Integrating human drivers' behavioural rules into microscopic traffic simulation

Martijn Vonk

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

Integrating human drivers' behavioural rules into microscopic traffic simulation

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Martijn Vonk

February 16, 2023

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Abstract

Autonomous vehicles (AVs) can solve a lot of problems related to traffic safety, comfort and congestion. While the sensors used by these vehicles are getting cheaper, more accurate, and software is improving, the first completely automated vehicle does not exist yet. When AVs start gradually driving on the roads, there will be a mixed traffic situation consisting of AVs and human driven vehicles. To understand the implications on the traffic performance, this thesis will implement a Model Predictive Controller (MPC) and the behavioural models for AVs into a numerical simulation in SUMO. The MPC can be used to control multiple AVs in larger traffic simulations to analyse the effects of certain penetration rates and traffic densities when changing the AVs driving behaviour. Both the Post-Encroachment Time (PET) and Time To Collision (TTC) will be used for decision making in overtaking and collision avoidance strategies. In this thesis, the MPC which is solving a quadratic programming problem has proven to successfully overtake other vehicles in an on- and off-ramp highway scenario which results in a smaller average travel time for the AVs than for the human drivers. It is shown that the decrease in average travel time for the AVs was not at the cost of the Human Vehicles (HVs). This is done by decreasing the TTC used by the AVs as a headway for longitudinal overtaking behaviour, and the PET used for lateral lane changing behaviour, independently to a minimum of 0.1 seconds to guaranty safety, and measure the average travel times for both HVs and AVs. Multiple simulations using different penetration rates ranging from 0 to 40 percent also show that the decrease in average travel time for the AVs was not at the cost of the HVs. We can conclude that using an MPC integrated together with the behavioural rules of human drivers into numerical simulation can give a good indication of the implications that AVs have on traffic performance.

Table of Contents

List of Figures

List of Tables

Glossary

List of Acronyms

SSM	Surrogate Safety Measures
TTC	Time To Collision
PET	Post-Encroachment Time
AVs	Autonomous Vehicles
MPC	Model Predictive Control
SUMO	Simulation of Urban Mobility
ZOH	Zero Order Hold
LTI	Linear Time Invariant
IDM	Intelligent Driver Model
PWA	Piecewise-affine
MLD	Mixed Logical and Dynamical
LQR	Linear-Quadratic Regulator
OTS	Optimal Target Selection
DARE	Discrete Algebraic Riccati Equation
LK	Lane-Keeping
ACC	Adaptive Cruise Control
AI	Artificial Intelligence
PFs	Potential Functions
LIDAR	Light Detection of Laser Imaging And Ranging
HVs	Human Vehicles

List of Symbols

β	Side slip angle
$\boldsymbol{u_p}$	Input sequence
δ_f	Steering angle
Δt	Sampling time
$\dot{x}$	Longitudinal displacement
$\dot{y}$	Lateral displacement
η	Output
$\in$	Is an element of
$\mathbb{R}$	Set of real numbers
$\mathbb{X}_f$	Terminal set
$\mathcal{U}$	Set of input constraints
$\mathcal{X}$	Set of state constraints
ω_b	Frequency band
ω_s	Sampling frequency
ψ	Heading angle
σ	Standard deviation
ξ	States
ξ_0	Initial state
a_x	Acceleration in x-direction
a_y	Acceleration in y-direction
a_{xi}	Acceleration of the leading vehicle
b	Maximum deceleration
d	Distance to dead end
f	A factor describing the time for a lane change
g	Headway in seconds
J	Cost function
J_p	Cost function based on the prediction horizon
k	Time step
l_f	Distance from center of gravity to front axle
l_r	Distance from center of gravity to rear axle
L_y	Vehicle width
L_{sc}	Sensor range
$L_{sx,PET}$	Safe PET distance
L_{sx}	Minimal headway to leading vehicle
L_{sy}	Minimal lateral distance to overtake
n	Number of dimensions
o	Target lane occupation
p	Prediction horizon

t_i	Time to maximum deceleration
t_r	Drivers reaction time
t_{PET}	Post-Encroachment Time
t_{TTC}	Time To Collision
u	Inputs
v	Vehicle speed
v_x	Velocity in x-direction
v_y	Velocity in y-direction
v_{const}	Constant longitudinal velocity
v_{des}	Desired speed
v_{max}	Maximum legal speed
v_{safe}	Safe speed
v_{xdes}	Reference longitudinal speed
v_{xri}	Relative longitudinal speed between two vehicles
v_{yri}	Relative lateral speed between two vehicles
x	X-coordinate
x_{ri}	Relative longitudinal distance between two vehicles
y	Y-coordinate
y_{CLi}	Y-coordinate of the center line of lane i
y_{max}	Upper road boundary
y_{min}	Lower road boundary
y_{ref}	Lateral coordinate reference
y_{ri}	Relative lateral distance between two vehicles

Acknowledgements

I would like to thank my supervisors, dr. Sergio Grammatico and dr. Haneen Farah for their assistance during the writing of this thesis. You have given me a lot of insights during our meetings, and always pushed me to try something new or different.

Secondly, I would like to thank dr. Narayana Raju and Luyao Zhang for the help with SUMO and getting a better understanding for the possibilities of MPC.

Finally, I want to thank my parents and all the people who have supported and guided me during this thesis.

Delft, University of Technology Martijn Vonk
February 16, 2023

"In the future, airplanes will be flown by a dog and a pilot. And the dog's job will be to make sure that if the pilot tries to touch any of the buttons, the dog bites him."

— *Scott Adams*

Chapter 1

Introduction

In this introduction, we start with the research motivation for this thesis, and formulate the research questions. Next we take a look at the current state of the art. Then, the contributions of this thesis will be discussed. We will finish with the structure of this thesis.

1-1 Research motivation

The first generations of autonomous driving cars have been driving on the roads for a few years now, and keep improving over time. Where the simplest versions of these cars started with some driver assistance like Lane-Keeping (LK) or Adaptive Cruise Control (ACC) at SAE level 1 [45], automated vehicles are slowly evolving to partial automation where the vehicle takes on more tasks from the driver. The next goal is of course to create fully automated vehicles that can drive safely under all different circumstances and in every situation without monitoring from the human driver. Since the developments will happen gradually, in the coming decades it is expected that we will see a mixed traffic situation consisting of both AVs and human driven vehicles. To get a better understanding of the implications of this mixed traffic situation on the traffic performance, this thesis will implement a controller using Model Predictive Control (MPC) and the behavioural models of AVs into a numerical simulation in SUMO, which will be used to run the simulation.

Apart from the extra comfort that Autonomous Vehicles (AVs) can bring and saving time from the driver which can be used for other activities, they have a lot more potential for being safe compared to human driven vehicles. Especially collision avoidance is a huge issue leading to accidents because of human errors such as misjudging situations, a delayed reaction time, unable to predict the Time To Collision (TTC) or misestimate the headway necessary for overtaking [59].

Next to comfort and traffic safety, AVs can solve a lot more problems that are caused by traveling by car. Where some highly densely populated countries like the Netherlands already have many highways and a good road structure, it falls short in solving the traffic congestion during rush hours. Since building more roads due to a lack of space is not an option, we have to use the roads already there more efficiently. This is where AVs could be used to increase the road capacity. Where human drivers have to keep a safe distance from other vehicles as a result of their reaction time, AVs could decrease this gap and therefore increase the capacity because they are able to react a lot quicker. This will lead to a higher highway capacity which could solve some congestion issues and traffic jams.

When we take a look at the physical limitations for AVs in the sense of sensor development, some huge progress has been made in reducing the production cost and improving the accuracy [3][25]. The more accurate and cheaper these sensors can be produced, the safer and better the autonomous vehicles can perform. Especially Light Detection of Laser Imaging And Ranging (LIDAR) is getting significantly cheaper [12][46]. LIDAR plays a huge role for AVs since it can be used to detect objects and estimate distances.

Finally the software used to control the autonomous vehicles is very important. Where companies like Tesla [69] update their software frequently, also a lot of research has been done on Artificial Intelligence (AI), sensor fusion, data analysis and a lot of different forms of control. Both the combination of the need for fully automated vehicles and the possibility to actually create one, lead to an acceleration of developments. When fully automated vehicles are finally available they will only gradually start driving on the roads. Some research is necessary to analyse the effects of different penetration rates of autonomous vehicles and how to implement the behavioural rules necessary for them to drive next to human drivers. These rules can be used to decide when to start a lane changing manoeuvre or how close can be driven behind another vehicle.
To conclude, all these research motivations have led to the following research question, which will be tested by running multiple simulations of a highway scenario:

What are the implications on traffic performance when implementing an MPC and human drivers' behavioural rules into a numerical simulation?
It is already known that autonomous vehicles can increase safety and comfort for their own passengers, but their effects on traffic performance in mixed traffic situations consisting of both AVs and Human Vehicles (HVs) are less known. Since this mixed traffic situation is a realistic scenario in the coming decades, this thesis will implement an MPC and human drivers' behavioural rules into a numerical simulation. This simulation will be run in SUMO. While the MPC is not specifically designed to improve the traffic flow, it can be interesting to see its implications on traffic performance.

1-2 Background

To answer the given research questions, first a controller making it possible to adjust the PET and TTC is necessary. In the literature, MPC is often used in highway scenario's because of its optimal solution and for the reason that it can handle constraints well, which is useful in collision avoidance. This can be used to fill in the gaps in research on different driving profiles.

1-2-1 Potential field functions

MPC is used in multiple different traffic applications, both in urban areas and highway scenario's. Especially in highway scenario's, MPC is often used for collision avoidance because it can handle constraints well. Simpler forms of MPC are used for realistic trajectory planning while keeping the computation times low which can be useful when implementing MPC real-time. More complex versions of MPC could be used, depending on the complexity of the vehicle model and the way the constraints are shaped. Sometimes a different model is used for trajectory planning than for motion control [37]. Since path planning using a shortest path algorithm can take a lot of time, Potential Functions (PFs) could be used to obtain the solution quicker [55]. They are often used in collision avoidance for robotic control and not that much in autonomous driving, but using them in autonomous driving could be useful because of their simplicity. These functions can model a 2D or 3D risk field which could be used for real-time path generation. While this is useful for collision avoidance, potential field based methods do not incorporate vehicle dynamics and can not ensure that the planned trajectory is feasible [35]. MPC could help to solve these problems by formulating the vehicle dynamics and using collision avoidance constraints in the optimization problem.

1-2-2 Linear MPC

A more common method than using PFs to simplify the collision avoidance, is to use linear or linearized vehicle models and constraints to simplify the computations. Although non linear models could describe car dynamics more realistically, a non linear solving method is required while research shows that linearized models used for trajectory planning could obtain similar results [17].
Some research has been done on using linear point mass models for collision avoidance [57][28]. This is an extremely simplified model which describes the dynamics in both the longitudinal and lateral direction, and needs multiple constraints to keep the model realistic. While being simple and easily usable real time, it raises questions whether this model captures the car dynamics correctly.
To describe the nonholonomic dynamics of a car more realistically, a linear kinematic bicycle model could be used. In this model the front and back wheels are lumped together, and it uses the steering angle and the longitudinal acceleration as inputs [37][35][38]. Research has shown that the kinematic bicycle model produces consistent results for motion planning, and generates feasible trajectories compared to a more complex 9 degrees of freedom model. This is the case as long as the lateral acceleration stays below 0.4g [24][31], which makes it ideal for highway driving. More research confirms that using a bicycle model is less computationally expensive than existing methods while giving realistic results [24].

1-2-3 Collision avoidance constraints

The hardest part in autonomous driving is using the right constraints for collision avoidance. Where avoiding the shape of a car or an elliptical zone on a highway would lead to non-convex constraints [28], a non linear solver is necessary or the constraints need to be simplified by approximating them by a convex function. This can be done by exploiting the road structure information and combine rear-end and side collision constraints together into one constraint, which reduces the optimization times drastically [57]. Using linear constraints, a quadratic programming problem could be formulated and solved efficiently.

1-2-4 Driving profiles

Multiple kinds of research have been done on different driving profiles. The most common driving profiles in the literature are the more aggressive driving profiles, and their influence on fuel consumption and the emission of air pollutants. Research has shown that aggressive driving in urban areas can lead to an increase in fuel consumption and an increase in pollutant emissions up to 30-40% compared to calm driving [68]. Other research obtains similar results comparing safe and aggressive driver styles [9]. Most of these kinds of research are based on environmental aspects and traffic safety [40], and only includes the aggressiveness of the driving style.
This is where Surrogate Safety Measures (SSM) can be used. SSM are measurements which uses microscopic traffic parameters such as the headway in distance or time headway, or vehicle speed or acceleration. These parameters can be used to describe the relation between multiple vehicles during traffic events to indicate the chance and severity of a possible traffic accident. While SSM are usually used to describe traffic safety for human drivers, they could also be used to describe overtaking behaviour for autonomous vehicles. Where autonomous vehicles have a shorter reaction time than human drivers, they could safely keep a shorter distance while following other vehicles. Where this could increase the road capacity and allow for more mobility in lane changing, it is not known whether this affects the other road users.

1-3 Results of this study

The goal of this thesis is to implement an MPC controller and human drivers' behavioural rules into a numerical simulation. The MPC will be used for overtaking and collision avoidance in an on- and off-ramp highway scenario. Using the PET for the lateral lane change behaviour and the TTC for the longitudinal lane change behaviour, different penetration rates of AVs used in different traffic densities are tested in traffic simulations in SUMO. The contributions of this thesis can be summarized by answering the research question:

What are the implications on traffic performance when implementing an MPC and human drivers' behavioural rules into a numerical simulation?
We can conclude that implementing the MPC and the behavioural rules of human drives into a numerical simulation with mixed traffic driving in an on- and off-ramp highway scenario gives a good overview of the effects that AVs have on the other road users. At high traffic flow situations, AVs have a lower average travel time than HVs because they can keep smaller headways than the human drivers. Next to this benefit for the AVs, lowering the PET that the AVs keep for lateral lane changing behaviour also decreases the average travel time of the HVs slightly. When the penetration rate increases, the average travel time of the HVs is decreased even further. While this is a positive effect of AVs, the decrease in travel time is very small and would not be a significant benefit for human drivers in a real world situation. For the longitudinal collision avoidance, linear constraints are used to keep a minimal TTC from a

possible leading vehicle. Because of these linear constraints, a decrease of the minimal TTC that an AV should keep from another vehicle, leads to a very small increase of the average travel time for other road users. This could be explained by the behavioural rules of the AVs caused by the linear constraints used in this thesis. When the traffic flow is relatively low we found that the effects caused by AVs keeping a low minimal headway are different since the maximum speed and other behavioural rules play a more important role.

1-4 Thesis structure

This thesis is organised in the following way. At first Chapter 2 will discuss all the research methodology necessary for this thesis, and is divided into 4 parts. Chapter 2-1 will describe the overview of this thesis, and how all building blocks are combined together to run the simulations. Finally the scenario will be discussed, and the details about the simulations are explained.

Next Chapter 2-2 will describe the vehicle models. To control the automated vehicles, a realistic vehicle model for both decision making and following the trajectories is necessary. To create some realistic human driving behaviour, different car-following models and lane change models are compared. Simulation of Urban Mobility (SUMO), a simulation program will be controlling these human driven vehicles.

Chapter 2-3 gives an overview of the traffic turbulence on highways and the SSM used in this thesis. Both traffic turbulence and SSM could be used to describe traffic quality, and can be included in the decision making process of the AVs. The SSM used for this thesis are the PET and TTC.

In Chapter 2-4, the control method used to control the AVs is explained. For this thesis, MPC will be used. The quadratic programming problem will be shown in this chapter, together with the cost function and constraints. Also the different variations of MPC are discussed.

Next, in Chapter 3 the results will be presented. An overtaking simulation is executed such that the MPC can successfully overtake other vehicles while using the right driving behaviour. Next some larger simulations are run to answer the 3 research questions.

Finally in Chapter 4 the work of this thesis will be summarized. The conclusions about the results are discussed, and some future challenges and recommendations are given.

Chapter 2

Research Methodology

In this chapter, the research methodology will be explained. This chapter is divided into 4 parts. First we will discuss the scenario and give an overview of the simulation. Next the vehicle models for both the human drivers and automated vehicles will be explained. After that we take a look at the surrogate safety measures which will be used to describe the vehicle behaviour. Finally the MPC will be discussed which will control the AVs in the simulation.

2-1 Scenario

In the first section of this chapter, the scenario will be described. An overview of all the different components necessary to answer the research questions will be given. How all these different components are connected to each other to gather the necessary data will be discussed in the next subsections. To run the simulations and generate data, Simulation of Urban Mobility (SUMO) will be used to control the humen drivers. NetEdit will be used to create a suitable traffic scenario and Matlab will be used to create the controllers to control the Autonomous Vehicles (AVs) and to gather and process the data. To make SUMO and Matlab connect with each other, Traci4Matlab will be used. But first an overview of this thesis will be given.

2-1-1 Overview

The goal of this thesis is to answer the research questions formulated in Chapter 1. To answer the research questions, we need to create a controller to control multiple AVs, and a simulation to test which effect it has on the other road users. To control the AVs, Model Predictive Control (MPC) will be used. The fact that MPC can find an optimal solution and can handle constraints, makes this a suitable and very popular choice for collision avoidance. To be able to use AVs using the desired behaviour, we have to translate this behaviour into some mathematical equations which can be used in the cost function. Next to the benefit of finding an optimal solution, there is also the possibility to implement constraints. This can be useful to help the AVs stay within the road boundaries and avoid collisions. How the MPC will be designed, will be discussed in more detail in Chapter 2-4.

The decision making process during a time step consists of 4 different parts. The first step is to get the necessary information from nearby vehicles. In the real world, AVs could gather information about the world around them through sensors like camera, lidar or radar [48]. This data can be used in the cost function and the constraints of the MPC to generate an optimal input to control the vehicle as desired. Instead of using the real world, we are using SUMO to run the simulations which will make it a lot easier to gather the necessary data without uncertainties. These data containing the information of nearby vehicles are led to the MPC which can execute the next step.

The second step is to run the optimization algorithm to obtain an optimal input to control the AVs. To create a realistic driving trajectory, a vehicle model is necessary. Because a car is nonholonomic, it is not possible to drive in lateral direction without moving in longitudinal direction. Since linear models require a lot less computation time than non linear models when using optimization, a linear vehicle model will be used by the MPC for trajectory planning. With this linear model, a cost function, the constraints and the information about nearby vehicles, it will be possible for the MPC to run an optimization algorithm to obtain an optimal input. This optimal input can be used in the next step to control the vehicle.

In the third step a non linear vehicle model will be used to calculate the actual trajectory of the vehicle. Since non linear models are more realistic because they include more complex behaviour, the optimal input will be used in a non linear model to calculate the trajectory of the vehicle. This will also demonstrate whether a linear model used for trajectory planning could result in realistic trajectories in the real non linear world. Or in this thesis, a non linear simulation. Both the linear and the non linear vehicle models used for this thesis will be further described in Chapter 2-2.

Finally, the trajectory obtained by the non linear model can be sent to SUMO to change the location of the vehicle in the simulation. Next SUMO will control all the other vehicles in the simulation which represent the human driven vehicles and proceed to the next simulation step. At that next time step the AVs can gather the data of the surrounding vehicles again which closes the circle. In Figure 2-1 the complete process containing the 4 steps is shown.

In the next section, the highway scenario and the corresponding traffic flows used in SUMO will be discussed.

2-1-2 On- and off-ramp scenario

To tune and test if the controller created in Matlab could work, running a simulation presenting a highway as realistic as the real world would be necessary. Since we want to use the traffic turbulence and some Surrogate Safety Measures (SSM), creating a scenario where the traffic density is relatively high would be useful for some increased traffic turbulence to occur. Usually at the bottlenecks, which could be an on-ramp, off-ramp or weaving segment, an increase in traffic turbulence can be found [71]. At these bottlenecks the total road capacity drops and the traffic density increases. This is because human drivers naturally increase their

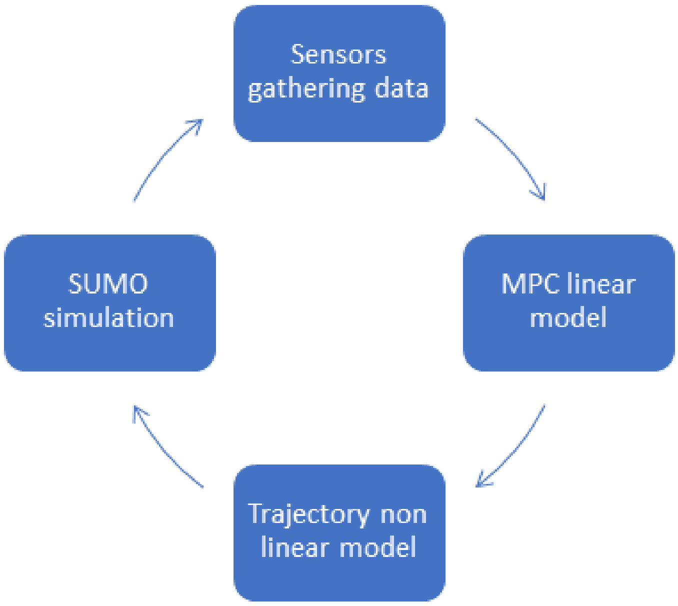

Figure 2-1: Simulation overview

headway to the leading vehicle or need to slow down. When the headway is too small and vehicles collide, or the road capacity exceeds its maximum capacity, traffic jams occur. The benefit of AVs is that since their reaction time is a lot smaller than that of human drivers, headways could be safely reduced, depending on the sensors and computation power. When headways are lowered, the road capacity will be increased since the same amount of vehicles occupy less space. Since some countries are densely populated, it is not always possible to construct more highways. This is where AVs could solve some problems.

For this thesis, an on- and off-ramp scenario is created. To keep the computations for a given simulation time reasonable, it is desirable to chose a small but realistic scenario. In the Netherlands there are also some guidelines for designing highways. These are described in 'Richtlijn Ontwerp Autosnelwegen' [63]. To keep the simulation small, a highway consisting of one lane with an on- and off-ramp will be created. This scenario is illustrated in Figure 2-2.

According to Rijkswaterstaat [63], a minimum road segment of 400 meter in length would be enough using this number of lanes. To make sure that all vehicles in the simulation have enough time to chose the right lane for their destination and for the traffic to stabilize, a length of 500m is chosen. This road segment is shown as segment 2 in Figure 2-2. Leading to this segment, we have segment 1. This segment will be only 100m in length since it consists of only one lane. This is the same for segment 3, which is where the main highway continuous. Both road segments leading to the on-ramp and continuing from the off-ramp are also 100 meter long.

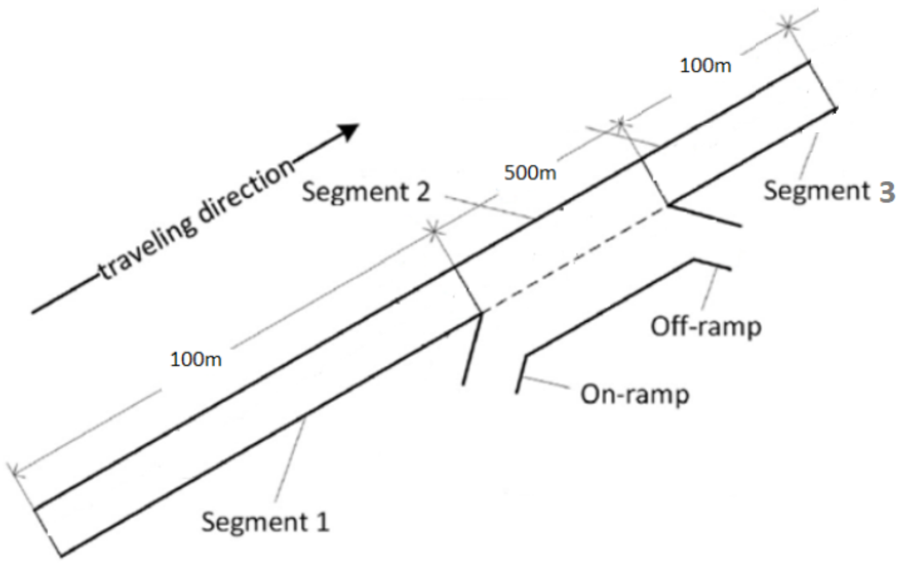

Figure 2-2: Highway scenario

Traffic flows

Next to the right scenario, which follows from the guidelines [63], the right traffic flows have to be selected. According to these same guidelines, the maxmium capacity of a scenario shown in Figure 2-2, consisting of 2 lanes, would be 4830 vh/h since the scenario is short, and assuming that the percentage of trucks will be 0 percent in this simulation. This maximum capacity has been verified using the on- and off-ramp scenario in SUMO. Because 4830 vh/h is the maximum capacity and we want some free space available for lane changing or accelerating. Usually around 75 percent of the maximum capacity is used. For comparison reasons, multiple different traffic flows will be used in this Thesis. The 2 most used traffic flows will be 1800 veh/h for some low density simulations and 3200 veh/h for some high density simulations. A possible realistic way to divide these flows is to make 75 percent of this flow start at segment 1, and 25 percent starts at the on-ramp. From both these flows, 75 percent has as destination segment 3, while 25 percent has the off-ramp as destination. An overview of the distribution of these flows can be seen in Table 2-1.

Total flow	Seg 1 - Seg 3	Seg 1 - Off	On - Seg 3	On - Off
1800	1013	338	338	113
3200	1800	600	600	200

Table 2-1: Overview of the different traffic flows

For the total traffic flow of 3200 vh/h, a total of 1800 + 600 = 2400 vh/h will start on the main highway which is segment 1, and consists of 1 lane. Since 2400 vh/h is around the maximum capacity of a 1 lane road, and when using a distribution as described above, 3200 vh/h as a total flow will be the maximum that we can test with this distribution. This is

because 600 + 200 = 800 vh/h start from the on-ramp using this distribution. Only when the on-ramp flow is also at the maximum capacity, which is not realistic, 4830 vh/h will be reached.
All the traffic flows in Table 2-1 are the total traffic flows used in the simulation, containing both AVs and human drivers. To show the effects of the AVs, different penetration rates of AVs will be used. For example, when a penetration rate of 10 percent will be used, 10 percent of all the traffic flows in Table 2-1 will consist of AVs. Over all these separate traffic flows, these AVs are chosen randomly, to make sure they can both influence each other, and the human drivers.

Simulation time

Now that we have discussed the scenario and the traffic flows, we have to chose a realistic simulation time. Often in traffic, some small effects can evolve over time and cause some huge effects like shock waves or traffic jams. For this reason it is important to create a simulation time which captures all these behaviours. To make sure the traffic flows in the simulation are stabilized, a warm up round of 10 minutes will be used. After this warm up round, all data will be gathered over the time span of 1 hour. According to [63], this would be long enough to make sure that most possible changes of traffic are captured in the data.

2-1-3 Software

In this next section, the software used for this Thesis will be described. To design and tune the controllers, Matlab will be used. For the simulations, SUMO, a microscopic simulation program is found suitable. To generate the network, NetEdit is chosen. And finally to make it possible to connect the controller in Matlab with the simulation in SUMO, Traci4Matlab will be used. These different tools will all be described below.

SUMO

To run the simulations, the simulation program SUMO will be used. SUMO is an open source simulation program, used to simulate microscopic traffic models, and is developed by the German Aerospace Center. Since it is open source, many packages are available to use together with SUMO. This is necessary when creating the MPC controller to control the AVs in Matlab. Different programs can be used to design a scenario, which can then be imported into SUMO. The output data is given in XML, which can be easily converted into CSV which can be opened and analysed in for example Excel or Matlab. When the scenario is imported, it is possible to add some vehicles or vehicle flows. Also routes and different vehicle types are included. For different kinds of driving behaviour, different vehicle models can be used. it is also possible to create own vehicle models.

NetEdit

The highway scenario used for the simulation in SUMO will be designed in NetEdit [5]. NetEdit is easy to use and often used in combination with SUMO. The network, vehicle flows and vehicle types can be created in NetEdit and then used by SUMO.

Traci4Matlab

In SUMO it is only possible to run simulations of road networks and load the necessary demand elements. To connect SUMO to Matlab to control the AVs, Traci4Matlab is necessary [2]. This is an implementation of the Trafic control interface that allows the user to interact with SUMO. This can be done in a client-server scenario where Matlab acts as the client, and SUMO as the server. In Matlab, several different functions can be used to gather data, give inputs to vehicles in SUMO and control all the settings of the simulation.

2-1-4 Summary

In this section, a quick overview of this thesis is given. It is visually shown how all different parts are connected to each other to run the simulations and gather the data. The on- and off-ramp scenario is explained, together with the different vehicle flows and the important details on how to set up the simulation. In the next chapter, Chapter 2-2, the different vehicle models used by the human drivers and the AVs are shown.

2-2 Vehicle models

The next step after creating the scenario is to create a model for the vehicle trajectories. First we will discuss the different kind of models that could be used to describe the dynamics of a car. Different models will be discussed and their pros and cons compared. As explained in Chapter 2-1 at the overview, a linear model is preferred for the MPC to decrease the computational load. But since we want realistic dynamics to get a better approximation of the real world, we use a more complex model for the execution of the trajectories.
Next the different possibilities for the vehicle models in SUMO are discussed. To describe the behaviour of the human drivers, controlled by SUMO, another other vehicle model is necessary. This model will describe the decision making for the human drivers, and can be divided into two separate models. One model for the longitudinal behaviour, and a second model to describe the lateral lane changing behaviour. These models will be discussed at the end of the chapter, but first the vehicle models for the AVs.

2-2-1 Vehicle models for the AVs

To simulate cars, a lot of different vehicle models could be used. It is important to select the right model that is as simple as possible, but complex enough to capture the right dynamics. One of the simpler models is the point mass model. This is a model that is often used for collision avoidance in combination with MPC to keep a low computational load [57] [28]. This model describes the vehicle as a point mass, which means that all mass is located in one point. When using second order equations, forces can act on this mass to create an acceleration. Usually a state space model is created where the acceleration in x- and y-direction are used as an input and where the position and velocity are used as states. A point mass model could look like this in discrete time:

$$v_x(k+1) = v_x(k) + a_x(k)\Delta t \tag{2-1}$$
$$x(k+1) = x(k) + v_x(k)\Delta t \tag{2-2}$$
$$v_y(k+1) = v_y(k) + a_y(k)\Delta t \tag{2-3}$$
$$y(k+1) = y(k) + v_y(k)\Delta t \tag{2-4}$$

In these equations, v_x is the velocity in x-direction, a_x is the acceleration in x-direction, and v_y and a_y are the velocity and acceleration in y-direction. These acceleration and velocities that are constant during a time step can be used to calculate the x-coordinate and y-coordinate. While these equations are very simple, the question is whether a model like this is realistic enough to describe a car. Since cars are non holonomic, which means they have wheels and can not drive sideways, some adjustments would be necessary to make this model realistic. When used for MPC, this can be done by using some constraints that only allow lateral movement in combination with some longitudinal movement. Even then the question remains whether this kind of model is realistic enough.

Next to some very easy models like the point mass models, also a lot more complex models are used. This could be necessary for trajectory planning when for example the forces on the body, tyre dynamics or wheel slip or all of them are included in the model [31][61]. For the given highway scenario, described in Chapter 2-1, this is not necessary. It is important to capture some realistic non holonomic behaviour and describe some simple trajectories for collision avoidance and overtaking. For this reason a kinematic bicycle model is chosen [37]. This model is often used in the literature for obstacle avoidance because its simplicity and realism [35][38]. Compared to the point mass model, a kinematic bicycle model provides a suitable compromise between model order and accuracy [37]. This is a simpler version of the dynamic bicycle model which is similar, but uses more parameters because it also takes the tyre dynamics into account. Since this is not strictly necessary for highway manoeuvres and obstacle avoidance, the kinematic bicycle model is chosen. This model will be described in the next section.

Kinematic bicycle model

In this section the kinematic bicycle model will be explained. A bicycle model is a simplified model of a car, where the two front wheels and back wheels are lumped together, such that it looks like a bicycle, as can be seen in Figure 2-3. This vehicle models assumes no slip between the tyres and the road when the lateral acceleration a_y is within certain bounds ($|a_y| \leq 0.4\text{g}$) [24][31], which makes this model suitable for highway driving. The kinematic bicycle model can be described by the following equations:

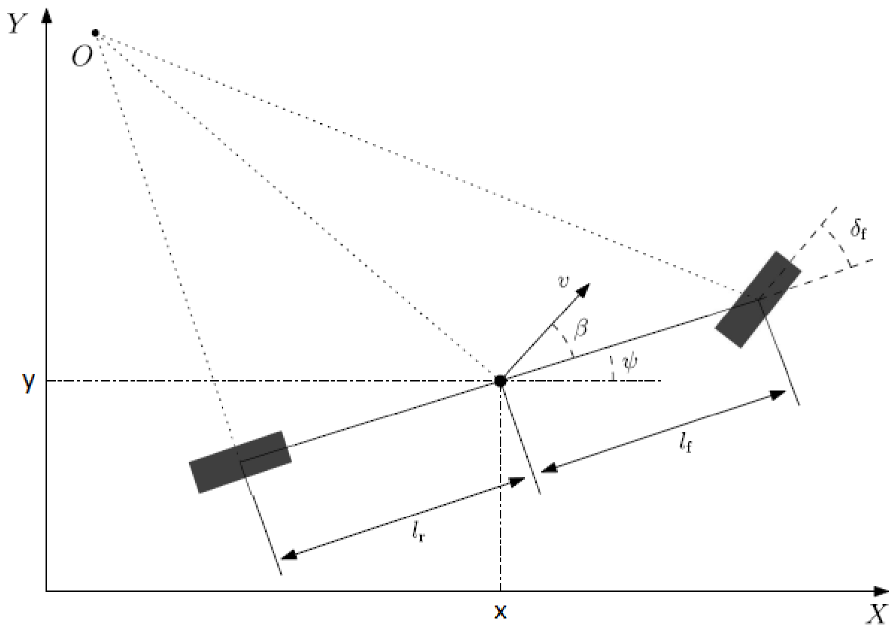

Figure 2-3: Kinematic bicycle model [37]

$$\dot{x} = vcos(\psi + \beta) \tag{2-5}$$
$$\dot{y} = vsin(\psi + \beta) \tag{2-6}$$
$$\dot{\psi} = \frac{v}{l_r} sin(\beta) \tag{2-7}$$
$$\dot{v} = a_x \tag{2-8}$$
$$\dot{\beta} = tan^{-1}(\frac{l_r}{l_f + l_r} tan(\delta_f)) \tag{2-9}$$

Under normal highway driving conditions, the steering angle δ_f, heading angle ψ and side slip angle β are small, which means we can use the small angle approximation for the sinus and cosine and substitute Equation 2-9 into Equation 2-5, 2-6 and 2-7 to obtain the following equations:

$$\dot{x} = v \tag{2-10}$$
$$\dot{y} = v\psi + \frac{l_r}{l_f + l_r} v\delta_f \tag{2-11}$$
$$\dot{\psi} = \frac{1}{l_f + l_r} v\delta_f \tag{2-12}$$
$$\dot{v} = a_x \tag{2-13}$$

In the first equation 2-10, we see that the longitudinal displacement of the center of gravity $\dot{x}$ equals the speed of the vehicle v. The second equation 2-11 describes the lateral displacement of the center of gravity $\dot{y}$, which depends on the speed v, the heading angle ψ, the steering angle δ_f, and l_f and l_r which are distances from the center of gravity to the front and rear axles. In Equation 2-13, the change in velocity depends on the longitudinal acceleration a_x. For this system, the steering angle δ_f and the longitudinal acceleration a_x can be used as inputs.

Non linear discrete time kinematic bicycle model

To make the AVs follow the right trajectory, a non linear kinematic bicycle model will be used. Since the model above is given in continuous time, the model has to be converted to discrete time first to use it in Matlab. This can be done using Zero Order Hold (ZOH) for a sample time Δt, which is the most common method. At every time step, a sample of the continuous time signal is taken, as can be seen in Figure 2-4. The continuous time signal can be reconstructed from these samples by holding a sample for a certain time until the next sample is received. This leads to the following discrete time non linear kinematic bicycle model:

$$x(k+1) = x(k) + v(k)\Delta t \tag{2-14}$$

$$y(k+1) = y(k) + v(k)\psi(k)\Delta t + \frac{l_r}{l_f + l_r} v(k)\delta_f(k)\Delta t \tag{2-15}$$

$$\psi(k+1) = \psi(k) + \frac{1}{l_f + l_r} v(k)\delta_f(k)\Delta t \tag{2-16}$$

$$v(k+1) = v(k) + a_x(k)\Delta t \tag{2-17}$$

In these equations, $x(k)$, $y(k)$, ψ(k), $v(k)$, $a_x(k)$ and $\delta_f(k)$ all depend on k which is the time step. This is the non linear model that will be used to describe the trajectories of the AVs. When the MPC calculates the most optimal inputs $a_x(k)$ and $\delta_f(k)$ at a certain time step k, these can be used to calculate the new x- and y-position and heading angle of the vehicle.

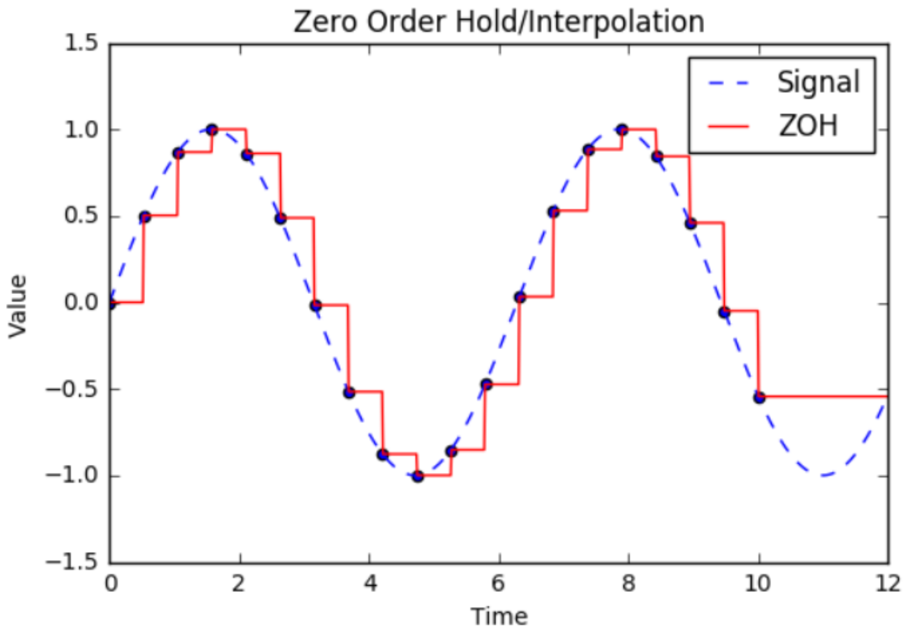

Figure 2-4: Zero Order Hold [51]

Sampling time

To obtain the best results, which can be done by minimizing the processing time while having a maximum accuracy, it is important to chose the right sampling time Δt. According to (M. Verhaegen and V. Verdult 2007), as a rule of thumb, the sampling frequency ω_s should be 10 times as high as the frequency band ω_b [72]. Using the step response, when we select about 8 or 9 samples during the time period from 0 to the steady state, we sample about $10\omega_b$ [72]. An example of what a step response looks like is visually illustrated in Figure 2-5. When taking the time it takes to perform a lane changing manoeuvre or to accelerate for a short period of time into account, a sampling time of 0.2 seconds is selected. This is the same order of magnitude compared to other sampling times used for collision avoidance for highway driving [37] [38].

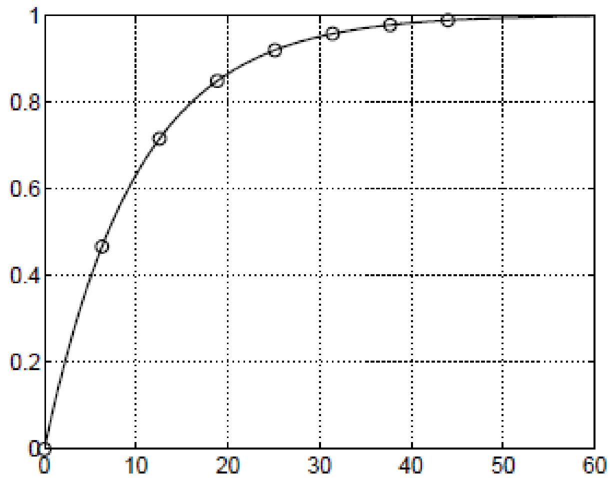

Figure 2-5: Step Response for selecting the sampling time [72]

Linear discrete time kinematic bicycle model

For the trajectory planning done by the MPC a simpler model than the non linear kinematic bicycle model is preferred. This can be done by linearizing the non linear model around a constant longitudinal speed. This makes it possible to use a quadratic cost function with linear constraints which can be solved using simpler optimization methods and has the benefit of not getting stuck in a local minimum. When linearizing around a constant longitudinal speed, it is important to note that this might lead to some inaccuracies in lateral and headway predictions which could lead to unsafe or unfeasible trajectories. Using the right constraints [13] which will be described in Chapter 2-4, the non linear model can be rewritten as a Linear Time Invariant (LTI) system. For the states, $\xi \triangleq [x(k), v_x(k), y(k), \psi(k)] \in \mathcal{X} \in \mathbb{R}^4$ will be used and the input will be $u \triangleq [a(k), \delta_f(k)] \in \mathcal{U} \in \mathbb{R}^2$, where $\mathcal{X}$ and $\mathcal{U}$ are the linear state and input constraint sets. This leads to the following LTI system:

$$\xi(k+1) = A\xi(k) + Bu(k) \tag{2-18}$$

$$\xi(k+1) = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & v_{const}\Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ v_x(k) \\ y(k) \\ \psi(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \Delta t & 0 \\ 0 & v_{const}\frac{l_r}{l_f+l_r}\Delta t \\ 0 & \frac{v_{const}}{l_f+l_r}\Delta t \end{bmatrix} \begin{bmatrix} a(k) \\ \delta_f(k) \end{bmatrix} \tag{2-19}$$

In these equations, v_{const} is the constant longitudinal velocity. Since the first state, x(k) is not of importance for the lane changing and trajectory planning done by the MPC, this state is removed which results in the following LTI system:

$$\xi(k+1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & v_{const}\Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x(k) \\ y(k) \\ \psi(k) \end{bmatrix} + \begin{bmatrix} \Delta t & 0 \\ 0 & v_{const}\frac{l_r}{l_f+l_r}\Delta t \\ 0 & \frac{v_{const}}{l_f+l_r}\Delta t \end{bmatrix} \begin{bmatrix} a(k) \\ \delta_f(k) \end{bmatrix} \tag{2-20}$$

This linear state space system consisting of 3 states and 2 inputs defined as $\xi \triangleq [v_x(k), y(k), \psi(k)] \in \mathcal{X} \in \mathbb{R}^3$ and $u \triangleq [a(k), \delta_f(k)] \in \mathcal{U} \in \mathbb{R}^2$ will be used by the MPC. In these equations, $\in\mathbb{R}^n$ means the states or inputs are a member of the set of real numbers in a given number of dimensions n. In the next section the vehicle models for the human drivers will be discussed.

2-2-2 Vehicle models for the human drivers

The vehicle models used by the human drivers in SUMO will be described next. Since these vehicle models should present human driving behaviour, it is important that these models are as realistic as possible. For the longitudinal behaviour, the Krauss model will be used. The lateral driving behaviour will be described by the lane changing model.

Krauss

In the literature, a lot of different car-following models for longitudinal behaviour have been used. To get as realistic human driving behaviour as possible in our simulation, it is important to chose the right model. In SUMO a lot of research have been done on realistic driving behaviour [47], where the hardest part is to find a simple model that is realistic enough for the right scenario [50]. While one model could excel in for example urban areas, it could perform worse compared to other models in for example a highway scenario. It is important to select the right model for the right circumstances.

A very common and widely used model is the Intelligent Driver Model (IDM) [16], which is developed by Treiber, Hennecke and Helbing in 2000 as an improvement for other intelligent driver models such as the Gipps' model which has been developed in the late 1970's. While being that widely used, some downsides of the IDM are the ill-posedness where velocities become negative and can converge to $-\infty$ [34]. Of course some adjustments have been made over the years, but some issues remain.

Next to the IDM, another widely used car-following model is the Krauss model [49], which is also the default car-following model in SUMO. In Urban areas, it has been shown that the IDM can perform better than the Krauss model when there are intersections involved [10][53]. These results are obtained by calibrating the IDM using a genetic algorithm. While the IDM gives good results in urban areas using this calibration, the results are far from perfect in other scenario's. Since this thesis is about AVs used in highway scenario's, it is important to compare these models in a similar scenario, and also test how well these models can be implemented in SUMO. There are papers written that show that IDM outperforms Krauss at describing realistic human driving behaviour on highways [32], but also papers which claim the opposite [15]. This last research also states that IDM has some troubles when implemented in SUMO, where Krauss is the default model and is used a lot more for research in SUMO [67]. Since both car-following models are evenly matched, but Krauss has the advantage in preventing implementation issues, the Krauss model is chosen for this thesis.

As with the IDM and other car-following models, the Krauss model has got some improvements over the years. This improved model is the default in SUMO and will be explained in this section. Like all other car-following models, the Krauss model determines its speed relatively to the vehicle ahead. This speed will be called the safe speed v_{safe}, and can be used to keep the right distance and a safe speed to the vehicle in front. The safe speed can be calculated using the following equation:

$$v_{\text{safe}} = -b(t_r + \frac{t_i}{2}) + \sqrt{b^2(t_r + \frac{1}{2}t_i)^2 + 2bg} \tag{2-21}$$

In this equation, b is the maximum deceleration of the ego vehicle, g is the headway to the leading vehicle in seconds, t_r is the drivers reaction time and t_i is the time it takes to actually press the brake until maximum deceleration is reached. This formula is based on the variation of deceleration of the whole braking process [49] where different braking phases have been analyzed. Since this 'safe' speed could exceed the maximum legal speed for that highway or can be larger than the highest reachable speed of the car given its maximum acceleration, another equation is necessary to prevent this kind of situations. This can be done using the following equation:

$$v_{\text{des}} = \min[v_{\text{max}}, v + at, v_{\text{safe}}] \tag{2-22}$$

As can be seen, the minimum of these three different values is taken, as the desired speed v_{des}. In Equation 2-22, the maximum legal speed is given by v_{max} and to keep the acceleration within the physical limits of the car, $v + at$ can be used. According to (Song J. et al. 2015), simulation results can conclude that this models outperforms other models in describing realistic human driving behaviour on the highway. This leads to good results in wait time, travel time, lane-change number and safe speed [49].

Where the Krauss model can only be used for car-following. A second model is necessary for the lateral behaviour of the human drivers in SUMO. This can be done using a lane changing model.

Lane changing model

While there is a lot of research done on car-following models and a lot of these have been created, to describe lane changing behaviour there are a lot less common models available. Since lane changing and deciding when to change lanes is a lot more complex than keeping a safe longitudinal distance or speed, and the lateral driving behaviour is also depending on the longitudinal behaviour, most lane changing models are more like a combination of different rules which try to describe the lane changing process well. To make it even more complex, not all human lane changing behaviour is exactly the same, since humans experience safety and possibilities different depending on their behaviour. Since there where some issues on implementing car-following models in SUMO, other than the default, it could be wise to chose the default lane changing model in SUMO, LC2013 [11]. This model is based on the older model, DK2008, and since it is the default in SUMO, it is often used for research in SUMO [18][73], where it delivers some good results.

The main idea of this model which will be explained in this thesis, is that it makes use of a 4-layered hierarchy of motivations to determine the lane changing behaviour of the vehicles during each time step. This motivations are based on the urgency to make a lane change. A lane change is considered urgent when the following relation is true [11]:

$$d - o < \text{lookaheadspeed} * \text{abs}(\text{bestlaneoffset}) * f \tag{2-23}$$

At the left side of this equation, d is the distance to the dead end of a lane, where o is a parameter which describes the occupation on the target lane. At the right side we have the look-ahead speed which is the presumed speed when driving near the point where a lane stops. Next the best lane offset is a parameter that describes the occupation of the best lane to switch to and finally, f is a factor that describes the time necessary to perform a successful lane change manoeuvre. This factor f is necessary since the amount of time necessary can vary, depending on a lot of difference circumstances and the fact that changing lanes to the left or right are not similar. When it is not possible to make the desired lane change, for example because the desired lane is blocked, the speed has to be adjusted according to the circumstances.

According to some research [11], this model has a better control over speed adjustments when the desired lane is blocked and also more deadlocks are avoided compared to other models. A deadlock is a situation where a vehicle has driven itself stuck in traffic and has to go with the flow to prevent any collisions from happening. Another strong aspect of this model are the choices made in overtaking other vehicles, which is better compared to other lane changing models.

2-2-3 Summary

This chapter was divided into two parts: the first part about the vehicle models most suitable to control the AVs, and the second part about the vehicle models that will describe human drivers as realistic as possible and can be used to control the human driving behaviour in SUMO. For the trajectory planning, done by the MPC, a kinematic linear bicycle model is used since this describes some realistic car behaviour, while keeping the computational load low by using an optimization problem with a quadratic cost function and linear constraints which will be explained in Chapter 2-4. To create the vehicle trajectories using the optimal inputs given by the MPC, a non linear version of the kinematic bicycle model will be used.
For the human driving behaviour, the Krauss model will be used to describe the longitudinal behaviour. Finally for the lane changing decision making, the LC2013 model which is also the default in SUMO, is selected.

2-3 Surrogate safety measures

When using the kinematic bicycle model, it is important to give it inputs that will not only control the vehicle realistically, but also safe. MPC can be used to optimize a certain cost function that creates a desired driving behavior. Next to driving on the right side of the highway when possible, it is also preferred to stay in the middle of your lane and keep a safe distance from the other road users to decrease the chance of accidents. The benefit of an autonomous vehicle is not only that it can react a lot quicker than a human driver could without getting tired or making mistakes, but also that the cost function can be shaped in a way that it can improve traffic quality, and thus help the other road users. An improvement in traffic quality could be measured in a lot of different ways, and could mean for example that the amount of accidents decreases. In this thesis the cost function and the constraints of the MPC will be designed in such a way that that the AVs will not only reach their destination as quick as possible, but will also try to decrease the chance an accident will occur and the severity of possible accidents. This can be realized by using traffic turbulence and different SSM. In the following subsections will be explained what traffic turbulence and SSM are, how these can be measured and how this can be used to improve the traffic quality.

2-3-1 Traffic turbulence

The first method that will be discussed that can describe traffic quality is the traffic turbulence. Because the total travel time of all road users together could increase when the traffic turbulence increases and traffic safety could decrease, it would be useful to prevent the AVs from making unnecessary lane changes and accelerating, where the latter is also called acceleration noise. Usually a highway is designed such that it allows high speeds under safe circumstances, because intersections are removed and replaced by on- and off-ramps. There are also multiple lanes, both traffic directions are separated and the bends in the highway are more gentle. By making use of ramp spacing, where on- and off-ramps are placed far enough from each other and using an extra lane, the traffic turbulence can be reduced. Since all these aspects that can improve traffic safety by creating a more predictable road are more costly, a consideration between the extra cost and traffic safety should be made.
Next to road design, there are also some traffic rules that should be followed to increase traffic safety. From example keeping below the legal speed limit and try to stay on the right lane as much as possible, unless overtaking. Because of this last rule, traffic will be separated based on driving speed which improves traffic safety. It is important to implement these rules into the behaviour of the AVs, to get a desired result. Some research shows by using GPS-equipped vehicles, that acceleration noise can be influenced by vehicle and driver characteristics, and that its degree of impact can be affected by the traffic levels [26]. Another study shows that when using acceleration noise as a standard deviation of accelerations resulted in the best measure to quantify congestion. This can also be used to study the impacts on fuel consumption and vehicle emissions.

Definition of traffic turbulence

While there is no real definition of traffic turbulence, according to (van Beinum A. 2018) [71], a more specific definition of turbulence can be given by: 'individual changes in speed, headways, and lanes (i.e. lane-changes) in a certain road segment, regardless the causes of the change'. Next the level of turbulence according to (van Beinum A. 2018) is given by: 'the frequency and intensity of individual changes in speed, headways and lane-changes in a certain road segment, over a certain period of time'. This definition could be translated to some mathematical equations and used in the cost function of the MPC. Since changes in speed, headways and lane changing always occur in traffic, there is always a minimum amount of traffic turbulence. By minimizing a cost functions which includes these, this thesis will combine them with SSM and show the benefits in the total travel time for the human drivers on the highway. The SSM will be further explained in the next subsections.

2-3-2 Surrogate Safety Measures

Next to the traffic turbulence, SSM can be used to improve the traffic quality by improving on traffic safety. Traffic safety has been studied mostly by using crash data. This data can be used to find the cause that has lead to a crash. While traffic safety has been studied a lot with the use of this crash data, it also has some limitations [1] [6]. As can be seen in 2-6, which describes the relation between severity and frequency of crashes, the real accidents are only the top of the pyramid, while traffic safety is a lot more than only the accidents that actually happened. By making dangerous decisions in traffic, dangerous situations can occur which could lead to an accident. This is why non-crash data could also be useful for

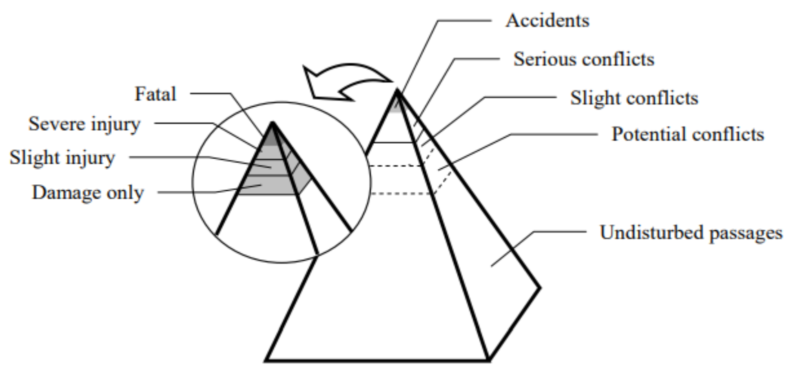

Figure 2-6: Safety pyramid [14]

analysing traffic safety and where SSM come into play. SSM are measurements which uses microscopic traffic parameters such as the headway in distance or time headway, or vehicle speed or acceleration. These parameters can be used to describe the relation between multiple vehicles during traffic events to indicate the chance and severity of a possible traffic accident. These traffic events could be anything, for example a lane change manoeuvre or two vehicles crossing each other closely at an intersection. The microscopic traffic parameters that can be used to analyse traffic safety can also be measured real time by for example radar by an autonomous vehicle. By understanding how to use this data, and implementing these measurements into the MPC, the autonomous vehicle could possibly decrease the frequency of accidents or decrease the severity. In this Thesis the MPC will make use of the Time To Collision (TTC) and Post-Encroachment Time (PET) which are both SSM. These will be explained in the following subsections.

Time To Collsion

The TTC is the time it takes when two vehicles, moving in the same direction, would make a collision when continuing with a constant speed, at which they are traveling at this moment. As said above, the size of the TTC can be used to measure the severity of a conflict. When a TTC value is low, that indicates that the probability of a collision is high. When the TTC value is higher, there is a smaller chance that a collision would occur. Usually, a TTC value of less than 1 second, is considered as a severe conflict [64]. To clarify this safety measure, Figure 2-7 shows two vehicles driving in the same lane. The leading vehicle in blue, has a velocity of 10 m/s while the following vehicle in black has a velocity of 15 m/s. Since the difference in speed is 5 m/s and the gap between both vehicles is 7.5 m, it will take 1.5 seconds until the black vehicles collides with the blue vehicle, thus the TTC is equal to 1.5 seconds. For human drivers it is important to keep the gap between the leading vehicle and themselves, large enough for them to react. Of course for autonomous vehicles, a safe TTC value could be a lot smaller since their reaction time is a lot smaller, depending on the sensors and computing power.
In the example in Figure 2-7, both vehicles are driving in the same lane. This safety measure could be used for an autonomous vehicle approaching another vehicle to decide when the headway gets to small and it has to change lanes, for example to overtake, or if it will decrease its speed instead. For this thesis, the TTC will be used as a time-varying constraint for the MPC to prevent collisions in the same lane. This constraint will be discussed in Chapter 2-4. When both vehicles are driving in a different lane, but heading into the same direction, it is not useful to calculate the TTC. For this reason another SSM can be used to determine if it is safe to change lanes when there are vehicles nearby in adjacent lanes. To make this decision, the PET will be used.

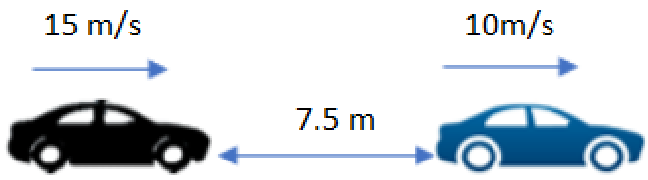

Figure 2-7: Time to collision

Post-Encroachment Time

For the AVs to decide if it is safe to make a lane change, the Post-Encroachment Time will be used. This is necessary since the trajectories of two vehicles driving in a different lane and staying there will never intersect. Where the TTC used the relative speed between two vehicles, the PET will use only the speed of a vehicle approaching another vehicle. When another vehicle is driving at an adjacent lane, an encroachment line perpendicular to the road as can be seen in Figure 2-8, can be drawn for example at the rear bumper where a collision could happen. The PET will be the time that it takes the ego vehicle, which is the vehicle that we are interested in, to cross that line. Whether the vehicle in the other lane will move and at which speed does not matter, since that encroachment line will stay at that exact location. Since the encroachment line is vertical and crosses all lanes, it also does not matter where the ego vehicle will cross this line. These properties make the PET more suitable to use as a SSM for collision avoidance while lane changing than the TTC. This concept is visualised in Figure 2-8. As can be seen in the figure, there are four different possibilities in which the PET can be used. This is shown with four different vehicles A-D where vehicle O is changing lanes. While vehicle O is changing lanes, it it possible to calculate the PET for all four vehicles. When the PET equals zero, the ego vehicle made contact with the encroachment line with a nearby vehicle. The higher the PET, the smaller the probability that a collision would occur. This means that the PET also indicates the severity of a conflict, similar to the TTC. But because of the differences between the TTC and the PET, the TTC would be more suitable for the longitudinal collision constraints, while the PET is more suitable for the lateral collision constraints, used by the AVs.

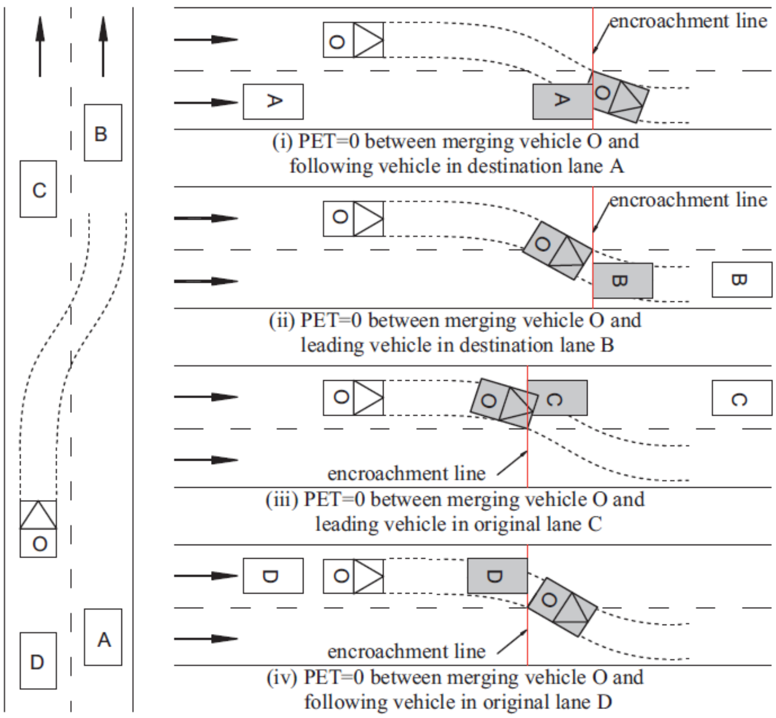

Figure 2-8: Post-Encroachment Time [52]

2-3-3 Summary

In this chapter, some possibilities to describe the behaviour of the AVs are discussed. The traffic turbulence, consisting of the acceleration noise and unnecessary lane changing behaviour could be used in the cost function to reduce for example the total travel time for all the vehicles in the scenario given in Chapter 2-1. For the collision avoidance, the TTC can be used for the longitudinal collision avoidance constraints while the PET can be used for the lateral collision avoidance constraints. How the cost function and constraints are designed will be discussed in Chapter 2-4.

2-4 MPC

Now that we have got all the ingredients for the MPC, we have to set up the optimization problem. First we will dive into the literature to explain the state of the art of MPC in traffic applications. A lot of research has been done on collision avoidance in highway scenario's, which could be useful for not reinventing the wheel again. Next we will give an overview of the optimization problem which has to be solved by the MPC. All the different components and some pseudo code will be shown to get a clearer view of the problem that has to be solved. After that we take a look at output MPC, which is necessary for reference tracking. This is important to make the AVs speed converge to a certain reference speed, which could be the legal driving speed for example, or make the lateral position converge to the right coordinate to prevent the vehicle staying in the left lane or in between different lanes. Finally the cost function and constraints used for this thesis will be discussed.

2-4-1 MPC in traffic applications

The first question that we have to ask is which form of MPC would be the most suitable for this thesis project. Because of its optimal solution, versatility and the reason that it can handle constraints, it is a popular method used in traffic applications. But since it is so versatile, it is important to analyse the state of the art to find out how MPC is mostly used in collision avoidance. The most common MPC methods used in traffic applications are distributed MPC, Robust MPC, non linear MPC, linear MPC and hybrid MPC. These different methods will be discussed in the following sections.

Distrubuted MPC

A lot of research has been done on distributed MPC [7]. Distributed MPC has the advantage that it is easy to scale. Since there are multiple MPC controllers distributed over multiple vehicles, they all have to solve a smaller part of an optimization problem. Usually these vehicles can communicate with other vehicles nearby, to share some variables. Hierarchical structures can be used to make this process more efficiently, which can be seen in Figure 2-9. In this figure, a lot of different levels of control are used to control different plants, which could be multiple AVs for this thesis. Despite the fact that distributed MPC is highly scalable and could be used for huge simulations, it increases the computational load for some smaller simulations. This method is mostly used in highway scenario's: for example for highway manoeuvres [19] or cooperative platooning [60].

Robust MPC

The next method of MPC that could be used is robust MPC. This has been proven to be successful when combining longitudinal and lateral driving behaviour, by ensuring that the given trajectory is feasible [36]. Robust MPC is mostly used when there are some uncertainties involved. For example weather influences or the uncertainty in decision making of nearby road users. These kinds of uncertainties can be handled by robust mpc and is mostly researched in highway scenario's for for example cruise control [4] and lane changing [30]. This method could also be used in urban areas [43].

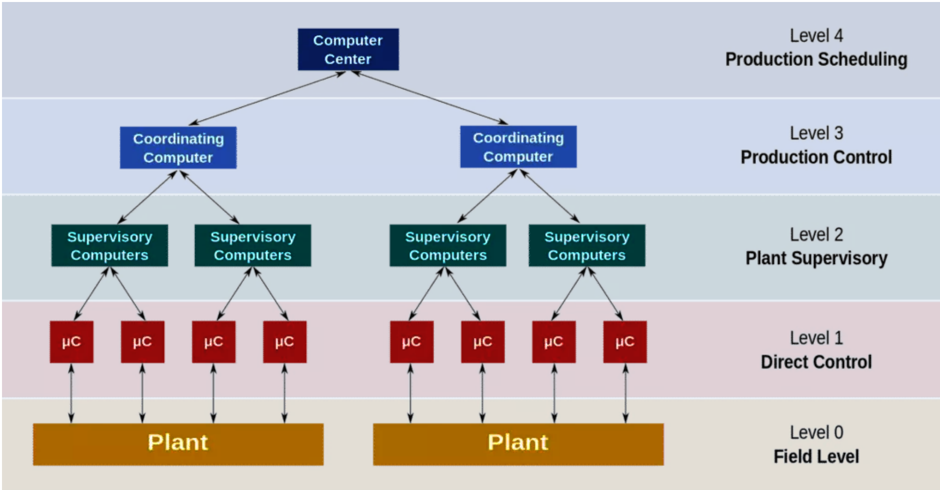

Figure 2-9: Hierarchical distributed control structure [70]

Non linear and linear MPC

Another form of MPC is non linear MPC [39]. Since this method needs more complex optimization methods to obtain the optimal solution and could get stuck in some local minima, it takes a lot more computational power to solve these kind of problems. This could be a problem since the MPC has to be run real time when applied to AVs. Because of its complexity, it could have the advantage of being more realistic than MPC with linearized models. Research shows that this is not always the case, and that some linear versions work equally well as non linear MPC [17]. Sometimes the non linear models can also be linearized to a linear but time-varying system [42]. Both linear and non linear models used for MPC can be used in highway scenario's and urban areas, since they are both very general.

Hybrid MPC

The last version of MPC discussed in this thesis that is applied in traffic scenario's is hybrid MPC. There are different forms of hybrid models, each with its own advantages, which could be useful when separating the longitudinal and lateral behaviour. Sometimes non linear models can be simplified by using Piecewise-affine (PWA) or Mixed Logical and Dynamical (MLD) functions [20] to reduce the computational load. But the real benefit is to use hybrid control to achieve results that can not be achieved by regular linear of non linear controllers [58].

MPC for this thesis

Many of these different methods discussed above have been used successfully for different kind of highway scenario's like obstacle avoidance [27][21], platooning [29] and minimizing fuel [22] consumption for trucks, and seem to work well in simulations. Often in these highway scenario's, the constraints used in any MPC method can be used to bound the physical limits of the vehicle and for passenger comfort [54].

Because the simulation used for this thesis is small and contains only a small number of vehicles at the same time, distributed MPC seems not the right choice. Where uncertainty is not necessarily in the scope of this project, and it is important to run a lot of different simulations with a simulation time of 1 hour, the simplicity of MPC using a linear model is preferred. As shown above, this could work equally well as the more complex non linear MPC, and we will use a non linear vehicle model to get more realistic driving trajectories. Because a linear state space model will be used, a quadratic programming problem can be formulated where a least squares algorithm will be used to find the optimal solution.
In the next section will be explained how this form of MPC using a linear model works.

2-4-2 The MPC problem

As explained earlier, MPC stands for Model Predictive Control. This means that this type of controller tries to predict the future, using an internal model. This model describes the dynamics of the system that has to be controlled. Usually the equations of a system are known, obtained by knowledge of the system or can be approximated by system identification [62]. Using this model, it can be predicted in which state you will end when using a certain sequence of inputs and knowing the initial state ξ_0. How many time steps in the future the MPC has to predict is determined by the prediction horizon. This is different in comparison with an Linear-Quadratic Regulator (LQR) controller which uses an infinite prediction horizon. In Figure 2-10 is shown how a discrete time MPC prediction horizon works.

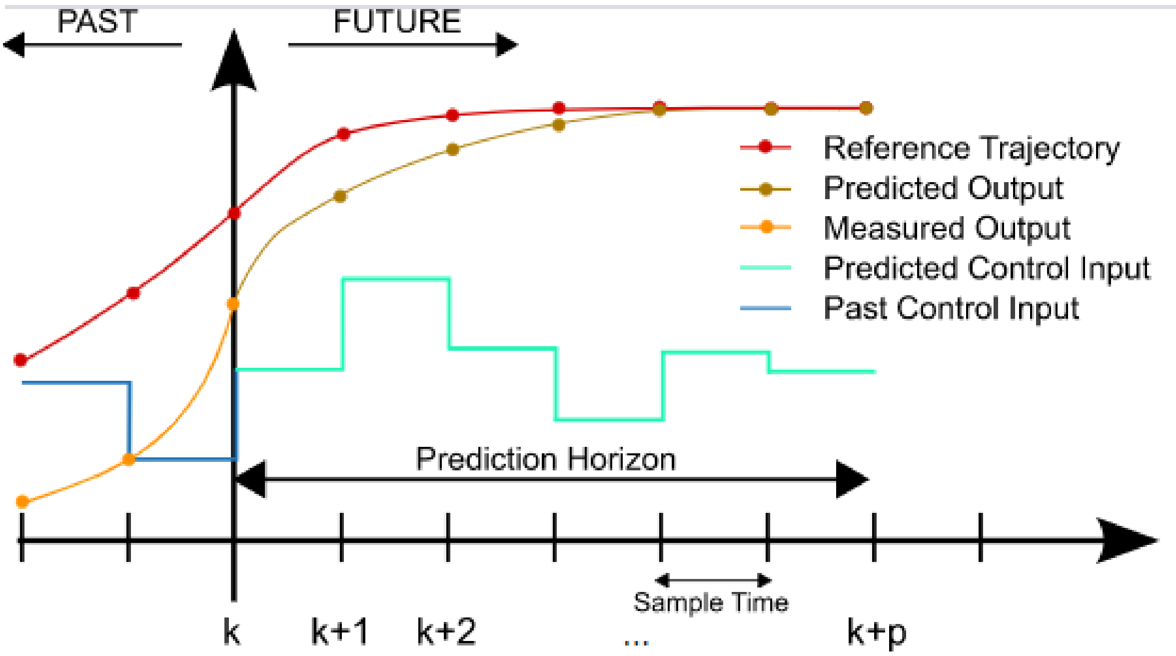

Figure 2-10: Discrete time MPC [56]

As can be seen from the figure, the current time step is k. At this point in time, the MPC makes a prediction of the states until a time step $k+p$ where p is the prediction horizon. To calculate the states in the prediction horizon, the intern model will be used. Using a cost function J_p based on the prediction horizon, which is a function depending on the input and

the different states within the control horizon, an optimal input u can be calculated. To tune the MPC, multiple weights can be used in the cost function to create the right driving behaviour. When the MPC minimizes the cost function while keeping the constraints into account, an optimal sequence of inputs is given. Since the MPC has to use these inputs real time, only the first input or inputs from this sequence at time step k are used. Then the controller will shift to the next moment in real time where it can make a decision again, and will predict the optimal solution for the new time step. This process will be repeated for as long as necessary.
To clarify a few things, the pseudo code of the optimal control problem is shown in Equation 2-24:

$$P_p(\xi_0, k) : \begin{cases} \min\limits_{\boldsymbol{u_p}} & J_p(\xi_0, \boldsymbol{u_p}) \\ \text{s.t.} & \xi(k+1) = A\xi(k) + Bu(k) \\ & F\xi(k) \leq e \\ & Gu(k) \leq a \\ & \xi(p) \in \mathbb{X}_f \end{cases} \tag{2-24}$$

In these equations, the cost function J_p has be be minimized using the initial state ξ_0 and a sequence of inputs $\boldsymbol{u_p}$ with its length depending on the prediction horizon p. From this input sequence $\boldsymbol{u_p}$, only the first inputs at the current time step will be used. In the case of this thesis, the inputs are the steering angle $\delta_f(k)$ and the longitudinal acceleration a(k). This optimization problem is subject to the system dynamics which are $\xi(k+1) = A\xi(k) + Bu(k)$ which is the linear kinematic vehicle model given in Chapter 2-2. Next there are state constraints $F\xi(k) \leq e$ and input constraints $Gu(k) \leq a$ and the terminal constraints $\xi(p) \in \mathbb{X}_f$. The terminal constraints are useful to guarantee stability. This happens when the final state at time instant p is inside a set of states $\mathbb{X}_f$.

Usually a controller steers the states to an equilibrium point which could be the origin for example. But sometimes you want to steer the states to a certain reference, which is shown in red in Figure 2-10. This could be a certain location, or a speed of 100 km/h for example. Using the past measured outputs, shown in yellow, the optimization problem has to be solved to converge the predicted output to the reference output. How this reference tracking works exactly, will be shown in the next section.

2-4-3 Output MPC

For this thesis, a reference of the desired longitudinal speed and the lateral position will be necessary. Using a reference for the longitudinal speed will make the AVss speed converge to the legal speed limit as fast as possible, while also taking the constraints into account. The lateral position is important for collision avoidance and staying in the middle of the lane. This kind of reference tracking can be done using output MPC, where we use the output η(k) to converge to the η_{ref}. Since $\eta_{ref} = C\xi_{ref}$, we also need a $\xi(k)$ that converges to ξ_{ref}, This will be a steady state where $\xi_{ref} = A\xi_{ref} + Bu_{ref}$, which means that to make the system converge to a certain reference output, a reference state and reference input are necessary. It is important for these references that the constraints are satisfied: $(\xi_{ref}, u_{ref}) \in \mathbb{X} \times \mathbb{U} =: \mathbb{Z}$ and $C\xi_{ref} \in \mathbb{Y}$. To find these target states and inputs, an Optimal Target Selection (OTS) is necessary. We can formulate this optimal target selection as an optimization problem:

$$(\xi_{ref}, u_{ref})(\eta_{ref}) \in \begin{cases} \underset{\boldsymbol{\xi_r}, \boldsymbol{u_r}}{\text{argmin}} & J(\xi_r, u_r) \\ \text{s.t.} & \begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} \xi_r \\ u_r \end{bmatrix} = \begin{bmatrix} 0 \\ \eta_{ref} \end{bmatrix} \\ & (\xi_r, u_r) \in \mathbb{Z} \\ & C\xi_r \in \mathbb{Y} \end{cases} \tag{2-25}$$

In this optimization problem, we use the η_{ref} as given parameter. For the cost function J, we chose $J = \frac{1}{2} u_r^T R_{ref} u_r$ where the R_{ref} is a weight matrix used to weight the different inputs. For this thesis, just the identity matrix is used as reference for the input weights. This cost function is subject to $\xi_r = A\xi_r + Bu_r$ and $C\xi_r = \eta_{ref}$ which is written differently in Equation 2-25 to obtain a sequence of reference states and inputs. For these sequences of reference states and inputs it is important that the constraints are satisfied.

This optimization problem could be solved offline, when the η_{ref} which consists of a y_{ref} and v_{xdes}, is given and constant. Since this is not the case for this thesis it has to be solved again when the reference changes. The y_{ref} will be selected using the following algorithm [28]:

$$\eta_{ref}(k) = \begin{bmatrix} v_{xdes}(k) \\ y_{ref}(k) \end{bmatrix} \tag{2-26}$$

$$y_{ref}(k) = \underset{y_{CL}}{\text{argmin}}(y(k) - y_{CL}) \tag{2-27}$$

where

$$y_{CL} \in \{y_{CL1}, y_{CL2}, ...\} \tag{2-28}$$

Here, y_{CLi} is the y-coordinate of the center line of each lane and y(k) is the y-coordinate of the ego vehicle. This algorithm is necessary to make the vehicle drive in the middle of the lane where it is currently driving. The reference for the longitudinal speed v_{xref} will be constant at 100 km/h = 27.78 m/s, which is the legal speed limit. To steer the ego vehicle to an adjacent lane for overtaking or to stay right, the cost function and constraints are necessary which will be explained in the next sections.

2-4-4 Cost function

The cost function of the MPC will describe the behaviour of the vehicle. Since we want the vehicle to drive in the middle of the lane and keep a certain driving speed, we include these in the cost function. The quadratic cost function J_p of this thesis will be shown below:

$$
\begin{aligned}
J_p = \min_{u_1,u_2} \quad & \sum_{k=0}^{p-1} q_1(y(k) - y_{ref}(k))^2 + q_2(v_x(k) - v_{xdes}(k))^2 \\
& +q_3\psi(k)^2 + P\xi(p)^2 + r_1u_1(k)^2 + r_2u_2(k)^2
\end{aligned} \tag{2-29}
$$

where

$$
Q = \begin{bmatrix} q_1 & 0 & 0 \\ 0 & q_2 & 0 \\ 0 & 0 & q_3 \end{bmatrix} \quad R = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix} \tag{2-30}
$$

Here Q and R are the weight matrices, where Q will be used for the states and R for the inputs. P is the weight matrix used for the final state $\xi(p)$ at the prediction horizon, and is the solution of the Discrete Algebraic Riccati Equation (DARE). This equation is given by:

$$
P = (A + BK)^T P(A + BK) + (Q + K^T RK) \tag{2-31}
$$

$$
K = -(B^T PB + R)^{-1} B^T PA^T \tag{2-32}
$$

and depends on the system matrices A and B of the state space model given in Chapter 2-2 and the weight matrices Q and R.
The cost function in Equation 2-29 will be minimized using the inputs u_1 and u_2. The first 2 terms at the right hand side will be used to make the vehicle go to the middle of the lane which is y_{ref} and tries to keep its longitudinal speed $v_x(k)$ as close to $v_{xdes}(k) = 27.78$m/s as possible. The $q_3\psi(k)^2$ will have the effect that the heading angle $\psi(k)$ will converge to zero to make the vehicle drive straight. Finally, $r_1u_1(k)^2$ and $r_2u_2(k)^2$ will penalize accelerating and steering when overused, to prevent the vehicle from wasting fuel and keeping the acceleration noise as low as possible. While this creates the preferred driving behaviour, we also expect the AVs to prevent colliding with other vehicles and staying on the road. This can be done by using constraints, which will be explained in the next section.

2-4-5 Constraints

For the constraints we make a distinction between 3 different kind of constraints. First the input constraints will be explained. Next the state constraints will be shown to keep the vehicle within certain bounds, and finally the collision constraints are discussed. As a recap, the LTI state space model being used is given by:

$$
\xi(k+1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & v_{const}\Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x(k) \\ y(k) \\ \psi(k) \end{bmatrix} + \begin{bmatrix} \Delta t & 0 \\ 0 & v_{const}\frac{l_r}{l_f+l_r}\Delta t \\ 0 & \frac{v_{const}}{l_f+l_r}\Delta t \end{bmatrix} \begin{bmatrix} a(k) \\ \delta_f(k) \end{bmatrix} \tag{2-33}
$$

where $\xi(k)$ are the states, and u(k) the inputs.

Input constraints

The first constraints are the input constraints. To make sure that the inputs are realistic and don't exceed the physical limits of the vehicle some constraints are necessary. It is also important to keep the maximum acceleration low to increase comfort for the passengers driving the vehicle [8]. When the maximum acceleration and deceleration are to high, the AVs looks more like a roller coaster, but when they are too small, some dangerous situations could occur. For this reason the same values used by the Krauss model in SUMO for the acceleration will be used:

$$-2.6 \leq u_1(k) \leq 3 \tag{2-34}$$

Where these constraints are in m/s^2. Next for the steering angle δ_f, the following constraints are used:

$$-0.02 \leq u_2(k) \leq 0.02 \tag{2-35}$$

which are both in radians. These values are very small because driving on the highway involves small steering inputs to make sure that the lateral acceleration stays within bounds [37].

State constraints

Next the state constraints are used to keep the vehicle within certain bounds. Since we want to use the cost function to create the right driving behaviour, the legal driving speed is not used as a state constraint. Instead a physical limit for the driving speed is used:

$$0 \leq \xi_1(k) \leq 40 \tag{2-36}$$

which is 40 m/s. A minimum of 0 m/s is used to prevent the vehicle to drive in reverse. Next some constraints are necessary to keep the vehicle within the road boundaries. Of course the MPC tries to keep the vehicle in the middle of the lane, but for safety reasons and preventing the vehicle from unfeasible lane changes the road boundaries are included in the state constraints [28]:

$$y_{min} + \frac{1}{2}L_y \leq \xi_2(k) \leq y_{max} - \frac{1}{2}L_y \tag{2-37}$$

In this equation, L_y is the width of the vehicle, and y_{min} and y_{max} are the road boundaries. These constraints are also used when there is a vehicle in one of the adjacent lanes. When another vehicle is within sensors range L_{sc}, and within a safe range $L_{sx,PET}$, that lane will be blocked to prevent the ego vehicle from lane changing and colliding with another vehicle. The $L_{sx,PET}$ can be determined by:

$$L_{sx,PET}(k) = v_x(k)\mathrm{t}_{PET} \tag{2-38}$$

In this equation, the longitudinal speed of the ego vehicle will be used and multiplied by a safe PET t_{PET} in seconds, to obtain a distance $L_{sx,PET}$, which can be used to block an adjacent lane when another vehicle is too close to perform a successful lane changing manoeuvre to that lane. This means that the y_{min} or y_{max} from Equation 2-37 can be changed when there is a vehicle in the lane next to the ego vehicle [28]:

$$y_{max} = \begin{cases} y_{max} - L_{sy} & \text{if } |x_{ri}(k)| \leq L_{sx,PET} \\ y_{max} & \text{otherwise} \end{cases} \tag{2-39}$$

or a vehicle in the lane right to the ego vehicle:

$$y_{min} = \begin{cases} y_{min} + L_{sy} & \text{if } |x_{ri}(k)| \leq L_{sx,PET} \\ y_{min} & \text{otherwise} \end{cases} \tag{2-40}$$

Where L_{sy} is the lane width and x_{ri}(k) the relative longitudinal distance between both vehicles. When a low t_{PET} is chosen, the distance $L_{sx,PET}$ gets smaller depending on the longitudinal speed of the ego vehicle. This parameter can be adjusted to change the minimum PET the AVs should keep from vehicles in an adjacent lane.

Collision avoidance constraints

The last constraints that need to be discussed are the collision avoidance constraints. For these constraints, we will make use of the relative distances and speeds between the ego vehicle and the leading vehicle in the same lane. These relative distances and speeds can be described using the following equations [28]:

$$v_{xri}(k+1) = v_{xri}(k) + (a_{xi}(k) - u_1(k))\Delta t \tag{2-41}$$

$$x_{ri}(k+1) = x_{ri}(k) + v_{xri}(k)\Delta t \tag{2-42}$$

$$y_{ri}(k+1) = y_{ri}(k) + (v_{const}\psi(k) - v_{const}\frac{l_r}{l_f + l_r}u_2(k))\Delta t \tag{2-43}$$

Here the relative x- and y-coordinate between both vehicles are given by $x_{ri}(k)$ and y_{ri}(k), where the relative speeds in x- and y-direction are given by v_{xri}(k) and v_{yri}(k). The a_{xi}(k) is the acceleration of the leading vehicle. This can be used to create the following time varying collision avoidance constraints [28]:

$$x_{ri}(k) \pm \frac{L_{sx}}{L_{sy}}y_{ri}(k) \leq -Lsx \tag{2-44}$$

The $\pm$ is a minus sign when the left lane is available for overtaking, and a plus sign when the right lane is available. The constraints are also shown in Figure 2-11 for clarification.

In Figure 2-11, the host vehicle which is the ego vehicle is driving in the right lane, where a slower driving vehicle S_1 is driving in the same lane. The collision avoidance constraints described by Equation 2-41 - 2-43 create a diagonal line as shown in Figure 2-11. The steepness of this line depends on the L_{sy} which is chosen to be the lane width in this thesis, and the L_{sx}, which can be described by the following equation:

$$L_{sx}(k) = v_{xri}(k)\mathrm{t}_{TTC} \tag{2-45}$$

The L_{sx} depends on the relative longitudinal speed between both vehicles, multiplied by the TTC t_{TTC}. The TTC will be chosen and varied in this thesis to analyse its effects on the

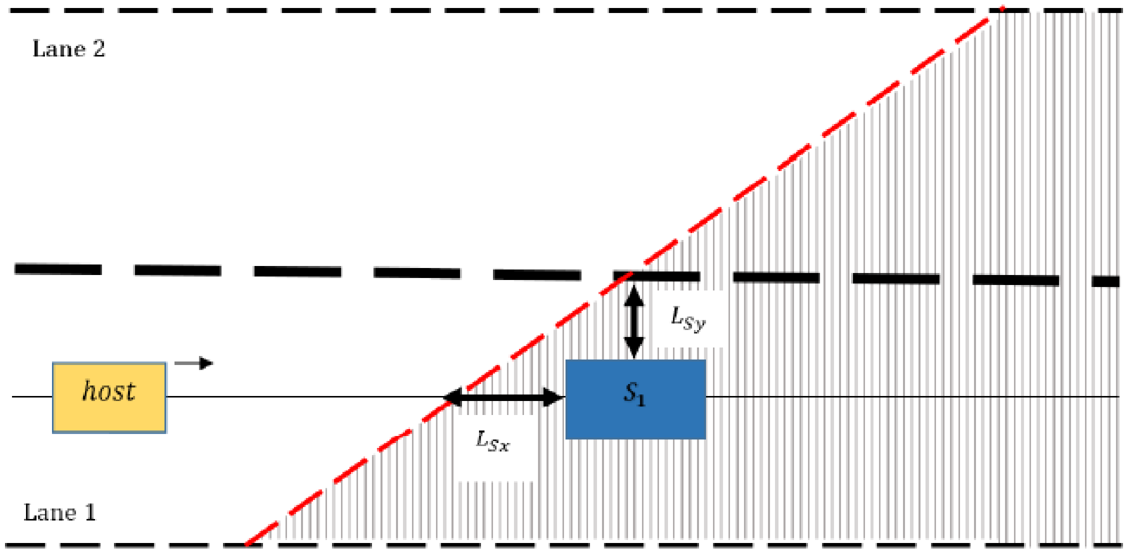

Figure 2-11: Collision avoidance constraints [28]

travel time of the other road users. This diagonal constraints are used to force the vehicle to the left lane to overtake the slower vehicle. When the ego vehicle enters the left lane, the constraint disappears and the lateral reference coordinate will be changed to the middle of the left lane. Now the right lane will be blocked by the state constraints described above, by adjusting the road boundaries. When the yellow ego vehicle has passed the blue vehicle by a distance $L_{sx,PET}$, the right lane will be unblocked and the ego vehicle can change lanes to the right again, by changing the lateral reference coordinate. The y_{ref} will always be the middle of the lane where the AVs is currently driving, unless the right lane is unblocked and it is possible to keep right.

For the change in relative y-distance between both vehicles, used in Equation 2-43, only the lateral speed of the ego vehicle is used and not the lateral speed of the leading vehicle. If the lateral speed of the leading vehicle was also used, the collision avoidance constraints would push the ego vehicle driving on the left lane off-road when getting cut by another vehicle coming from the right lane, because the constraints keep moving upward. While using the lateral speed of the leading vehicle works in the research of (M. Jalalmaab et al. 2015)[28], this does not work when using other vehicles which are also changing lanes.

2-4-6 Summary

In this chapter we have discussed the state of the art for MPC applied in traffic scenario's. A linear state space model (the linear kinematic bicycle model) will be used together with a quadratic cost function and linear constraints, such that a least squares algorithm can be used to solve the optimization problem. Next the reference tracking, used by the MPC is explained and the necessary input and state constraints are discussed. Finally the time-varying collision avoidance constraints are shown to get a complete overview of how the MPC controlling the

AVs in this thesis works. In the next chapter we take a look at the results when running different scenario's in SUMO.

Chapter 3

Results

Now that the complete MPC and all components necessary for the simulations are discussed, the results can be obtained. First, an easy scenario with one human driver and one AVs will be run in a simulation to test if the controller can overtake a vehicle as expected. Next, the scenario as discussed in Chapter 2-1 will be run with multiple AVs. This is necessary to answer the research question. First is tested whether lowering the PET and TTC has any effects on the human drivers in the simulation. The next batch of simulations has to show whether an increase in penetration rate increases the effect on the human drivers. And finally will be tested if the traffic flow influences effect on the human drivers, given the same penetration rates.

3-1 Testing the controller

The first test for the MPC is to overtake another vehicle, when driving on a highway with 2 lanes. In this scenario the cyan AV is starting at the right lane with an initial speed of 100 km/h. On the same lane, another blue vehicle is driving with a constant speed of 80 km/h, starting 100 meter ahead, as can be seen in Figure 3-1.

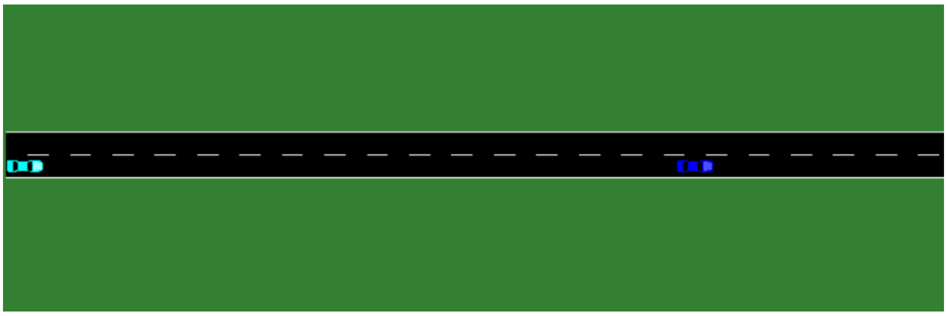

Figure 3-1: Overtaking a slower vehicle

The objective of the AVs is to overtake the slower vehicle, while keeping its speed constant. For this test a PET of 1 second is used and a TTC of 5 seconds is used. For the prediction horizon, 10 time steps are used. When tuning the controller, the weights used in the cost function that obtained the best results are given in Table 3-1.

q_1	200
q_2	0.02
q_3	10
r_1	1.5
r_2	200

Table 3-1: Weights used for the overtaking test

Running the simulation using this set up shows that the AVs has no trouble keeping the desired longitudinal speed, while overtaking the slower vehicle successfully. As can be seen in Figure 3-2, the constraints push the vehicle to the left lane to avoid a collision. As soon as the AVs crosses the boundary to the left lane, the lateral reference position changes to the center line of the left lane. When the AVs has passed the other vehicle and the PET > 1, the reference position switches back to the middle of the right lane as can be seen in the figure.

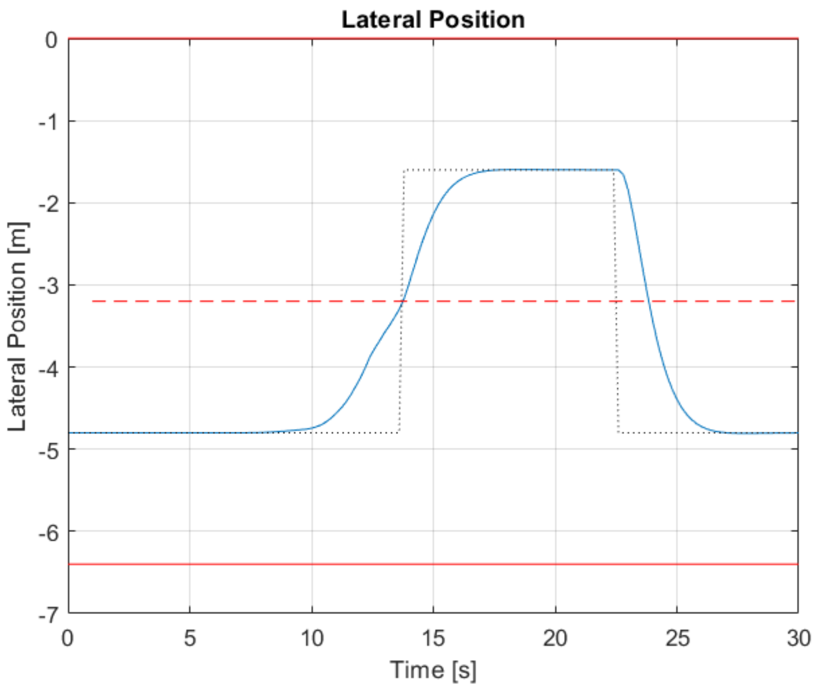

Figure 3-2: Lateral position

In Figure 3-3 the change in steering angle is illustrated. The AVs seems to have some trouble choosing the right steering angle which could be caused because of the time-varying constraints or the fact that the non linear model used in the simulation is not exactly the same as the linear model used for the trajectory planning. When the AVs crosses the line from the right to the left lane and the lateral reference position is changed, the vehicle fluidly steers to the middle of the left lane. This lane changing manoeuvre was executed while staying away from the maximum steering angle which is 0.02 radians. When the AVs performs its second lane change back to the right lane, a more fluid steering motion is used.

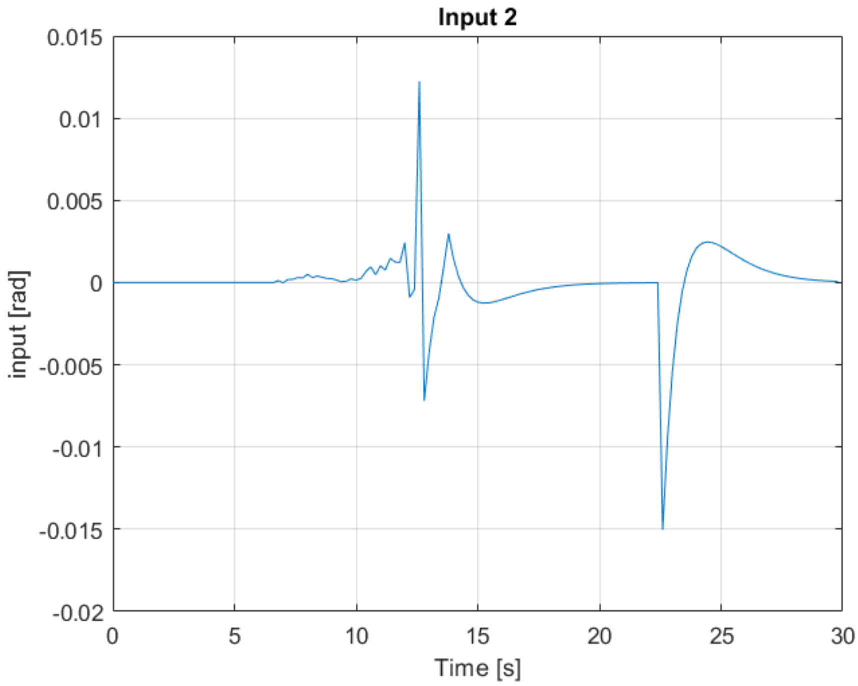

Figure 3-3: Steering angle

The effect of the steering angle on the vehicle angle is shown in Figure 3-4. Where the first steering action was not that fluid, a non fluid change in vehicle angle can be seen. This first lane changing manoeuvre consists of 2 parts. First when the collision avoidance constraints are used and the vehicle is pushed to the left lane, we see an increase in vehicle angle, and a small decrease in vehicle angle to steer the vehicle back. Then around 13.5 seconds when the vehicle passes the line to the left lane, the reference is changed and the vehicle makes a fluid motion to the middle of the left lane. When the lateral reference is changed for the second time to return to the right lane, also a more fluid graph of the vehicle angle can be seen.

Although the steering inputs could describe a more fluid motion, the effects on the lateral position are small since the steering angles are also small (0.02 rad = 1.15 degrees). When tuning the controller differently, it is possible to get a more fluid steering motion which results in a more fluid vehicle angle. This comes at the cost of increasing the time necessary for a lane changing manoeuvre.
The controller succeeds in keeping the longitudinal speed constant, because the first input which is the longitudinal acceleration or deceleration is unused. In the next sections, some larger and more complex simulations are run to test the effects of a smaller TTC and PET.

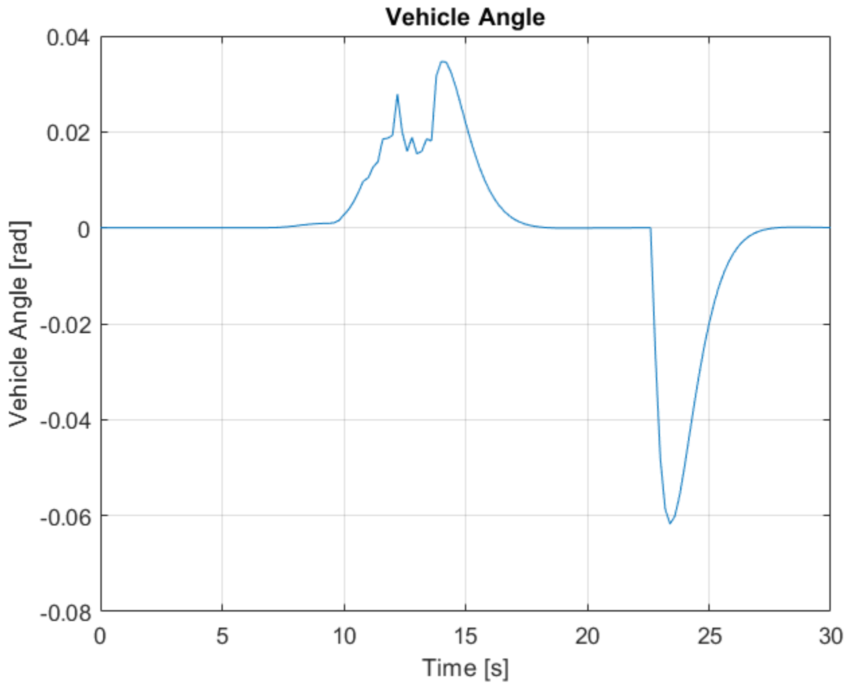

Figure 3-4: Vehicle angle

3-2 Full simulation

Now that we have seen that the MPC is successful in overtaking other vehicles without colliding, the full simulation with the scenario described in Chapter 2-1 can be run. To clarify some things, the scenario is shown again in Figure 3-5, as can be seen below.

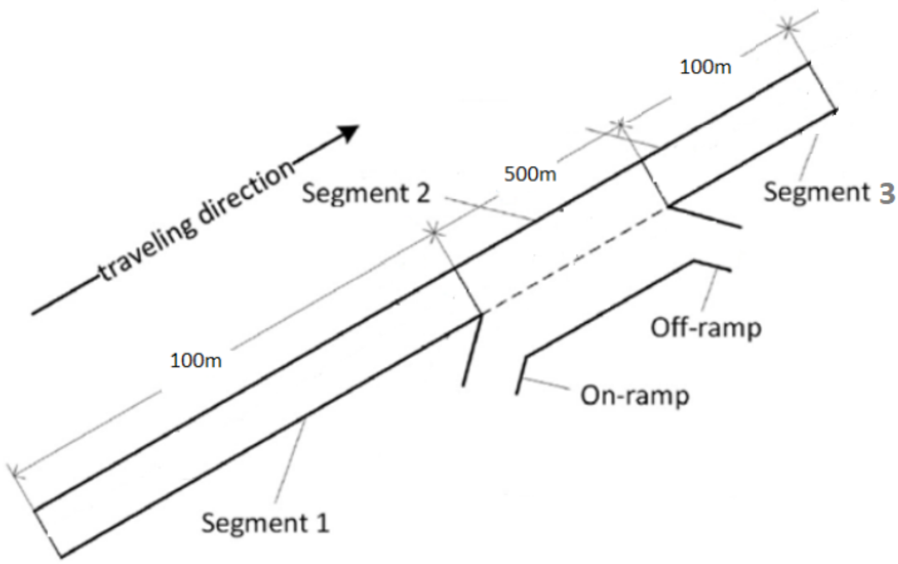

Figure 3-5: Recap highway scenario

When the vehicles in the simulation enter the highway from Segment 1 or the on-ramp, they enter segment 2 which has a length of 500 meter and consists of 2 lanes. For the first 400 meter on segment 2, the AVs will stay on the right lane unless the MPC decides to overtake, for example because it wants to keep a constant speed. After 400 meter, the MPC will switch its lateral reference to the left or right lane, depending on its destination.
At the start of the simulation, the highway is completely empty. When the simulation starts and the first vehicles enter the simulation, they are created at the start of segment 1 or on the on-ramp with an initial speed of 100 km/h. Different vehicle flows are used, and the vehicles are randomly distributed over the full simulation time. For all the simulations, a warm up round of 10 minutes will be used to make sure that the traffic is stabilized enough to create a realistic scenario. After that 10 minutes, a complete simulation of 1 hour will be run where all the data will be measured. The first tests have to point out whether lowering the PET or TTC has a positive result.

For the human drivers controlled by SUMO, the parameters used are displayed in Table 3-2.

Length	5.00 m
Min gap	2.50 m
Max speed	55.56 m/s
Max acceleration	2.60 m/s^2
Max deceleration	4.50 m/s^2
Sigma	0.50
Tau	1.00 s

Table 3-2: Parameters used for the human drivers

The length of the vehicles in SUMO is 5m, and the min gap when standing still is 2.50m. The sigma describes the driver imperfection which is given by a value between 0 and 1. A sigma of 0 denotes perfect driving. Finally the tau is the drivers desired (minimum) headway in seconds. These are the default parameters used in SUMO and should represent realistic human driving behaviour.

When running the first tests of the full simulations, it can be concluded that some simplifications are necessary. While the idea of time-varying constraints that takes the acceleration of the other vehicles into account seems good for some simple simulations, it did not work for the more complex highway scenario's with a lot more vehicles involved. When the ego vehicle is driving at the left lane, and another human driven vehicle in front of you also changes lane to your lane in front of you, the collision avoidance constraints will push you off road since the constraints keep moving with the predicted lateral speed of the vehicle changing lanes in front of you. These time-varying constraints had to be made constant in lateral direction to prevent this from happening in more complex simulations. For this reason, the collision avoidance constraints in Chapter 2-4 are described without the influence of the lateral acceleration of another vehicle.

3-2-1 Lowering the PET and TTC for AVs

For these simulations, a vehicle flow of 3200 vehicles per hour will be used. It is expected that when the flow is high, the benefits of the extra mobility gained by decreasing the PET and TTC will be easier to show. Realistic headways on highways can vary between less than a second and 2-3 seconds depending on the vehicle density [65][41], and other research shows that realistic PET and TTC values are also in the range of less than a second to 2-3 seconds [33][23].
First we will keep the minimal TTC at 0.1 seconds with a varying minimal PET for the AVs. Next some simulations will be run with a PET at 0.1 seconds and a varying TTC. As a recap, the PET is used for lateral collision avoidance, and the TTC is used for the longitudinal collision avoidance. The results of these simulations are shown in Table 3-3 and Table 3-4 and illustrated in the scatter plots in Figure 3-6 and Figure 3-7. In these tables and figures the average travel time of the HVs and AVs is shown.

PET [s]	TTC [s]	Average Travel Time HVs (σ)[s]	Average Travel Time AVs (σ)[s]
1.5	0.1	25.805 (0.769)	25.185 (0.280)
1	0.1	25.756 (0.766)	25.196 (0.281)
0.5	0.1	25.681 (0.749)	25.184 (0.285)
0.1	0.1	25.655 (0.729)	25.188 (0.262)

Table 3-3: Changing the PET for a constant TTC at a penetration rate of 10% and a flow 3200 vh/h

PET [s]	TTC [s]	Average Travel Time HVs (σ)[s]	Average Travel Time AVs (σ)[s]
0.1	3	25.596 (0.692)	25.211 (0.292)
0.1	2	25.609 (0.695)	25.196 (0.290)
0.1	1	25.643 (0.722)	25.189 (0.296)
0.1	0.1	25.655 (0.729)	25.188 (0.262)

Table 3-4: Changing the TTC for a constant PET at a penetration rate of 10% and a flow 3200 vh/h

The first thing that can be noticed from the tables, is that the average travel times of the AVs are in all cases smaller than the average travel times of the HVs. This is caused by the fact that the AVs can keep a shorter TTC or PET from other vehicles than the HVs. Next we take a look at the effect that a decrease in a minimal PET or TTC used by the AVs has on the HVs. As can be seen from both tables, when using a flow of 3200 vehicles per hour and a penetration rate of 10%, decreasing the PET while keeping the TTC constant, a decrease in average travel time for the human driven vehicles has been measured. Also the standard deviation σ is shown between brackets next to the average travel time. As can be seen from the table, the effect of decreasing the PET on the average travel time of the human drivers is very small. Although the effect is small, for every decrease in PET for the AVs, the average time of the Human Vehicles (HVs) is decreased. This was expected since keeping a smaller PET gives more flexibility in overtaking. Since the decrease in average travel time for the HVs is very small at a penetration rate of 10%, lowering the PET for AVs would not benefit

HVs that much in a real world situation. Although the HVs do not benefit by a significant decrease in travel time, it is important that is shown that the faster travel times of the AVs are not at the cost of the HVs.

Next we take a look at different values for the TTC used by the AVs. When decreasing only the TTC while keeping the PET constant, a small increase in average travel time by all human drivers in the simulation is measured. As can be seen in Figure 3-8, this effect is smaller than when the PET is decreased and the TTC is kept constant. It was not expected that a decrease in TTC for the AVs, would lead to an increase in travel time for the HVs, since this should also give the AVs more possibilities to overtake because a shorter headway is allowed. This result could occur due to other behaviours of the AVs which have a larger impact on the average travel time of HVs than decreasing the TTC. One of these possible behaviours could be caused by the collision avoidance constraints. Since the overtaking constraint gradually pushes the AV to the left lane when overtaking on the right lane, the steepness of this constraint depends on the allowed TTC used by the AV. It could be the case that when programming collision avoidance using linear constraints as done in this thesis, the AVs hinder the other road users less when a higher TTC is used, because this reduces the steepness of this constraint and will create a slower and more predictable overtaking manoeuvre which makes it easier for human drivers to adjust.
To compare the effects on the HVs between decreasing the TTC and the PET for the AVs, both figures are combined in Figure 3-8.

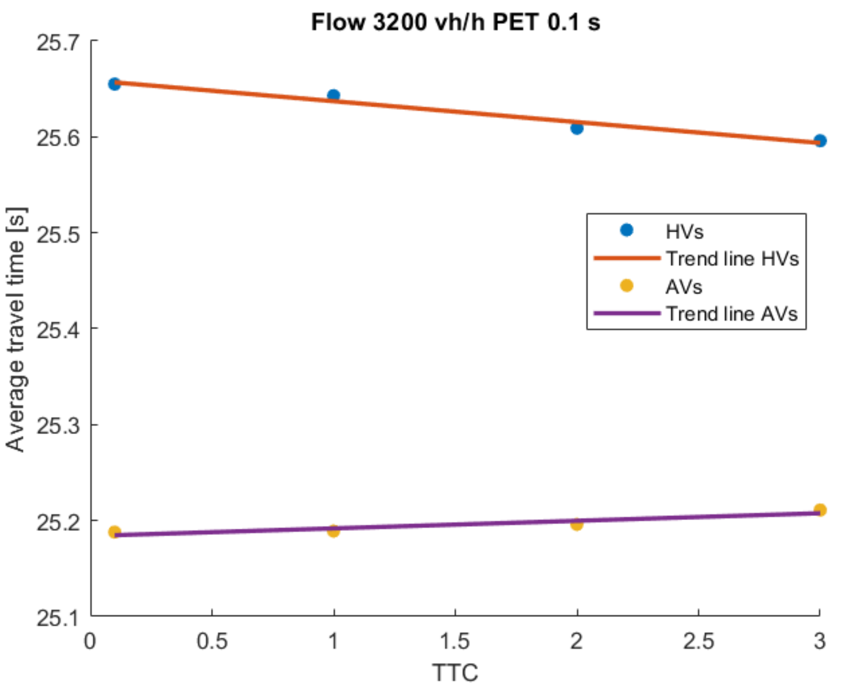

Figure 3-6: TTC scatter plot of 3200 vh/h with a penetration rate of 10%, comparing the average travel times of the HVs with AVs

The trend lines show the trend when both SSM vary. As can be seen in Figure 3-8, the effects of decreasing the TTC are smaller compared to decreasing the PET.

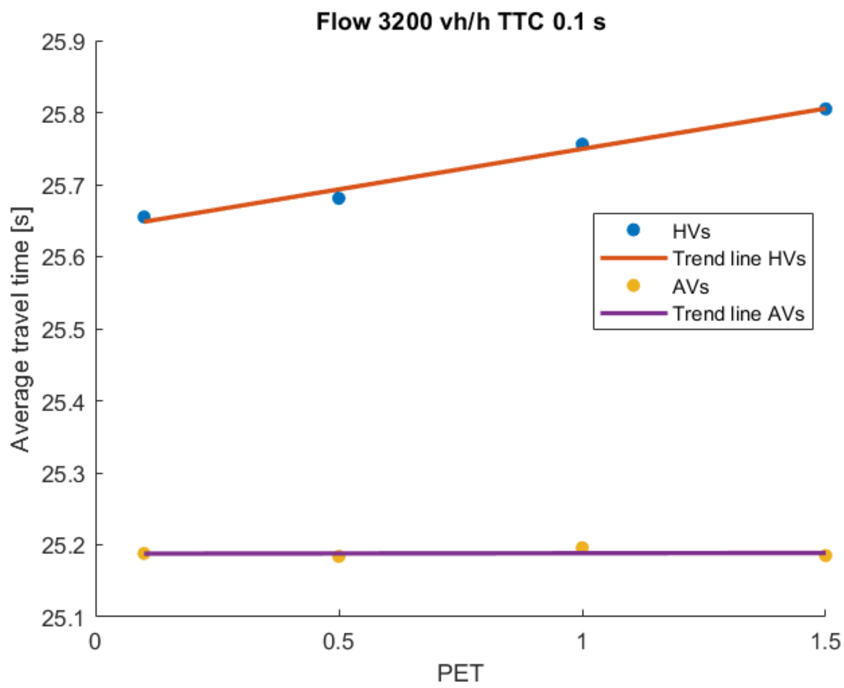

Figure 3-7: PET scatter plot of 3200 vh/h with a penetration rate of 10%, comparing the average travel times of the HVs with AVs

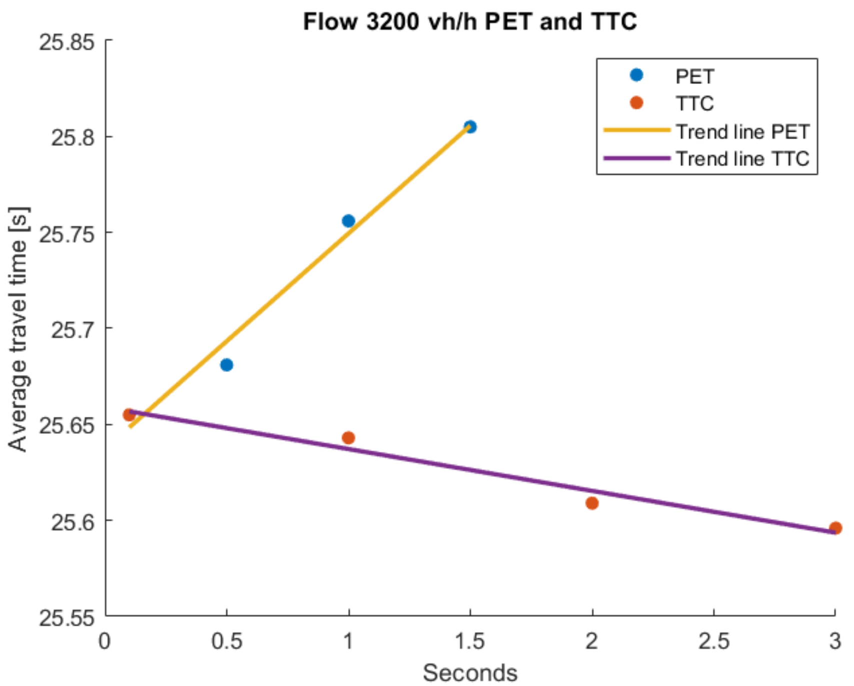

Figure 3-8: Combined PET and TTC scatter plot of 3200 vh/h with a penetration rate of 10%, showing the average travel times of the HVs

3-2-2 Increasing the penetration rate

The next simulations have to show the effect of an increased penetration rate. A penetration rate of 0%, when there are no AVs driving in the simulation, will be used for comparison against simulations with different penetration rates. A value of 0.1 seconds for both the PET and TTC will be used by the AVs to test different penetration rates at a flow of 3200 vehicles per hour. The results are shown below in Table 3-5.

Penetration rate [%]	Average Travel Time HVs (σ) [s]	Average Travel Time AVs (σ) [s]
0	25.717 (0.677)	-
5	25.660 (0.687)	25.164 (0.247)
10	25.655 (0.729)	25.188 (0.262)
15	25.616 (0.765)	25.179 (0.282)
20	25.521 (0.745)	25.149 (0.265)
30	25.485 (0.742)	25.138 (0.264)
40	25.429 (0.821)	25.128 (0.298)

Table 3-5: Different penetration rates for 3200 vh/h with a PET and TTC of both 0.1 s

As can be seen from the table, an increase in penetration rate decreases the average travel time of the human drivers in the simulation. This is also shown in Figure 3-9.

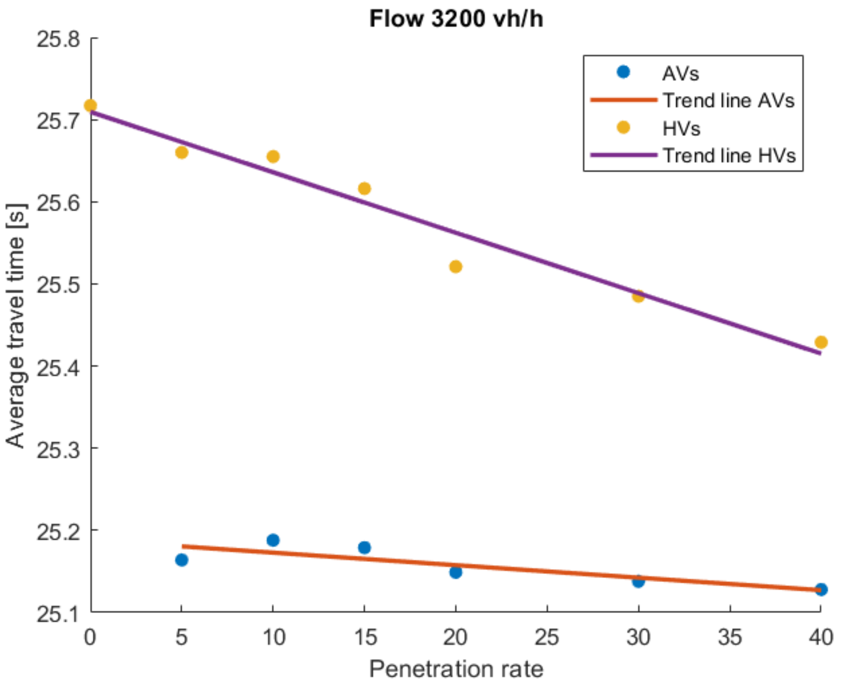

Figure 3-9: Different penetration rates for a flow of 3200 vh/h

Although this decrease in average travel time is relatively small, even at a penetration rate of 5%, a decrease in average travel time for the HVs can be noticed. When increasing the penetration rate even further to 40%, a decrease of 1.120% in average travel time has been measured, compared to the situation with only HVs and no AVs. The trend is shown by the trend line in Figure 3-9. This figure shows that that the decrease in average travel time for the AVs at different penetration rates is not at the cost of the average travel times of the HVs.

To show the distribution of the travel time of the human drivers compared to the AVs and the complete traffic flow, a box plot is shown in Figure 3-10. The average travel time and standard deviation are also shown Table 3-6.
The box plot shows that the median (the red horizontal line) of the travel time of AVs is lower than that of the HVs. The table also shows that the average travel time is smaller for the AVs than for the HVs. The boxplot shows that the standard deviation σ is a lot smaller for the AVs than for the HVs because they are all programmed with the same driving behaviour while human driving behaviour varies more. The data of the HVs has more outliers while the data of the AVs is spread closer around the median.

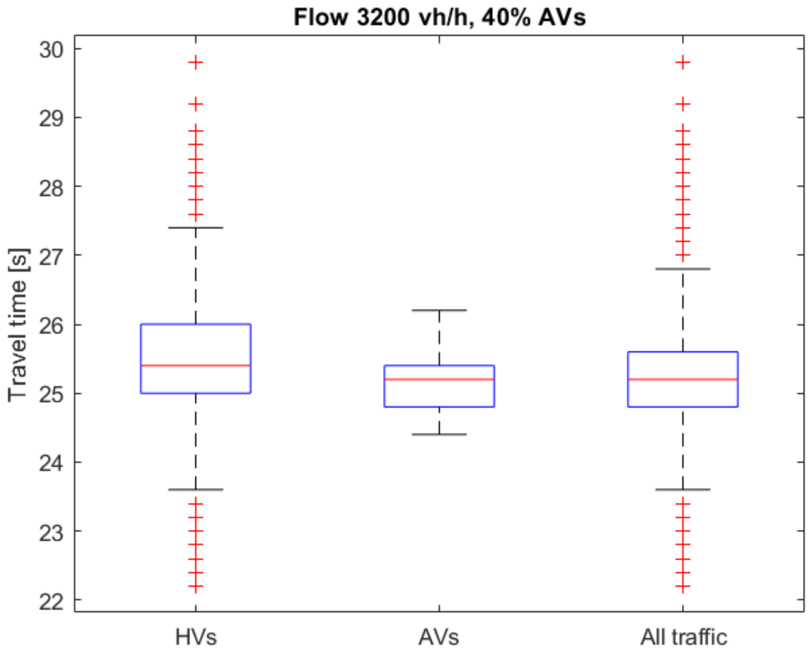

Figure 3-10: A boxplot comparing the travel time for Human Vehicles, AVs and both together for a flow of 3200 vh/h at a penetration rate of 40%

Vehicles	Penetration rate [%]	PET [s]	TTC [s]	Average Travel Time (σ)[s]
HVs	60	-	-	25.429 (0.821)
AVs	40	0.1	0.1	25.128 (0.299)
All traffic	100	-	-	25.311 (0.682)

Table 3-6: The average travel time for Human Vehicles, AVs and both together for a flow of 3200 vh/h at a penetration rate of 40%

3-2-3 Changing the traffic flow

The last simulations have to show the effect of a decrease in traffic flow. When the traffic flow of 3200 vehicles per hour is used, the density of vehicles is relatively high. When the traffic flow is smaller and not as close to the roads maximum capacity, it is expected that vehicles have a lot more space to switch lanes or overtake other vehicles, which could decrease the effect of using a small TTC and PET. The next simulations will be run with different traffic flows at a penetration rate of 0%. The results from these simulations can be compared with simulations using a penetration rate of 10% and a PET and TTC of 0.1 seconds. Using data with no AVs and 10% Avs at each traffic flow, the improvement can be calculated. The results are shown below in Table 3-7 and Table 3-8.

Flow [vh/h]	Average Travel Time (σ) [s]
1800	24.630 (1.110)
1900	24.655 (1.044)
2000	24.718 (1.115)
2100	24.775 (1.001)

Table 3-7: Average travel times at 0% penetration rate for different flows

Flow [vh/h]	P. rate [%]	PET [s]	TTC [s]	Average Travel Time (σ) [s]	Improvement [%]
1800	10	0.1	0.1	24.637 (1.106)	-0.028
1900	10	0.1	0.1	24.659 (1.052)	-0.016
2000	10	0.1	0.1	24.711 (1,135)	0.028
2100	10	0.1	0.1	24.757 (1.015)	0.073

Table 3-8: Improvement for different vehicle flows using a 10% penetration rate

Where a 10% penetration rate resulted in a decrease of the average travel time at a flow of 3200 vehicles per hour, the effects seem different at smaller vehicle flows. This result is also shown in Figure 3-11.

As can be seen from the figure at both 1800 and 1900 vh/h, a penetration rate of 10% has a negative effect on the average travel time of the human drivers. At a vehicle flow of 2000 vh/h, we have measured a positive effect. This effect is increased when the traffic flow increases further to 2100 vh/h. Where the maximum capacity of the scenario used in this thesis is at 4830 vh/h, positive effects on the average travel time of other road users can be measured above around 40% of this maximum capacity, using the AVs from this thesis.

The negative effects at the lower traffic flows could be explained by the fact that a certain amount of the human driven vehicles controlled by SUMO exceeds the legal speed limit. Where the legal speed limit is 27.78m/s, an overview of the average driving speed at the traffic flows of 3200 vh/h and 1800 vh/h, both with no AVs are shown in the histogram in Figure 3-12. As can be seen in the figure, many vehicles exceed the legal speed limit at both traffic flows. This happens because SUMO assigns different speed factors to individual vehicles, such that 95% of the vehicles drives between 80% and 120% of the legal speed limit

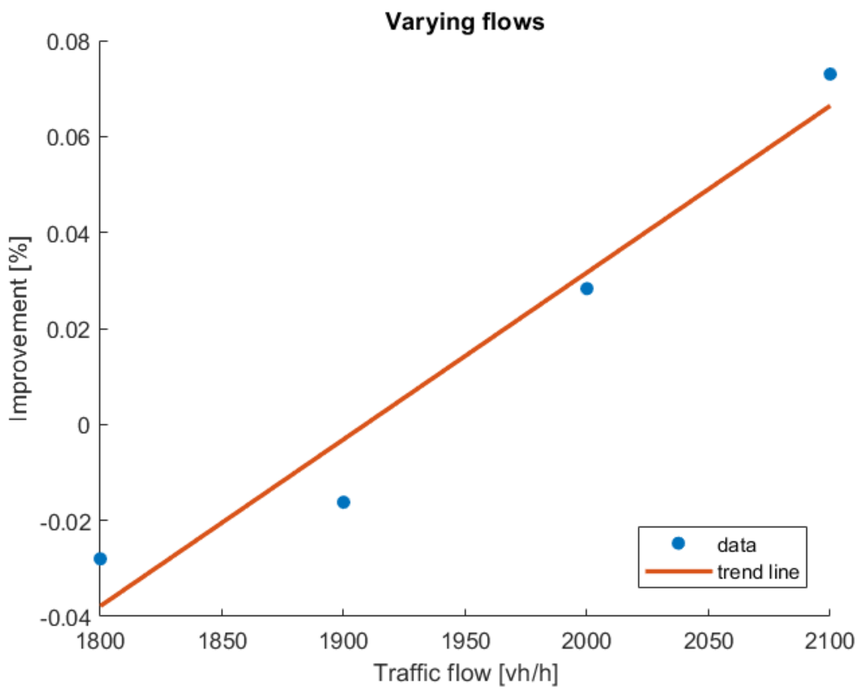

Figure 3-11: Different vehicle flows and their improvement at a penetration rate of 10%

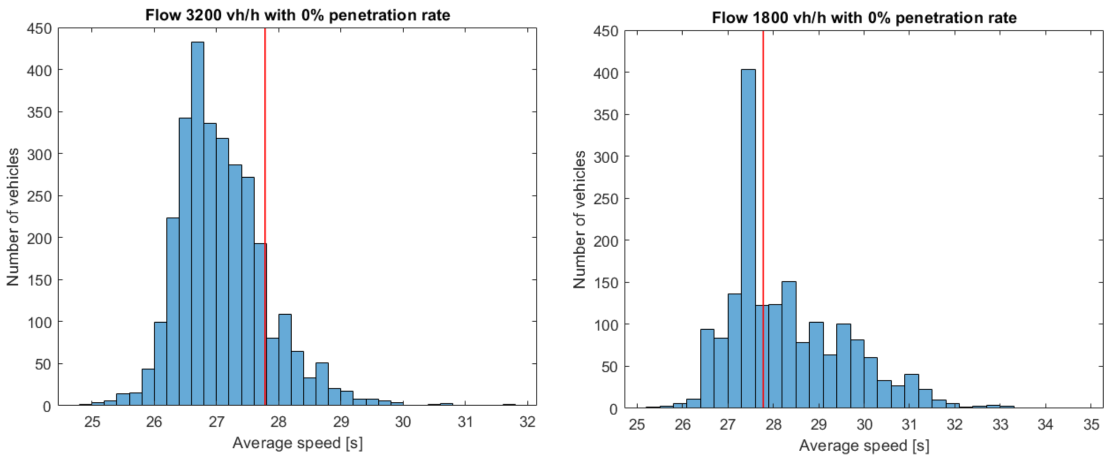

Figure 3-12: A histogram of the average vehicle speed of the vehicles controlled by SUMO at 3200 and 1800 vh/h. The red line is the legal speed limit of 27.78m/s.

[66]. According to SUMO, having a distribution of speed factors is beneficial to the realism of a simulation. People exceeding the legal speed limit is also common in the real world. This could be a reason why the AVs have a negative effect at lower traffic flows, since none of the AVs exceeds this limit in any of the simulations in this thesis.
Where the average speed of the human drivers at a flow of 3200 vh/h is 27.11 m/s when there are no AVs involved, the average speed increases to 28.34 m/s at a flow of 1800 vh/h when there are no Avs involved. This means that the average speed of the human drivers at 1800 vh/h is higher than that of the Avs, since the human drivers drive faster than the legal speed

limit. This means that the AVs will slow down the human drivers as a result. For the vehicle flows of 1800 and 1900 vh/h, this effect seems bigger than that of the decreased PET, having a negative effect as a result.

3-3 Summary

In this section, we have been analysing the results of this thesis. In the first section, a successful overtake manoeuvre of the MPC is shown. This was later used in more complex and bigger simulations to answer the research question: How does decreasing the PET and TTC, used in the behaviour of autonomous vehicles, influences the travel time of the whole simulation? Some more complex simulations showed that decreasing the PET has a positive effect, while decreasing the TTC has a small negative effect, which could occur due to the linear collision avoidance constraints. Next was shown that an increase in penetration rate has a positive effect on the average travel time of the human drivers, which increases when the penetration rate increases. Finally has been shown, that the effects mentioned above where only present when the density of vehicles was high. When the traffic flow is low and the vehicles have enough space to switch lanes or overtake other vehicles, the effects were different. Although it is questionable how much AVs would actually help HVs in highway scenario's, it is important to get a good overview of the implications on traffic performance when AVs start driving gradually on the highway next to HVs. In the next chapter, we take a look at the summary of this thesis, the conclusion that can be made from the results, and we discuss some interesting changes that could be applied for further research.

Chapter 4

Discussion and Conclusion

In this chapter we start with a summary of this thesis. An overview is given of all the different choices that have been maid during the process of this research. Next we will discuss the results, the research question will be answered and some conclusions can be made. Finally the there is a section with future challenges and recommendations which will point out how to progress on these results and how the MPC can be improved to obtain an even better controller.

4-1 Summary

In this work, an MPC is created to control multiple AVs in a simulation containing an on-ramp and off-ramp scenario, to research the implications on traffic performance. To run the controller and the simulations containing both AVs and human drivers, SUMO and Matlab are used. To create a realistic human driving behaviour, the Krauss car-following model is used to describe the longitudinal driving behaviour for all the different simulations. Next the LC2013 lane changing model is used for the lateral driving behaviour. These human drivers are all controlled by SUMO. To control the AVs, MPC is used because of its optimal solution and for the reason that it can handle constraints, which is useful for collision avoidance. The kinematic bicycle model has been used because it can describe the non holonomic driving behaviour realistically, while being simple enough to keep the computational load low. A Linear Time Invariant (LTI) state space model has been used by the MPC for the trajectory planning, next to a quadratic cost function and linear, time-time varying constraints for collision avoidance. To actually control the AVs by realistic trajectories using the optimal inputs given by the MPC, a non linear kinematic bicycle model has been used to show that using a linear kinematic bicycle model in the MPC for trajectory planning gives some accurate results. The input constraints of the MPC are used to make sure that the physical limitations of the vehicle (maximum acceleration and steering) are not exceeded, and driver comfort is secured. While the time-varying linear constraints for collision avoidance have its limitations, they have been proven effective in successfully overtaking vehicles in highway scenario's. Finally some more complex simulations are run on an on- and off-ramp scenario, where is shown that lowering the PET while still guarantying traffic safety by not lowering it beyond the minimum of 0.1 seconds, has a positive effect on the average travel time of the human drivers when the vehicle density is relatively high. Also increasing the penetration rate of AVs using this decreased PET has a positive effect on the average travel time of the human drivers under the same circumstances. Although the effects were very small, they were consistently shown running simulations of 1 hour simulation time.

4-2 Discussion

The goal of this thesis was to investigate the implications on traffic performance when integrating an MPC and the behavioural rules of human drivers into a numerical simulation. This is important because situations with mixed traffic consisting of both AVs and HVs become realistic when autonomous vehicles start driving on the highways gradually. To reach this goal, the following research question should be answered:

What are the implications on traffic performance when implementing an MPC and human drivers' behavioural rules into a numerical simulation?

The main contribution of this thesis, is the implementation of the MPC and the human drivers' behavioural rules into a numerical simulation. To measure the effects of the MPC on the performance of the mixed traffic scenario, we first decreased the PET and TTC which the AVs should keep for collision avoidance. A minimum allowable PET is used by the AVs for lateral collision avoidance which unblocks the adjacent lane for overtaking. A minimum allowable TTC is used for longitudinal collision avoidance to decide whether the AV has to slow down or can overtake when it approaching a leading vehicle. The traffic flow used for these simulations is 3200 vehicles per hour. While the MPC is not specifically designed to improve the traffic flow, decreasing the minimum allowable PET and TTC did decrease the average travel time of the other road users, while also decreasing the average travel time of the AVs itself. This thesis shows us that, when decreasing the PET while keeping the TTC constant, a small but positive effect on the travel time of other road users can be measured, when using a penetration rate for the AVs of 10%. The next step was to also decrease the TTC while keeping the PET constant. Decreasing the TTC, which is used for longitudinal collision avoidance, increased the average travel time of other road users. This could be caused by other driving behaviours of the AVs, probably due to the linear collision avoidance constraints. The increase in average travel time of the HVs caused by decreasing the TTC of the AVs was small compared to the decrease in average travel time of the HVs when decreasing the PET of the Avs.

Because the decrease in average travel time of the human drivers is very small, it is unlikely that AVs could be used to help HVs decrease their travel time, when using an MPC as used in this thesis. When changing the linear collision avoidance constraints, it could be possible to get a decrease in travel time for the HVs when decreasing the TTC used by AVs for longitudinal collision avoidance. For example, more complex non linear constraints could be used to change the driving behaviour of the AVs, as are used in (M. Ammour et al. 2021) [57]. The downside of an approach with non convex constraints is an increase in computation time.

While this could work for a single AV, running simulations with hundreds or even thousands of AV would increase the computation times even more.
It is also possible to change the cost function to help the other road users even more. Where the time-varying linear constraints that are used for collision avoidance depend on the relative speeds and distances between the AV and nearby cars, these relative states could also be used in the cost function of the MPC. In this case the cost function can influence the other road users directly instead of indirectly by tuning the weights.

While the decrease in average travel time of the HVs by decreasing the PET of AVs is already shown at a penetration rate of 5%, this effect gradually increases when the penetration rate is further increased, as was expected. At a penetration rate of 40%, a 1.120% decrease in average travel time for the other road users is measured at a flow of 3200 vehicles per hour. This decrease is too small to be meaningful for human drivers in the real world. What is more important is that can be shown how human drivers react when driving in mixed traffic situations next to AVs. We can conclude that the AVs reach their destination quicker than their human counterparts when the traffic is relatively dense. And although the decrease in average travel time for the HVs as a result of using different penetration rates of AVs is small, we can exclude unforeseen negative effects.

Finally, when using the on- and off-ramp scenario described in this thesis, positive effects on the average travel time of other road users can be measured above around 40% of the maximum capacity, using the AVs from this thesis at a penetration rate of 10%. When the traffic flow is further increased above 40% of the maximum capacity, the positive effects improve. When the traffic flow is below 40% of the maximum capacity, other driving behaviours like the maximum speed of the AVs seem to play a bigger role than decreasing the PET and TTC used for collision avoidance. It can be seen that in realistic human driving behaviour, speeding above the legal speed limit happens. This happens mostly outside rush hours when the traffic flows are low. When the vehicle density is low and less overtaking and lane changing is necessary, the speed at which a vehicle is driving influences the average travel time more than keeping a smaller headway does. When AVs are programmed to drive below the legal speed limit and a percentage of the HVs do not, the AVs will slow down these human drivers. This could be prevented when the decision making of the AVs is improved, for example when they can detect a vehicle approaching with a high relative distance compared to its own speed. This could be improved when implementing relative states in the cost function of the MPC. These will be described further in the recommendations section.

4-3 Conclusion

We can conclude that implementing the MPC and the behavioural rules of human drives into a numerical simulation with mixed traffic driving in an on- and off-ramp highway scenario gives a good overview of the effects that AVs have on the other road users. At high traffic flow situations, AVs have a lower average travel time than HVs because they can keep smaller headways than the human drivers. Next to this benefit for the AVs, lowering the PET that the AVs keep for lateral lane changing behaviour also decreases the average travel time of the HVs slightly. When the penetration rate increases, the average travel time of the HVs is decreased even further. While this is a positive effect of AVs, the decrease in travel time is very small and would not be a significant benefit for human drivers in a real world situation. For the longitudinal collision avoidance, linear constraints are used to keep a minimal TTC from a possible leading vehicle. Because of these linear constraints, a decrease of the minimal TTC that an AV should keep from another vehicle, leads to a very small increase of the average travel time for other road users. This could be explained by the behavioural rules of the AVs caused by the linear constraints used in this thesis. When the traffic flow is relatively low we found that the effects caused by AVs keeping a low minimal headway are different since the maximum speed and other behavioural rules play a more important role.

4-4 Future challenges and recommendations

For this thesis a simple algorithm is used, but it could be made a lot more complex to see if this would decrease the travel time of the other road users even more. In this thesis, the cost function of the AVs takes care of only the ego vehicle itself. It could be useful to create a cost function that uses, like the time-varying constraints with relative positions and speeds, more relative states. In this case the cost function could have a positive effect on the other vehicles directly, since the equations describing how the AVs will influence other vehicles are integrated into this cost function and can be optimized. This would be different from the method used in this thesis where only the states describing the dynamics of the AVs are integrated into the cost function, and the indirect effects of the AVs on the other road users are measured when the simulation is finished. By using the relative states in the cost function, we could improve the average travel time of the other road users even more. For example when a faster vehicle is approaching the AV (which is overtaking a vehicle) from behind on the left lane. The state $v_{xri}(k)$ that described the relative longitudinal speed between both vehicles could be used with the weight q_4 to converge the speed of the AV with the speed of the approaching vehicle. This is shown in the cost function below:

$$
\begin{aligned}
J_p = \min_{u_1,u_2} \quad & \sum_{k=0}^{p-1} q_1(y(k) - y_{ref}(k))^2 + q_2(v_x(k) - v_{xdes}(k))^2 \\
& \quad + q_3\psi(k)^2 + P\xi(p)^2 + \mathbf{q_4 v_{xri}(k)^2} \\
& \quad + r_1 u_1(k)^2 + r_2 u_2(k)^2
\end{aligned} \tag{4-1}
$$

When the weight q_4 is larger than the weight q_2, it will overrule converging to the desired speed and will converge to the speed of the approaching vehicle. In this case the AV prevents blocking the lane for the approaching vehicle, and will return to the right lane when it has finished its overtaking manoeuvre. Using this relative state in the cost function will probably also result in exceeding the legal speed limit for the AV. Where this is usually undesired, it might be desired in cases as described above to improve traffic safety. It is also possible to prevent that the AV exceeds the legal speed limit in such cases by making some changes in the decision making process of the AV before starting the lane change manoeuvre.

Next there are also some parts which have been simplified for this thesis. While the idea of time-varying constraints that take the acceleration of the other vehicles into account seems good for some simple simulations [28], it did not work for the more complex highway scenario's with a lot more vehicles involved. When another vehicle makes a lane changing manoeuvre and ends within sensor range in front of an AV, the constraints of the MPC will push the AV of the road since the constraints are moving with the predicted lateral speed of the vehicle changing lanes in front of you. These time-varying constraints had to be made constant in lateral direction to prevent this from happening in more complex simulations.

Where the simulation times were 1 hour in simulation time, the actual time running these simulations using higher penetration rates could easily exceed 3-4 days. This depends on the computational power of the computer and is caused by the fact that multiple AVs need to be controlled at the same time, which takes more time than running the MPC for one AV which could be done in real time. Especially the communication between SUMO and Matlab, and the fact that Matlab has to run the optimization algorithms takes a lot of time. Next to using a faster computer, it would be useful to integrate the MPC in a simulation program to decrease communication between various applications. This will increase the amount of data that can be gathered for a larger variation of simulations.

Finally the AVs can only overtake on the left side, but not on the right side. In certain situations, driving behaviour could be improved when overtaking on the right side is allowed. For example in traffic jams or when pre-sorting to the right lane. This could be done by running two different MPC problems, and mirroring the collision constraints over the longitudinal line. This would give two optimal solutions from which one could be chosen. When taking the final destination into account, it could be necessary to also add a path finding algorithm to decide which solution to chose.

Appendix A

Matlab Files

A-1 Main File

```
clc
clear all

% Parameters
lr = 2.1;                          % [m] Distance front axle to CoG
lf = 2.1;                          % [m] Distance rear axle to CoG
min_speed = 0;                     % [m/s]
max_speed = 40;                    % [m/s]
car_width = 1.8;                   % [m]
car_length = 5;                    % [m]
lane_width = 3.2;                  % [m]
roadbounds = [-6.4 -3.2 0];        % [m] lower, middle, upper bound
max_acc = 2.6;                     % [m/s^2]
max_dec = 3;                       % [m/s^2]
min_delta = 0.02;                  % [rad] = 1.1459 degrees
max_delta = min_delta;             % [rad]
Lsy = lane_width;                  % [m]

% Control data
h = 0.2;                           % Sample time
T = (3600+600)/h;                  % Simulation horizon
dim.N = 10;                        % Control horizon
t = 0:h:T*h;                       % Time vector for plots

% Weight matrices and PET
flow = 3200;                                          % Main flow
p_rate = 0.30;                                        % Penetration rate
safe_PET = 0.1;                                       % Allowable PET
safe_TTC = 0.1;                                       % Allowable TTC
weight.Q = diag([200 0.02 10]);                       % Weight on states: vx,
    y, theta
```

```
weight.R = diag([1.5 200]);                    % weight on input: a, delta

% Output only x1-2 for reference tracking
LTI.C = eye(2,3);
LTI.D = zeros(2,2);
[dim.ny,dim.nx] = size(LTI.C);        % order or the system
[~,dim.nu] = size(LTI.D);             % output dimension

% Input constraints
inputc.G = [eye(dim.nu*dim.N);-eye(dim.nu*dim.N)];
inputc.a = [repmat([max_acc;max_delta],dim.N,1);repmat([max_dec;min_delta],dim.N,1)];
Z1 = kron(eye(dim.N),eye(2,3));                              % State constraints on only x1 and x2

% Create a list of the AV
[AV] = randAVlist(flow,p_rate);                        % Creates a list beforehand to check which vehicles in the simulation need to be controlled by the MPC
p(1:length(AV)) = true;                                % Variable to measure only the first state
%% SUMO

% Start Scenario
[scenarioPath,~,~] = fileparts(which(mfilename));
cd(scenarioPath);
traci.start('sumo-gui -c ./FlowScenario3200.sumocfg --step-length 0.20 --lateral-resolution 1.6 --tripinfo-output Data.xml --start'); % This runs the right configuration file containing the simulation details. Next the step length is given. The lateral resolution is necessary to switch to the sublane model to make lane changing behaviour vissable. By default vehicles teleport when they perform a lane change. The tripinfo saves the data into an XML file. Finally the simulation will be started.
traci.gui.setSchema('View #0',  'real world'); % Changes the view to make the simulation look like the real world
% SUMO Loop.
for i = 1:T                               % This is a for loop controlling the time steps of the simulation
    traci.simulation.step();              % This makes the simulation go to the next time step

    % Vehicle List
    vehicles = traci.vehicle.getIDList();    % Creates a list of all vehicles currently in the simulation

    for k = 1:length(AV) % Loop to go through the AV list

    %% Measurements ego vehicle
    if any(strcmp(vehicles,AV{k})) % Check if the AV needs to be controlled, by checking if the vehicle ID is on the AV list created above
```

```
    angle = (traci.vehicle.getAngle(AV{k})-90)*(pi/180); % Change the coordinate system that SUMO uses to the system that is preffered for the vehicle model
    lane = traci.vehicle.getLaneID(AV{k});
    ego_lane = traci.vehicle.getLaneIndex(AV{k});
    edgeID = traci.lane.getEdgeID(lane);
    pos = traci.vehicle.getPosition(AV{k});              % Contains both the x- and y-position
    speed = traci.vehicle.getSpeed(AV{k});
    acc = traci.vehicle.getAcceleration(AV{k});
    Lsx2 = car_length+speed*safe_PET;                     % Safe range for lane changing
    v_const = max(speed+acc,0.01);                        % Update speed for A matrix (max function necessary to solve the DARE)
    traci.vehicle.setColor(AV{k},[255,0,0,255]);          % Change the vehicle colour to red

    if pos(1) > 100 && pos(1) < 570                  % Check if the AV is at the right road segment to be controlled
    if p(k) == true                                  % Only measure the state the first time, is necessary because the trajectory planning tries to predict and SUMO has data describing the old trajectories
        x(:,k) = [speed+acc;pos(2);angle];           % current state
        p(k) = false;                                % To make sure the measuring happens only once for each vehicle
    end
    x_0 = x(:,k);                                    % Select the right initial state

    %% Measurements nearby vehicles
    j = length(vehicles);                                  % Amount of vehicles in the simulation
    no_overtake_up = false;                                % Unblock left lane
    no_overtake_down = false;                              % Unblock right lane
    collision_constraint = false;                          % Set the parameter for detection vehicles in the same lane to false which means no vehicles detected
    xr = 65;                                                   % Sensor measuring distance
    for l = 1:j
        pos2 = traci.vehicle.getPosition(vehicles{l});         % Position other vehicle
        other_lane = traci.vehicle.getLaneIndex(vehicles{l});  % Lane index other vehicle

        % Collision constraints, vehicle same lane
        if pos(1) < pos2(1) && (pos2(1) - pos(1))< 65 && ego_lane == other_lane
            speed2 = traci.vehicle.getSpeed(vehicles{l});      % Speed x direction other vehicle
```

```
            acc2 = traci.vehicle.getAcceleration(vehicles{1}); % Acceleration other vehicle

            xr = pos(1) - pos2(1);                              % Relative x position
            yr = pos(2) - pos2(2);                              % Relative y location
            vxr = speed - speed2;                               % Relative speed x direction

            x_0r = [xr;vxr;yr;angle];                           % Initial relative states
            Lsx = max(car_length+vxr*safe_TTC,car_length);      % Safe longitudinal distance
            Z2 = kron(eye(dim.N),[1 0 -Lsx/Lsy 0]);             % Collision constraint transformation matrix
            d = kron((1:dim.N)',[0;-acc2*h;0;0]);               % Disturbance matrix for collision constraints
            collision_constraint = true;                        % Set to true to create the collision avoidance constraints later in the code
        end

        % Left lane blocked road constraints
        if abs(pos(1) - pos2(1))< Lsx2 && other_lane > ego_lane
            road_upperbound = roadbounds(2)+car_width/2;    % Block left lane
            no_overtake_up = true;                          % Set true to update road boundaries later
        end

        % Right lane blocked road constraints
        if abs(pos(1) - pos2(1))< Lsx2 && other_lane < ego_lane
            road_lowerbound = roadbounds(2)-car_width/2;    % Block right lane
            no_overtake_down = true;                        % Set true to update road boundaries later
        end
    end

    %% Update road boundaries when lanes are not blocked
    if no_overtake_up == false
        road_upperbound = roadbounds(3);    % Reset road upper boundary when a lane is no longer blocked
    end
    if no_overtake_down == false
        road_lowerbound = roadbounds(1);    % Reset road lower boundary when a lane is no longer blocked
    end

    %% Update state space with current constant speed
    LTI.A = [1 0 0;0 1 v_const*h;0 0 1];                                % states: vx, y, theta
```

```
    LTI.B = [h 0;0 h*v_const*lr/(lr+lf);0 h*v_const/(lr+lf)];             % inputs: a, delta
    weight.P = dare(LTI.A,LTI.B,weight.Q,weight.R);                        % DARE solution for stability

    % Update collision constraints with current constant speed
    con.A = [1 h 0 0;0 1 0 0;0 0 1 v_const*h;0 0 0 1];                     % states: xr, vxr, yr, theta
    con.B = [0 0;h 0;0 h*v_const*lr/(lr+lf);0 h*v_const/(lr+lf)];          % inputs: a, delta

    %% State boundary constraints
    upx = [max_speed road_upperbound-car_width/2]';              % State constraint upper bound
    lowx = [min_speed -(road_lowerbound+car_width/2)]';          % State constraint lower bound
    [V,W,M,N] = statecon(LTI,dim,upx,lowx);                      % Create state constraint matrices

    %% Collision constraints
    if collision_constraint == true                              % Check if there is a vehicle in the same lane in sensor range
        dim.nx = 4;                                              % Change state dimensions to be able to use the statecon function
        [V2,W2,~,~] = statecon(con,dim,upx,lowx);                % Create the collision avoidance constraints matrices
        dim.nx = 3;                                              % Change state dimensions back to the LTI system for trajectory planning

        statec.F = [Z1*W;-Z1*W;Z2*W2];                           % Create constraints with collsion avoidance
        statec.e = [M-Z1*V*x_0;N+Z1*V*x_0;-Z2*V2*x_0r-Z2*d-Lsx*ones(dim.N,1)];
    else
        statec.F = [Z1*W;-Z1*W];                                 % Create constraints without collision avoidance when there is no vehicle in sensor range in the same lane
        statec.e = [M-Z1*V*x_0;N+Z1*V*x_0];
    end

    %% Reference selection normal conditions
    if ego_lane == 0 || no_overtake_down == false           % If youre right or no one is right
        LTI.yref = [100/3.6;roadbounds(2)-1.6];             % 100km/h, right lane
    end
    if ego_lane == 1 && no_overtake_down == true            % If youre left and right is blocked
        LTI.yref = [100/3.6;roadbounds(2)+1.6];             % 100km/h, left lane
    end

```

```matlab
    % Reference selection when near destination
    if pos(1) > 400 && mod(AV{k}(6),2) == 1 && no_overtake_down == false % Go right for destination if possible
        LTI.yref = [100/3.6;roadbounds(2)-1.6];         % 100km/h, right lane
    end
    if pos(1) > 400 && mod(AV{k}(6),2) == 0 && no_overtake_up == false % Go left for destination if possible
        LTI.yref = [100/3.6;roadbounds(2)+1.6];         % 100km/h, left lane
    end

    ref(:,i) = LTI.yref;

    % OTS (Optimal Target Selection)
    [xr,ur] = OTS(LTI,dim);                              % Could also be done offline

    %% MPC
    [H,hx0,hh] = costfun(LTI,weight,dim);                         % Creare cost function
    u = sdpvar(dim.nu*dim.N,1);                                   % Define optimization variable
    Constraint = [inputc.G*u<=inputc.a, statec.F*u<=statec.e];  % Define constraints
    Objective = 0.5*u'*H*u+(hh*[x_0; xr; ur])'*u;                 % Define cost function
    optimize(Constraint,Objective);                               % Solve the problem
    u = value(u);
    u_rec = u(1:dim.nu);                                          % Select the first input only

    %% Nonlinear model
    xpos(k) = pos(1) + x(1,k)*h;                                        % Updates the x-position
    x(2,k) = x(2,k) + x(1,k)*x(3,k)*h+(lr/(lr+lf))*x(1,k)*u(2)*h;     % Updates the y-position
    x(3,k) = x(3,k) + (x(1,k)*u(2)*h)/(lr+lf);                          % Updates the vehicle angle
    x(1,k) = x(1,k) + u(1)*h;                                           % Updates the longitudinal speed

    %% Next state in SUMO
    traci.vehicle.moveToXY(AV{k},edgeID,1,xpos(k),x(2,k),(x(3,k)*(180/pi))+90,1); % This gives the next position and orientation of the AV back to SUMO, a change of coordinates is used again for the vehicle angle

    clear u     % Reset the input generated by the MPC to prevent mistakes

```

```
    end         % End of the loop to check if the AV has entered the main part of the highway
    end         % End of the loop to check if AV is on any road segement in the simulation
    end         % Loop through AV list

end
traci.close() % This stops the simulation when the simulation time T given above is over

%% Functions
%% lateralspeed.m
function latspeed = lateralspeed(i,pos,h)
    if i == 1
        latspeed = 0;
    else
        latspeed = (pos(2,i) - pos(2,i-1))/h;
    end
end

%% lateralacc.m
function latacc = lateralacc(i,latspeed,h)
    if i == 1
        latacc = 0;
    else
        latacc = (latspeed(i) - latspeed(i-1))/h;
    end
end

%% costfun.m
function [H,hx0,hh] = costfun(LTI,weight,dim)

% Weights over control horizon
Qbar = blkdiag(kron(eye(dim.N),weight.Q),weight.P);
Rbar = kron(eye(dim.N),weight.R);

% Prediction matrix x0
T = zeros(dim.nx*(dim.N+1),dim.nx);
for k = 0:dim.N
    T(k*dim.nx+1:(k+1)*dim.nx,:) = LTI.A^k;
end

% Controllability matrix
S = zeros(dim.nx*(dim.N+1),dim.nu*(dim.N));
for k = 1:dim.N
    for i = 0:k-1
        S(k*dim.nx+1:(k+1)*dim.nx,i*dim.nu+1:(i+1)*dim.nu) = LTI.A^(k-1-i)*LTI.B;
    end
end

% Cost Matrices
H = S'*Qbar*S+Rbar;
```

```
hx0 = S'*Qbar*T;

% Optimal Target Selection
hxref = -S'*Qbar*kron(ones(dim.N+1,1),eye(dim.nx));
huref = -Rbar*kron(ones(dim.N,1),eye(dim.nu));
hh = [hx0 hxref huref];
end

%% OTS.m (Optimal Target Selection)
function[xr,ur] = OTS(LTI,dim)

% Eq constraints
eq_A = [eye(dim.nx)-LTI.A -LTI.B; LTI.C zeros(dim.ny,dim.nu)];
eq_b = [zeros(dim.nx,1); LTI.yref];

% Cost function for target selection
H = blkdiag(zeros(dim.nx),0.5*eye(dim.nu));
h = zeros(dim.nx+dim.nu,1);

% Solver
options1 = optimoptions(@quadprog);
options1.OptimalityTolerance = 1e-20;
options1.ConstraintTolerance = 1.0000e-15;
options1.Display = 'off';
xur = quadprog(H,h,[],[],eq_A,eq_b,[],[],[],options1);

% Select state and input target
xr = xur(1:dim.nx);
ur = xur(dim.nx+1:end);
end

%% statec.m
function[V,W,M,N]=statecon(LTI,dim,upx,lowx)
% Prediction matrix x0
V = zeros(dim.nx*dim.N,dim.nx);
for k = 1:dim.N
    V((k-1)*dim.nx+1:k*dim.nx,:) = LTI.A^k;
end

% Controllability matrix
W = zeros(dim.nx*dim.N,dim.nu*dim.N);
for k = 1:dim.N
    for i = 0:k-1
        W((k-1)*dim.nx+1:k*dim.nx,i*dim.nu+1:(i+1)*dim.nu) = LTI.A^(k-1-i
            )*LTI.B;
    end
end

M = repmat(upx,dim.N,1);
N = repmat(lowx,dim.N,1);
end

%% AVlist.m
```

```
function[AV] = AVlist(flow,p_rate)

flow2 = round(flow*0.75*0.25);       % (A to D)
flow3 = round(flow*0.25*0.75);       % (B to C)
flow4 = round(flow*0.25^2);          % (B to D)
flow = round(flow*0.75^2);           % (A to B)

% Create a list of the AV
for i = 1:round(flow*p_rate) % AV list from flow 1
    AV{i} = sprintf('flow_0.%d',round(flow/6)+round((i-1)/p_rate));
end
j = length(AV);
for i = 1:round(flow2*p_rate) % AV list from flow 2
    AV{j+i} = sprintf('flow_1.%d',round(flow2/6)+round((i-1)/p_rate));
end
j = length(AV);
for i = 1:round(flow3*p_rate)% AV list from flow 3
    AV{j+i} = sprintf('flow_2.%d',round(flow3/6)+round((i-1)/p_rate));
end
j = length(AV);
for i = 1:round(flow4*p_rate) % AV list from flow 4
    AV{j+i} = sprintf('flow_3.%d',round(flow4/6)+round((i-1)/p_rate));
end

end

%% randAVlist.m
function[AV] = randAVlist(flow,p_rate)

flow2 = round(flow*0.75*0.25);       % (A to D)
flow3 = round(flow*0.25*0.75);       % (B to C)
flow4 = round(flow*0.25^2);          % (B to D)
flow = round(flow*0.75^2);           % (A to B)

P1 = randperm(flow,round(flow*p_rate));          % randomize AV flow 1
P2 = randperm(flow2,round(flow2*p_rate));        % randomize AV flow 2
P3 = randperm(flow3,round(flow3*p_rate));        % randomize AV flow 3
P4 = randperm(flow4,round(flow4*p_rate));        % randomize AV flow 4
for i = 1:length(P1) % AV list from flow 1
    AV{i} = sprintf('flow_0.%d',round(flow/6)+P1(i));
end
for i = 1:length(P2) % AV list from flow 2
    AV{length(P1)+i} = sprintf('flow_1.%d',round(flow2/6)+P2(i));
end
for i = 1:length(P3) % AV list from flow 3
    AV{length(P1)+length(P2)+i} = sprintf('flow_2.%d',round(flow3/6)+P3(i
        ));
end
for i = 1:length(P4) % AV list from flow 4
    AV{length(P1)+length(P2)+length(P3)+i} = sprintf('flow_3.%d',round(
        flow4/6)+P4(i));
end

```

```
end
```

A-2 Functions

A-2-1 Cost function

[44]

```
function [H,hx0,hh] = costfun(LTI,weight,dim)

% This function calculates the prediction matrix x0 and the controllability
% matrix necessary to determine all the states over a prediction horizon,
% given a list of inputs over the prediction horizon. These matrices can be
% used in the cost function, combined with the state references and input
% references from the Optimal Target Selection

% Weights over control horizon
Qbar = blkdiag(kron(eye(dim.N),weight.Q),weight.P);
Rbar = kron(eye(dim.N),weight.R);

% Prediction matrix x0
T = zeros(dim.nx*(dim.N+1),dim.nx);
for k = 0:dim.N
    T(k*dim.nx+1:(k+1)*dim.nx,:) = LTI.A^k;
end

% Controllability matrix
S = zeros(dim.nx*(dim.N+1),dim.nu*(dim.N));
for k = 1:dim.N
    for i = 0:k-1
        S(k*dim.nx+1:(k+1)*dim.nx,i*dim.nu+1:(i+1)*dim.nu) = LTI.A^(k-1-i)*LTI.B;
    end
end

% Cost Matrices
H = S'*Qbar*S+Rbar;
hx0 = S'*Qbar*T;

% Optimal Target Selection
hxref = -S'*Qbar*kron(ones(dim.N+1,1),eye(dim.nx));
huref = -Rbar*kron(ones(dim.N,1),eye(dim.nu));
hh = [hx0 hxref huref];
end
```

A-2-2 Optimal Target Selection

[44]

```
function[xr,ur] = OTS(LTI,dim)

% This function is used for the optimal target selection in the MPC.
% Because of the reference tracking with output MPC and the output
    references are
% known, an offline optimization algorithm can be run to calculate the
% state reference and input reference, which can be used in the cost
% function of the MPC

% Eq constraints
eq_A = [eye(dim.nx)-LTI.A -LTI.B; LTI.C zeros(dim.ny,dim.nu)];
eq_b = [zeros(dim.nx,1); LTI.yref];

% Cost function for target selection
H = blkdiag(zeros(dim.nx),0.5*eye(dim.nu));
h = zeros(dim.nx+dim.nu,1);

% Solver
options1 = optimoptions(@quadprog);
options1.OptimalityTolerance = 1e-20;
options1.ConstraintTolerance = 1.0000e-15;
options1.Display = 'off';
xur = quadprog(H,h,[],[],eq_A,eq_b,[],[],[],options1);

% Select state and input target
xr = xur(1:dim.nx);
ur = xur(dim.nx+1:end);
end
```

A-2-3 State constraints

[44]

```
function[V,W,M,N]=statecon(LTI,dim,upx,lowx)

% This function creates the prediction matrix x0 and the controllability
% matrix to calculate the next states for the given prediction horizon,
    given a list of inputs. These states
% can be used for the state constraints in the MPC.

% Prediction matrix x0
V = zeros(dim.nx*dim.N,dim.nx);
for k = 1:dim.N
    V((k-1)*dim.nx+1:k*dim.nx,:) = LTI.A^k;
end

% Controllability matrix
W = zeros(dim.nx*dim.N,dim.nu*dim.N);
for k = 1:dim.N
    for i = 0:k-1
        W((k-1)*dim.nx+1:k*dim.nx,i*dim.nu+1:(i+1)*dim.nu) = LTI.A^(k-1-i
            )*LTI.B;
```

```
    end
end

M = repmat(upx,dim.N,1);     % Upper bound state constraints
N = repmat(lowx,dim.N,1);    % Lower bound state constraints
end
```

A-2-4 AV list

```
function[AV] = AVlist(flow,p_rate)

% This function creates a list of all AV that has to be controlled by the
% MPC, but not chosen randomly within each of the 4 traffic flows, depending on the
% penetration rate and the total flow in the simulation

flow2 = round(flow*0.75*0.25);       % (Main highway to off-ramp)
flow3 = round(flow*0.25*0.75);       % (On-ramp to main highway)
flow4 = round(flow*0.25^2);          % (On-ramp to off-ramp)
flow = round(flow*0.75^2);           % (Main highway to main)

% Create a list of the AV
for i = 1:round(flow*p_rate) % AV list from flow 1
    AV{i} = sprintf('flow_0.%d',round(flow/6)+round((i-1)/p_rate));
end
j = length(AV);
for i = 1:round(flow2*p_rate) % AV list from flow 2
    AV{j+i} = sprintf('flow_1.%d',round(flow2/6)+round((i-1)/p_rate));
end
j = length(AV);
for i = 1:round(flow3*p_rate)% AV list from flow 3
    AV{j+i} = sprintf('flow_2.%d',round(flow3/6)+round((i-1)/p_rate));
end
j = length(AV);
for i = 1:round(flow4*p_rate) % AV list from flow 4
    AV{j+i} = sprintf('flow_3.%d',round(flow4/6)+round((i-1)/p_rate));
end

end
```

A-2-5 Random AV list

```
function[AV] = randAVlist(flow,p_rate)

% This function creates a list of all AV that has to be controlled by the
% MPC, chosen randomly within each of the 4 traffic flows, depending on the
% penetration rate and the total flow in the simulation

flow2 = round(flow*0.75*0.25);       % (Main highway to off-ramp)
flow3 = round(flow*0.25*0.75);       % (On-ramp to main highway)
```

```
flow4 = round(flow*0.25^2);          % (On-ramp to off-ramp)
flow = round(flow*0.75^2);           % (Main highway to main)

P1 = randperm(flow,round(flow*p_rate));          % randomize AV flow
P2 = randperm(flow2,round(flow2*p_rate));        % randomize AV flow 2
P3 = randperm(flow3,round(flow3*p_rate));        % randomize AV flow 3
P4 = randperm(flow4,round(flow4*p_rate));        % randomize AV flow 4

for i = 1:length(P1) % AV list from flow 1
    AV{i} = sprintf('flow_0.%d',round(flow/6)+P1(i));
end

for i = 1:length(P2) % AV list from flow 2
    AV{length(P1)+i} = sprintf('flow_1.%d',round(flow2/6)+P2(i));
end

for i = 1:length(P3) % AV list from flow 3
    AV{length(P1)+length(P2)+i} = sprintf('flow_2.%d',round(flow3/6)+P3(i
        ));
end

for i = 1:length(P4) % AV list from flow 4
    AV{length(P1)+length(P2)+length(P3)+i} = sprintf('flow_3.%d',round(
        flow4/6)+P4(i));
end

end
```

A-2-6 Lateral speed

```
function latspeed = lateralspeed(i,pos,h)

% This function calculates the lateral speed in SUMO, given the y-
    position
% of the vehicle over 2 different time steps

if i == 1
    latspeed = 0;
else
    latspeed = (pos(2,i) - pos(2,i-1))/h;
end

end
```

A-2-7 Lateral acceleration

```
function latacc = lateralacc(i,latspeed,h)

% This function is used to determine the lateral acceleration in SUMO,
% given the lateral speed over 2 different time steps

if i == 1
```

```
    latacc = 0;
else
    latacc = (latspeed(i) - latspeed(i-1))/h;
end

end
```

Bibliography

[1] Abdulhafedh A. "road traffic crash data: An overview on sources, problems, and collection methods". 2017.

[2] Acosta A. Traci4matlab. https://www.mathworks.com/matlabcentral/fileexchange/44805-traci4matlab), 2021. Accessed: 2022-11-16.

[3] K. Aliev. "internet of things applications and artificial neural networks in smart agriculture", 2018.

[4] M.H. Terra C. M. Filho and D.F. Wolf. "safe optimization of highway traffic with robust model predictive control-based cooperative adaptive cruise control", 2017.

[5] German Aerospace Center. Netedit. https://sumo.dlr.de/docs/netedit.html, 2001. Accessed: 2022-11-16.

[6] T. Ceunynck. "defining and applying surrogate safety measures and behavioural indicators through site-based observations". 2017.

[7] F.A. de Souza et al. "distributed mpc for urban traffic networks: A simulation-based performance analysis", 2014.

[8] K.N. de Winkel et al. "standards for passenger comfort in automated vehicles acceleration and jerk", 2022.

[9] F. Zannikos E. Tzirakis and S. Stournas. "impact of driving style on fuel consumption and exhaust emissions: Defensive and aggressive driving style", 2007.

[10] J. Erdmann. "lane-changing model in sumo", 2014.

[11] J. Erdmann. "sumo's lane-changing model", 2015.

[12] ET. "cut-price lidar sensor could slash cost of autonomous driving". https://eandt.theiet.org/content/articles/2020/11/cheap-as-chips-lidar-sensor-offers-cut-price-autonomous-driving/. Accessed: 2022-12-27.

[13] A. Carvalho et al. "predictive control of an autonomous ground vehicle using an iterative linearization approach". 2013.

[14] A. Laureshyn et al. "evaluation of traffic safety, based on micro-level behavioural data: Theoretical framework and first implementation". 2010.

[15] Bjärkvik E. et al. "simulation and characterisation of traffic on drive me route around gothenburg using sumo", 2017.

[16] C. Chen et al. "calibration of mitsim and idm car-following model based on ngsim trajectory datasets". 2010.

[17] C. Portilla et al. "non-linear model predictive control based on game theory for traffic control on highways", 2012.

[18] E. Bjärkvik et al. "simulation and characterisation of traffic on drive me route around gothenburg using sumo", 2017.

[19] F. Mohseni et al. "distributed model predictive control for highway maneuvers", 2017.

[20] Groot N. et al. "integrated model predictive traffic and emission control using a piecewise-affine approach", 2013.

[21] H. Guo et al. "simultaneous trajectory planning and tracking using an mpc method for cyber-physical systems: A case study of obstacle avoidance for an intelligent vehicle", 2018.

[22] J. Borek et al. "economic optimal control for minimizing fuel consumption of heavy-duty trucks in a highway environment", 2019.

[23] J. Joseph et al. "analysis of surrogate safety performance parameters for an interurban corridor", 2021.

[24] J. Kong et al. "kinematic and dynamic vehicle models for autonomous driving control design", 2015.

[25] J. Su et al. "methods for improving the accuracy of sensors with a significant influence of non-informative factors", 2016.

[26] J.Ko et al. "analysis of effects of driver/vehicle characteristics on acceleration noise using gps-equipped vehicles", 2010.

[27] J.V. Frasch et al. "an auto-generated nonlinear mpc algorithm for real-time obstacle avoidance of ground vehicles", 2013.

[28] M. Jalalmaab et al. "model predictive path planning with time-varying safety constraints for highway autonomous driving". 2015.

[29] M. Razzaghpour et al. "impact of communication loss on mpc based cooperative adaptive cruise control and platooning", 2021.

[30] M. Schuurmans et al. "safe, learning-based mpc for highway driving under lane-change uncertainty: A distributionally robust approach", 2022.

[31] P. Polack et al. "the kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles". 2017.

[32] Pourabdollah M. et al. "calibration and evaluation of car following models using real-world driving data", 2017.

[33] R. Jiang et al. "determining an improved traffic conflict indicator for highway safety estimation based on vehicle trajectory data", 2021.

[34] S. Albeaik et al. "limitations and improvements of the intelligent driver model (idm)", 2021.

[35] S. Dixit et al. "trajectory planning for autonomous high-speed overtaking using mpc with terminal set constraints". 2018.

[36] S. Dixit et al. "trajectory planning for autonomous high-speed overtaking in structured environments using robust mpc", 2019.

[37] S. Dixit et al. "trajectory planning for autonomous high-speed overtaking in structured environments using robust mpc". 2020.

[38] S. Taherian et al. "autonomous collision avoidance using mpc with lqr-based weight transformation". 2021.

[39] S.K. Zegeye et al. "nonlinear mpc for the improvement of dispersion of freeway traffic emissions", 2011.

[40] S.N. Abdulwahid et al. "modeling motorcyclists' aggressive driving behavior using computational and statistical analysis of real-time driving data to improve road safety and reduce accidents", 2022.

[41] T. Ayres et al. "preferred time-headway of highway drivers", 2001.

[42] T. Ming et al. "mpc-based trajectory tracking control for intelligent vehicles", 2016.

[43] T. Tettamanti et al. "robust control for urban road traffic networks", 2014.

[44] S. Grammatico. Ideas from the model predictive control course at the tu delft sc42125, 2019.

[45] V. Ilková and A. Ilka. "legal aspects of autonomous vehicles – an overview", 2017.

[46] ITU. "the price of lidar is falling. will driverless cars be on the road sooner?". https://www.itu.int/hub/2020/04/the-price-of-lidar-is-falling-will-driverless-cars-be-on-the-road-sooner/. Accessed: 2022-12-27.

[47] R.J.F. Rossetti J. Gonçalves. "extending sumo to support tailored driving styles", 2009.

[48] N. Jovičić J. Kocić and V. Drndarević. "sensors and sensor fusion in autonomous vehicles". 2018.

[49] Z. XU. J. SONG, Y. WU and X. LIN. "research on car-following model based on sumo", 2015.

[50] K. Suziki J. Zheng and M. Fujita. "evaluation of car-following models using trajectory data from real traffic", 2012.

[51] jckantor. "zero-order hold and interpolation". https://jckantor.github.io/CBE30338/08.01-Zero-Order-Hold-and-Interpolation.html. Accessed: 2022-11-16.

[52] I. K. Zheng L. and M. X. "freeway safety estimation using extreme value theory approaches: Acomparative study". 2014.

[53] M. Junghans L. Bieker-Walz, M. Behrisch and K. Gimm. "evaluation of car-following-models at controlled intersections", 2018.

[54] R. Lattarulo and J.P. Rastelli. "a hybrid planning approach based on mpc and parametric curves for overtaking maneuvers", 2021.

[55] P. Lin and M. Tsukada. "model predictive path-planning controller with potential function for emergency collision avoidance on highway driving", 2022.

[56] Behrendt M. "creative commons attribution-share alike 3.0 unported". https://creativecommons.org/licenses/by-sa/3.0), 2009.

[57] R. Orjuela M. Ammour and M. Basset. "collision avoidance for autonomous vehicle using mpc and time varying sigmoid safety constraints". 2021.

[58] N.H. McClamroch and I. Kolmanovsky. "performance benefits of hybrid control design for linear and nonlinear systems", 2000.

[59] M.M. Minderhoud and P.H.L. Bovy. "extended time-to-collision measures for road traffic safety assessment", 2001.

[60] A. Kurt P. Liu and U. Ozguner. "distributed model predictive control for cooperative and flexible vehicle platooning", 2019.

[61] A. Lambert R. Pepy and H. Mounier. "path planning using a dynamic vehicle model". 2006.

[62] J.B. Rawlings. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison Wisconsin, 2015.

[63] Rijkswaterstaat. "richtlijn ontwerp autosnelwegen 2019". 2022.

[64] S.G. Shelby. "delta-v as a measure of traffic conflict severity". 2011.

[65] M. Tamannaei S.M. Abtahi and H. Haghshenash. "analysis and modeling time headway distributions under heavy traffic flow conditions in the urban highways: case of isfahan", 2011.

[66] SUMO. "definition of vehicles, vehicle types, and routes". https://sumo.dlr.de/docs/Definition_of_Vehicles%2C_Vehicle_Types%2C_and_Routes.html. Accessed: 2022-11-30.

[67] SUMO. "sumo publications". https://sumo.dlr.de/docs/Publications.html. Accessed: 2022-11-30.

[68] E.M. Szumska and R. Jurecki. "the effect of aggressive driving on vehicle parameters", 2020.

[69] Tesla. "software updates". https://www.tesla.com/support/software-updates. Accessed: 2022-11-25.

[70] PLC tutorial point. "dcs plant architecture". https://www.plctutorialpoint.com/what-is-dcs-distributed-control-system/. Accessed: 2022-11-23.

[71] A. van Beinum. "turbulence in traffic at motorway ramps and its impact on traffic operations and safety". 2018.

[72] M. Verhaegen and V. Verdult. *Filtering and system identification*. Cambridge, 2007.

[73] W. XIAO. "safety evaluation of heterogeneous traffic", 2020.