the RADAR sensor model detects target 2 very late. To generate a critical case, target 1 is not considering target 2. Figure 3.7 illustrates, that after nearly 12s target 2 is detected. At this point a relative distance (x component) of about 69 m is measured and the HWC is not able to prevent a crash. Perhaps if the HWC is equipped with an AEB an accident could be avoided. Due to the fact, that target 1 is not considering target 2, the RADAR sensor loses target 1 when it drives through target 2. This behaviour can be seen in figure 3.7 at approximately 13.5 s.

The simulation shows, that using IPG CarMaker[®]'s RADAR HiFi sensor model on a straight road and applying the approach of the smallest TTC as the relevant criteria for a lane change manoeuvre does not deliver the desired advantages.

The same manoeuvre is also tested on a curvy road. The starting conditions are the same as in test manoeuvre on the straight road. Figure 3.8 depicts the relative distance between the ego and the two targets. Due to the curvature of the road, target 2 is detected a lot earlier (at approximately 8.3s). At this point the x component of the relative distance between ego and target 2 is 155 m. Figure 3.9 illustrates that a lane change manoeuvre to the left is initiated due to the very slow target 2. The lost signal of target 1 between 12.5 s and 13 s, depicted in figure 3.8, is due to target 1 driving through target 2.

The result of the second simulation shows, that using the RADAR HiFi sensor model in the approach with the smallest TTC on each lane works on curvy roads, because the targets in front are less covered by other vehicles due to the curvature of the road.

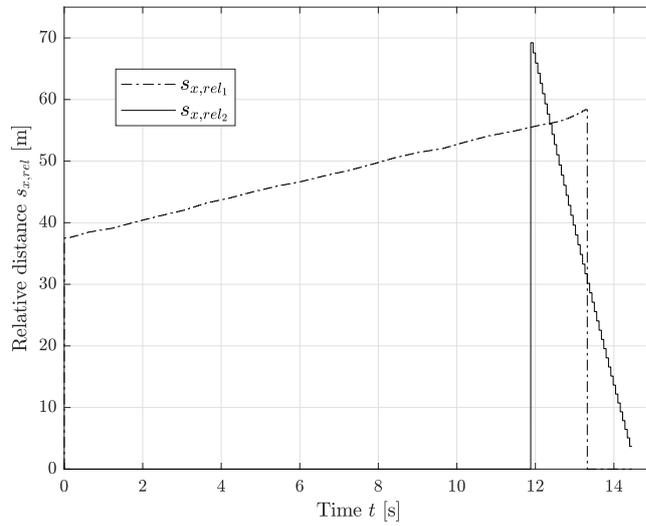


Figure 3.7.: Relative distance $s_{x,rel}$ between the ego and the targets (straight road)

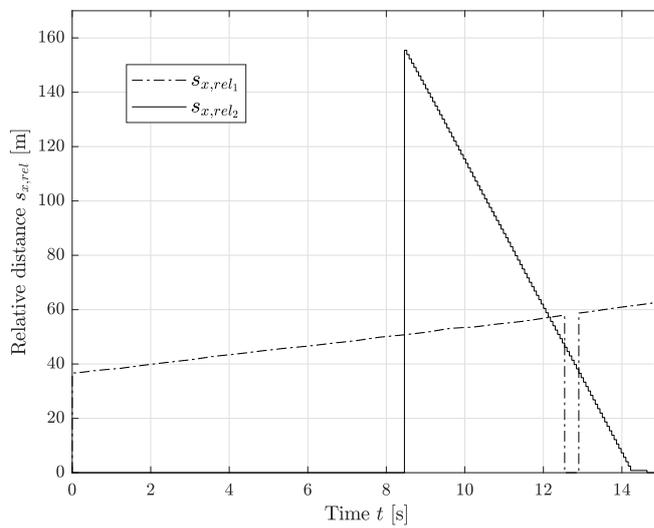


Figure 3.8.: Relative distance $s_{x,rel}$ between the ego and the targets (curvy road)

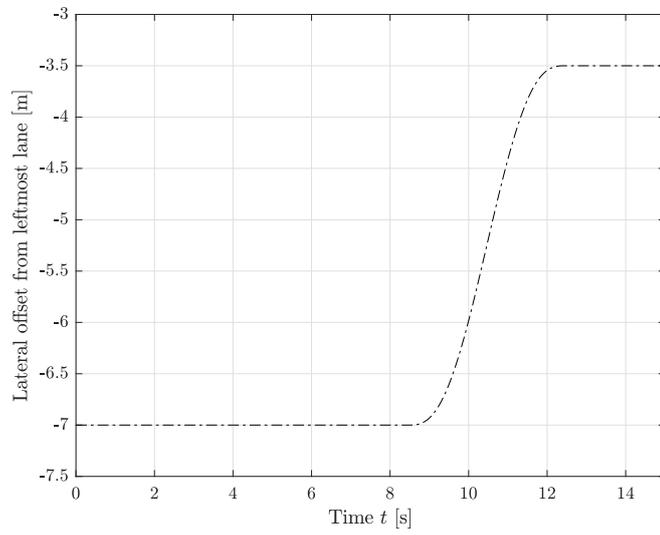


Figure 3.9.: Lateral offset (curvy road)

4. Summary

This section summarizes the main points of this master thesis for each chapter

Chapter 1: Introduction

In the first chapter of this thesis the motivation for the increasing deployment of *Advanced Driver Assistance Systems* (ADAS) in modern vehicles was described. Besides the reduction of energy consumption and the improvement of driving comfort, safety is one of the main drivers of these systems. Due to the fact that most of the accidents are caused by human misconduct, parts of the driving task are more and more performed by assistance systems to avoid accidents or to mitigate their effects. It was shown that supporting the driver has been one of the main goals since the beginning of the automotive industry and that step by step new systems were introduced.

As a very important basis, the classification of the driving task according to Donges and the categorization of the human behaviour according to Rasmussen as well as the interaction between these models were explained.

Using the levels of automation according to the *Society of Automotive Engineers (SAE)*, several ADAS were described as well as the requirements for a Highway Chauffeur. Hence environmental recognition is necessary in all ADAS, an overview of the used sensor types with the performance influencing factors was given.

New ADAS require extensive testing and validation before these systems can be integrated into modern vehicles. Due to the large amount of software, the development process was explained via the V-Model. It was shown that integrating different In-the-Loop methods into the development process has many advantages compared to the conventional method.

Finally IPG CarMaker[®] was presented as the used software tool for virtual testing in this thesis.

Chapter 2: Methodology

In the second chapter the procedure and the approaches with the necessary derivations, needed for the implementation of the HWC were explained. A virtual road, generated from measurement data, had to be converted because it was only available in OpenDrive[®], a file format, which is not compatible with IPG CarMaker[®] 6.0.4. Unfortunately, the conversion process was not faultless and for this reason, the virtual road had to be adjusted manually.

In the next step, a consisting model of a HWC was refloated. This model was only able to perform one lane change manoeuvre to the next left lane on a straight highway and was limited by the amount of targets. A lane change manoeuvre back to the right lane was not implemented. Due to the limitations of this model it was decided to implement a new model instead of trying to extend it. Like in the first model, the *Time To Collision* (TTC) was used as a main criteria for decision making. For this reason, a detailed derivation of this parameter was shown. Also a polynomial of fifth order was chosen to generate the trajectory needed for a lane change manoeuvre.

Different kinds of sensor types were investigated according to their suitability and usability. After appropriate sensors were chosen, a method for assigning the detected targets to the according lanes was explained. Then an overview of the HWC model, implemented in MATLAB/Simulink, with the different subsystems was given. Very important parts of this section were the implementation of the TTC approach, the assignment method, the decision making module as well as the calculation of the lateral displacement, which is used to place the ego on the road, relative to a static course.

Furthermore the simulation process in IPG CarMaker[®] was described. For longitudinal control CarMaker[®]'s ACC controller was used. The lateral control of the ego was performed by IPGDriver, a controller that adjusts the steering wheel according to the lateral displacement from the static course, calculated in the Simulink model of the HWC. At this point the limitations of the Line Sensor model, used for the assignment method, were discussed and the reason was explained, why tests were not made on the converted road.

Chapter 3: Results and Discussion

In this chapter the polynomial approach for the trajectory generation, originally intended for the use on straight roads, was investigated according to the suitability on curvy roads. The same overtaking manoeuvre (single lane change manoeuvre to the next left lane due to a target in front on the same lane and a lane change back to the right lane) was used to compare the occurring longitudinal and lateral accelerations as well as the calculated lateral displacements in both cases. Depending on the curvature of the road, the polynomial is a bit deformed. However, the polynomial approach can also be applied on curvy highways.

In addition to that, the method with the smallest TTC on each lane as a relevant criteria for a lane change manoeuvre was investigated. The simulations on a straight road showed, that the lateral displacement of the targets within one lane (same lane as the target is driving on) are very small and that the RADAR sensor model detects the vehicle behind the nearest target very late. Until a vehicle in front of the nearest target is detected, the system works in the conventional way (using the values of the nearest target).

On curvy roads however, the predicted advantages are reached.

Recommendations for further steps:

The following points give an overview of the possibilities how to continue this work .

- Varying the manoeuvre duration t_m to influence the shape of the polynomial for the trajectory, depending on specific situations.
- Extending the HWC of this thesis by using a consisting Simulink model of an AEB, implemented in an other master thesis.
- Updating to a higher release of IPG CarMaker[®] and test, if the problems of the Line Sensor model and the Lane Sections can be avoided.
- Using RADAR sensors for different ranges by adjusting the relevant parameters.
- Extending the HWC of this thesis that a lane change manoeuvre to the left is made, if the amount of lanes is reduced.
- Extending the function of the HWC that the ego can also enter and exit the highway

4. Summary

List of Figures

1.1. Annual road fatalities in Austria since 1983, [5]	1
1.2. Main causes of traffic accidents with personal injury in 2015, [14]	2
1.3. Three layers of human behaviour according to Rasmussen and three layers of the driving task according to Donges, [10]	4
1.4. SAE levels of automation, [21]	6
1.5. Basic functionality of a Highway Chauffeur (SAE level 3), [17]	8
1.6. Sensors for environmental recognition, [3]	10
1.7. Development process according to V-model [10]	15
1.8. V-Model with integrated In-the-Loop methods [10]	15
2.1. Lane Sections, [13]	22
2.2. Ramp, [13]	23
2.3. Routes and Paths	23
2.4. Converted road without adjustments	24
2.5. TTC without acceleration	27
2.6. TTC with modified approach: Case 1	28
2.7. TTC with modified approach: Case 2	28
2.8. TTC with modified approach: Case 3	29
2.9. Shape of the polynomial	31
2.10. Lateral displacement for a lane change manoeuvre to the next left lane	31
2.11. Quantities of CarMaker [®] 's Object Sensor [12]	34
2.12. Assignment of detected target to the according lane	37
2.13. Assignment of detected target to the according lane (detail)	38
2.14. Markers for assigning the detected targets to the according lanes	41
2.15. Finding the smallest values of the detected targets (front)	42
2.16. Ego with front and rear sensors	45
2.17. Target consideration in the decision making algorithm for a lane change manoeuvre to the next left lane	46
2.18. Sensors used for decision making	47
2.19. Lane change manoeuvre due to target on the same lane	48
2.20. Lane change manoeuvre due to target on first lane to the left of the ego	48
2.21. Lateral offset from static course	53
3.1. Comparison of a lane change manoeuvre on two different roads	55
3.2. Lateral displacement at different velocities (straight road)	56
3.3. Lane change manoeuvre to next left lane (straight road)	57

3.4. Lane change manoeuvre to next left lane (curvy road)	57
3.5. Lateral offset - straight vs. curve	58
3.6. Lane change manoeuvre to the next left lane (straight road)	60
3.7. Relative distance $s_{x,rel}$ between the ego and the targets (straight road) . .	61
3.8. Relative distance $s_{x,rel}$ between the ego and the targets (curvy road) . .	61
3.9. Lateral offset (curvy road)	62

List of Tables

1.1. Transition from a virtual environment to the real world, [10]	17
2.1. Considered targets in the scenario depicted in figure 2.17	46
A.1. Marker	VII
A.2. Variables of the subsystems and function blocks concerning the front of the ego	VIII
A.3. Variables of the subsystems and function blocks concerning the rear of the ego	IX

Bibliography

- [1] H. M. Waser A. Eichberger, W. Hirschberg. Skriptum Fahrzeugdynamik. Sommersemester 2017.
- [2] Z. Magosi D. Lindvai-Soos A. Eichberger, S. Bernsteiner. Integration of a phenomenological RADAR sensor model in IPG CarMaker for simulation of ACC and AEB systems. Available at <https://ipg-automotive.com/pressmedia/media-library/>. Accessed on 21 November 2018.
- [3] AAA Automotive. ADAS Sensor Calibration Increases Repair Costs. Available at <https://www.aaa.com/autorepair/articles/adas-sensor-calibration-increases-repair-costs>. Accessed on 8 December 2018.
- [4] S. Bernsteiner. *Integration of Advanced Driver Assistance Systems on Full-Vehicle Level - Parametrization of an Adaptive Cruise Control System Based on Test Drives*. Verlag der Technischen Universitaet Graz, 2016.
- [5] Bundesministerium für Inneres. Straßenverkehrstote in Österreich. Available at https://www.bmi.gv.at/202/Verkehrsangelegenheiten/files/Jahresvergleich_Verkehrstote_2017.pdf. Accessed on 1 November 2018.
- [6] M. Dupuis. *OpenDRIVE[®]/Style Guide For OpenDRIVE[®] Databases*. VIRES Simulationstechnologie GmbH, March 2014.
- [7] M. Dupuis. *OpenDRIVE[®]/Format Specification, Rev. 1.4*. VIRES Simulationstechnologie GmbH, November 2015.
- [8] D. Firesmith. Using V Models for Testing. Available at https://insights.sei.cmu.edu/sei_blog/2013/11/using-v-models-for-testing.html. Accessed on 13 November 2018.
- [9] Klein H. Steffan, E. Tomasch. Slides Vehicle Safety 2. Sommersemester 2017.
- [10] F. Lotz C. Singer H. Winner, S. Hakuli. *Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme fuer aktive Sicherheit und Komfort*. Springer, 2015.
- [11] IPG. *CarMaker[®]/Programmer's Guide Version 6.0.4*. IPG Automotive GmbH, 2017.
- [12] IPG. *CarMaker[®]/Reference Manual Version 6.0.4*. IPG Automotive GmbH, 2017.
- [13] IPG. *CarMaker[®]/User's Guide Version 6.0.4*. IPG Automotive GmbH, 2017.

- [14] Kuratorium für Verkehrssicherheit. Hauptunfallursachen bei Verkehrsunfällen mit Personenschaden. Available at <https://unfallstatistik.kfv.at/index.php/verkehr-mobilitat/hauptunfallursachen-bei-verkehrsunfaellen-mit-personenschaden>. Accessed on 1 November 2018.
- [15] Z. Magosi. Modellbildung und Simulation eines Radarsensors für die virtuelle Entwicklung von Fahrerassistenzsystemen. Master's thesis, Technische Universität Széchenyi István/Technische Universität Graz, 2013.
- [16] NGIN Mobility. Welche Sensoren gibt es im Auto - und wofür sind sie da? Available at <https://ngin-mobility.com/artikel/welche-sensoren-gibt-es-im-auto-und-wofuer-sind-sie-da/>. Accessed on 11 November 2018.
- [17] PEGASUS. The Highway-Chauffeur. Available at https://www.pegasusprojekt.de/files/tmp1/PDF-Symposium/03_The-Highway-Chauffeur.pdf. Accessed on 22 October 2018.
- [18] K. Reif. *Fahrstabilisierungssysteme und Fahrerassistenzsysteme*. Vieweg+Teubner, 2010.
- [19] Rohde & Schwarz. Generating radar interference for the automotive industry. Available at https://www.rohde-schwarz.com/au/solutions/test-and-measurement/automotive/radar/interference-test-/radar-interference_231752.html?change_c=true. Accessed on 11 November 2018.
- [20] R. Kazemi A. Eichberger S. Samiee, S. Azadi. Towards a decision-making algorithm for automatic lane change manoeuvre considering traffic dynamics. *Promet - Traffic - Traffico*, 2016.
- [21] Stanford Law School. SAE Levels of Driving Automation. Available at <http://cyberlaw.stanford.edu/blog/2013/12/sae-levels-driving-automation>. Accessed on 2 November 2018.
- [22] Texas Instruments. Cars are becoming rolling sensor platforms. Available at https://e2e.ti.com/blogs_/b/behind_the_wheel/archive/2014/09/25/cars-are-becoming-rolling-sensor-platforms#. Accessed on 11 November 2018.
- [23] VIRES Simulationstechnologie GmbH. Welcome to the World of OpenDrive®! Available at <http://www.opendrive.org/>. Accessed on 16 November 2018.
- [24] A. Zukancic. Functional Requirements Lane Change Assist. Copyright ©MAGNA STEYR Fahrzeugtechnik AG & Co KG 2015.

A. Appendix

A.1. HWC model in Simulink

In this section, the variables used in MATLAB/Simulink are described.

Table A.1.: Marker

MATLAB	Description
$marker = 10$	detected target is left, but more than two lane widths away
$marker = 4$	detected target on second lane to the left of the ego
$marker = 3$	detected target on first lane to the left of the ego
$marker = 2$	detected target on same lane as the ego
$marker = 1$	detected target on first lane to the right of the ego
$marker = 0$	no target detected
$marker = -1$	detected target on second lane to the right of the ego
$marker = -10$	detected target is right, but more than two lane widths away
End of the table	

Table A.2.: Variables of the subsystems and function blocks concerning the front of the ego

Thesis	MATLAB	Description
TTC_{all}	TTC_all	Vector with TTC values of all objects in the object list
TTC_{2l}	TTC_2l	Vector with TTC values of targets on second lane to the left of the ego
TTC_{1l}	TTC_1l	Vector with TTC values of targets on first lane to the left of the ego
TTC_s	TTC_s	Vector with TTC values of targets on same lane as the ego
TTC_{1r}	TTC_1r	Vector with TTC values of targets on first lane to the right of the ego
TTC_{2r}	TTC_2r	Vector with TTC values of targets on second lane to the right of the ego
s_{x,all}	sx_all	Vector with x components of rel. distances of all objects in the object list
s_{x,2l}	sx_2l	Vector with x components of rel. distances of targets on second lane to the left of the ego
s_{x,1l}	sx_1l	Vector with x components of rel. distances of targets on first lane to the left of the ego
s_{x,s}	sx_s	Vector with x components of rel. distances of targets on same lane as the ego
s_{x,1r}	sx_1r	Vector with x components of rel. distances of targets on first lane to the right of the ego
s_{x,2r}	sx_2r	Vector with x components of rel. distances of targets on second lane to the right of the ego
v_{x,all}	vx_all	Vector with x components of rel. velocities of all objects in the object list
v_{x,2l}	vx_2l	Vector with x components of rel. velocities of targets on second lane to the left of the ego
v_{x,1l}	vx_1l	Vector with x components of rel. velocities of targets on first lane to the left of the ego
v_{x,s}	vx_s	Vector with x components of rel. velocities of targets on same lane as the ego
v_{x,1r}	vx_1r	Vector with x components of rel. velocities of targets on first lane to the right of the ego
v_{x,2r}	vx_2r	Vector with x components of rel. velocities of targets on second lane to the right of the ego
TTC_{min}	TTC_min	Vector with smallest TTC value of each lane
<i>TTC_{min,2l}</i>	TTC_min(1)	smallest TTC value on second lane to the left of the ego
<i>TTC_{min,1l}</i>	TTC_min(2)	smallest TTC value on first lane to the left of the ego
<i>TTC_{min,s}</i>	TTC_min(3)	smallest TTC value on same lane as the ego
<i>TTC_{min,1r}</i>	TTC_min(4)	smallest TTC value on first lane to the right of the ego
<i>TTC_{min,2r}</i>	TTC_min(5)	smallest TTC value on second lane to the right of the ego
<i>s_{x,min,2l}</i>	sx_min(1)	x component of rel. distance of the nearest target on second lane to the left of the ego
<i>s_{x,min,1l}</i>	sx_min(2)	x component of rel. distance of the nearest target on first lane to the left of the ego

continued on next page

Thesis	MATLAB	Description
$s_{x,min,s}$	sx_min(3)	x component of rel. distance of the nearest target on same lane as the ego
$s_{x,min,1r}$	sx_min(4)	x component of rel. distance of the nearest target on first lane to the right of the ego
$s_{x,min,2r}$	sx_min(5)	x component of rel. distance of the nearest target on second lane to the right of the ego
$v_{x,min,2l}$	vx_min(1)	x component of rel. velocity of the nearest target on second lane to the left of the ego
$v_{x,min,1l}$	vx_min(2)	x component of rel. velocity of the nearest target on first lane to the left of the ego
$v_{x,min,s}$	vx_min(3)	x component of rel. velocity of the nearest target on same lane as the ego
$v_{x,min,1r}$	vx_min(4)	x component of rel. velocity of the nearest target on first lane to the right of the ego
$v_{x,min,2r}$	vx_min(5)	x component of rel. velocity of the nearest target on second lane to the right of the ego
End of the table		

Table A.3.: Variables of the subsystems and function blocks concerning the rear of the ego

Thesis	MATLAB	Description
TTC_{all}	TTC_all	Vector with TTC values of all objects in the object list
TTC_{1l}	TTC_1l	Vector with TTC values of targets on first lane to the left of the ego
TTC_{1r}	TTC_1r	Vector with TTC values of targets on first lane to the right of the ego
s_{x,all}	sx_all	Vector with x components of rel. distances of all objects in the object list
s_{x,1l}	sx_1l	x components of rel. distances of targets on first lane to the left of the ego
s_{x,1r}	sx_1r	x components of rel. distances of targets on first lane to the right of the ego
TTC_{min}	TTC_min	Vector with TTC values of nearest targets of each lane
$TTC_{min,1l}$	TTC_min(1)	TTC value of nearest target on first lane to the left of the ego
$TTC_{min,1r}$	TTC_min(2)	TTC value of nearest target on first lane to the right of the ego
End of the table		