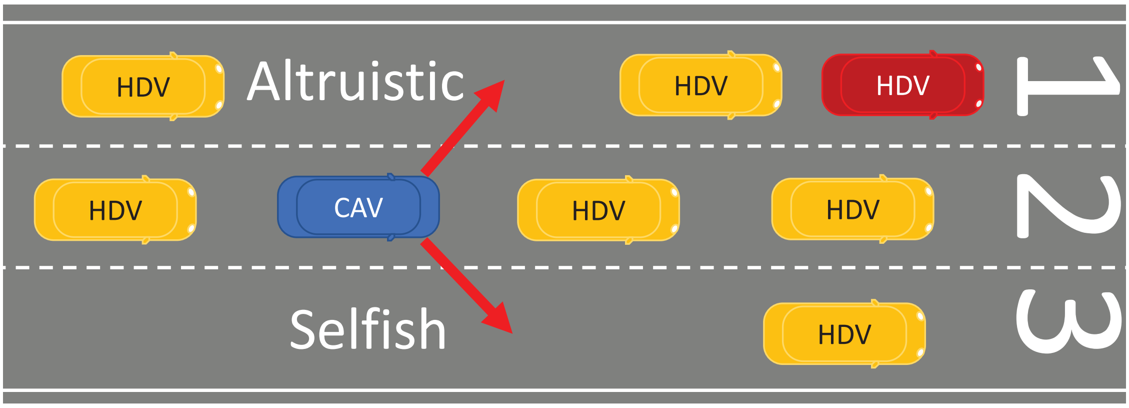
CHALMERS
UNIVERSITY OF TECHNOLOGY



Altruistic model predictive control in mixed-autonomy multi-lane traffic

Master's thesis in Systems, Control and Mechatronics

Jacob Larsson

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021

Master's thesis 2021

Altruistic model predictive control in mixed autonomy multi-lane traffic

Jacob Larsson

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Automatic Control
Chalmers University of Technology
Gothenburg, Sweden 2021

Altruistic model predictive control in mixed autonomy multi-lane traffic
Jacob Larsson

Supervisor: Furkan Keskin, Chalmers University of Technology
Examiner: Balázs Adam Kulcsár, Electrical Engineering

Master's Thesis 2021:NN
Department of Electrical Engineering
Division of Automatic Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Visualization of selfish and altruistic control strategies. The red HDVs is causing disturbances on lane 1. To mitigate this, the altruistic CAV chooses to turn left on to lane 1.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2021

Altruistic model predictive control in mixed autonomy multi-lane traffic
Jacob Larsson
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Traffic jams in transportation networks, e.g. road networks, is a challenge and concern for road safety, fuel economy, comfort and throughput . In such conditions, small disturbances in the road network due to accidents, closed lanes or random braking of preceding vehicles can cause serious propagation throughout the traffic. This is caused by the car-following dynamics of human driven vehicles (HDVs). To mitigate the effect of such events and to improve traffic flow, connected automated vehicles (CAV) are used. The idea is that CAVs have the capabilities of smoothing traffic on the highway, thus reducing jamming. Current control strategies for CAVs increase comfort, efficiency and safety of the CAV through using a MPC framework. However, the current approaches disregards surrounding HDVs driving objectives in order to benefit the selfish CAV. They also do not consider overall traffic situation and possibilities of traffic jamming based on control decisions. This is so called selfish driving. A new approach to controlling the CAV's is through altruistic control. Here, the idea is to incorporate information from surrounding vehicles in order to optimize the overall traffic situation for the benefits of other vehicles on the road, effectively sacrificing ones-self for the benefit of others. The work presents methods of how this type of control can be formulated and what the benefits of altruistic control are. The methods are validated in a high fidelity traffic simulator using single lane and multi lane scenarios with varying degree of traffic conditions.

Keywords: Altruism, MPC, traffic, simulation, autonomous, vehicle.

Acknowledgements

Special thanks goes the my supervison Furkan Keskin and my examiner Balázs Kulcsár for their supprt throughout the research process. I have been fortunate to be able to work closely with ongoing research together with them and that we were able to submit our results in a special issue T-ITS journal on "Deployment of Connected and Automated Vehicles in Mixed Traffic Environment and the Implications on Traffic Safety and Efficiency". Without Furkan and Balázs's support, this work would not be where it is today. Many thanks for entrusting me with this work and for providing me a great foundation for future work and research.

Jacob Larsson, Gothenburg, June 2021

Contents

List of Figures

1
Introduction

This thesis aims to explore the area of model predictive control for autonomous vehicles in transportation networks. This chapter provides some background information regarding the reasons why the problem is interesting, the overall aim of the project as well as the research questions and limitations.

1.0.1 Background

Traffic jams in transportation networks, e.g. road networks, is a challenge and concern for road safety, fuel economy, comfort and throughput [1]. In such conditions, small disturbances in the road network due to accidents, closed lanes or random braking of preceding vehicles can cause serious propagation throughout the proceeding traffic. This is caused by the car-following dynamics of human driven vehicles (HDV). To mitigate the effect of such events and to improve traffic flow, connected automated vehicle (CAV) flow control has become a popular topic [2, 3]. The idea is that CAVs have the capabilities of smoothing traffic on the highway, thus reducing jamming. Current control strategies for CAVs increase comfort, efficiency and safety of the CAV through using a MPC framework. However, the current approaches disregards surrounding vehicles driving objectives in order to maximize the driving goals of the selfish CAV. They also do not consider overall traffic situation and possibilities of traffic jamming based on control decisions. This is so called *selfish driving*. A new approach to controlling the CAV's is through altruistic control [4], which is an active area of research at the automatic control group. Here, the idea is to incorporate information from surrounding vehicles and other CAVs in order to optimize the overall traffic situation.

1.0.2 Aim

The aim of the thesis is to explore the design of altruistic driving strategies for CAVs and how altruistic control of CAVs can affect the traffic smoothness on mixed-autonomy, multi-lane roads. More specifically, the thesis work will implement an altruistic MPC based controller design and test the performance on mixed-autonomy multi-lane roads in real life scenarios using the PTV Vissim simulation environment [5]. The simulations concern individual altruistic driving and investigates its impact on the attenuation of jamming. Altruism will also be compared to selfish driving for the CAV's using low and high fidelity traffic simulations. The overall goal of the thesis is to produce and publish a journal paper. The journal paper can be seen in Appendix A.

1.0.3 Research questions

Research questions are important in order to guide the development work throughout the thesis and are mainly used to design and evaluate experiments and their results. For the purpose of improving traffic using altruistic behaviour, some main areas of interest are considered:

- Can altruistic driving reduce impact of congestion compared to other driving modes?
- What are the traffic improvements that can be made?
- In what traffic situations can/cannot altruistic driving be beneficial?
- Can the controller handle disturbances by vehicles?
- Under what conditions does altruism work/not work?

1.0.4 Limitations

The aim of the project is to implement and validate altruistic MPC control on multi-lane roads. The work will thus be limited to multi-lane highways without intersections. This means that other driving scenarios, such as urban driving, will not be included in the work. Furthermore, CAV penetration rate and multi-CAV optimization problems will be left for foture work on this topic.

2

Methodology

This chapter covers the methods that are used in order to investigate the main hypothesis presented previously. Here, the MPC problem formulation is covered to provide an in-depth view of the equations. The simulation software is then presented, along with scenario designs, ways of programming the MPC problem and the simulation. The optimization programming toolbox is also briefly covered.

2.1 Altruistic vs. selfish control strategy

An altruistic control strategy can, in the context of a CAV, be described as sacrificing ones own driving objectives for the benefit of the HDVs driving objectives. The objectives can be anything from keeping a steady pace to lowering the emissions [4].

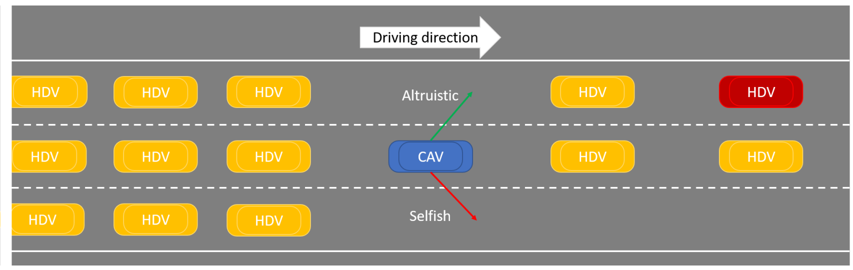


Figure 2.1: Altruistic vs. selfish action: Red HDV causes congestion on the lane

Figure 2.1 depicts a hypothetical scenario where the CAV has two options: A) go in the left lane in order to reduce traffic oscillations caused by the leading HDV or B) change to the right lane and keep a steady pace. It is also assumed that the driving objective of the CAV is to increase the comfort of vehicles. Under these circumstances, an altruistic controller would sacrifice its own comfort for the benefit of the succeeding HDVs by choosing option A. If the CAV were to act in a selfish manner, the best option would be B since it would not impact the CAVs own acceleration.

2.2 Model Predictive Control

Model predictive control (MPC) schemes are set-up as quadratic optimization problems where an objective function is minimized, when subjected to constraints, over a future time horizon [6]. By using a dynamic model and problem specific constraints, MPC is able to predict the systems trajectories in to future time and optimize them using control input variables. The optimal control input is then obtained, and the first input is used to control the actual system.

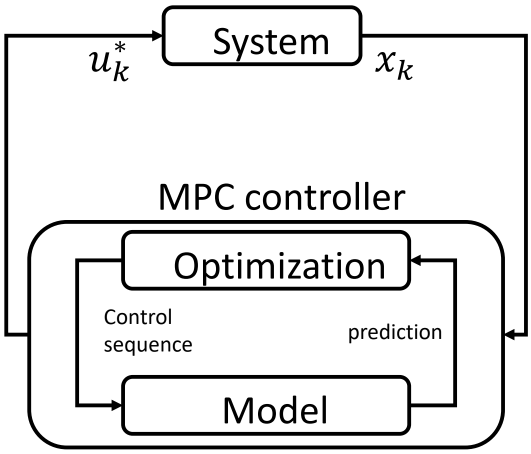


Figure 2.2: A basic diagram of system and controller interaction

Figure 2.2 shows a simple diagram of how the controller interacts with the system. The system output x_k is the system states and the MPC controller uses these to calculate an optimal control sequence by predicting the systems future trajectories over a time horizon N using a model of the system. When the optimization is sufficiently satisfied, the optimal control input u_k^* is applied to the system, and the cycle repeats. Since the implementation of MPC is most often made on a microcontroller, only discrete time k is considered for the controller. Thus, a MPC controller suitable for controlling a CAV on a highway consists of three main components; a discrete time LTI system $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k$, an objective function J and an optimization algorithm to minimize J over the time horizon using the control input sequence $\mathbf{u}$.

The problem of MPC can be formulated based on the knowledge the discrete dynamical system along with the structure of quadratic optimization. As previously described, MPC optimizes the control input u over a horizon of length N in order to minimize some objective function J. The general MPC optimization problem can be described as;

$$\begin{aligned} \underset{u}{\mathit{minimize}} \; & J(\mathbf{x}, \mathbf{u}) \\ \mathit{subject\ to} \; \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \\ \mathbf{x}_{min} \leq \mathbf{x}_k &\leq \mathbf{x}_{max} \\ \mathbf{u}_{min} \leq \mathbf{u}_k &\leq \mathbf{u}_{max} \end{aligned} \tag{2.1}$$

From (2.1), the MPC task is divided into three main categories: The dynamical system model, the objective function and additional design constraints. Since MPC can be very notation heavy, and that there are many vehicles that are taken into account in the optimization problem, some clarification of notations will be made throughout.

2.2.1 Dynamical system model

Control systems are most often implemented on a microcontroller. Since a computer is a digital device, continuous time is difficult to work with. The MPC controller is instead implemented in a discrete time fashion and thus the dynamical system model must take this in to account. Therefore, a discrete time dynamical system is used in the controller implementation.
A discrtete time LTI system [7] can be represented in a state space formulation as

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \\ where\ k &= 1, ..., N \end{aligned} \tag{2.2}$$

For the purposes of MPC, one would like to make predictions along a horizon of fixed length. This is accomplished by utilizing the state model in 2.2 over a horizon of length N. To use this representation in an optimization friendly format, one can see that step $k = 2$ will depend on step $k = 1$ and so forth, i.e.

$$\begin{aligned} & x_1 = Ax_0 + Bu_0 \\ & x_2 = Ax_1 + Bu_1 = A(Ax_0 + Bu_0) + Bu_1 \\ & = (A^2x_0 + ABu_0) + Bu_1 \\ & \vdots \\ & x_N = Ax_{N-1} + Bu_{N-1} = A^{N-1}x_0 + \begin{bmatrix} A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix} \begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix} \end{aligned} \tag{2.3}$$

From 2.3, it is clear that future states depend only on the initial state, succeeding control inputs and the dynamical system matrices from 2.2. Furthermore, 2.3 can be represented with matrix formulation;

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N-1} \end{bmatrix} x_0 + \begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix} \begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix} \tag{2.4}$$

In compact form, one can formulate 2.4 as;

$$X = A_b x_0 + B_b U \tag{2.5}$$

The general formulation can then be applied to the specific scenario of a CAV controlling HDVs in traffic. First, the system states are defined. The CAV system model contains three states:

$$\begin{aligned} \mathbf{x}_k^{CAV} &= \begin{bmatrix} p_k^{CAV} & v_k^{CAV} & y_k^{CAV} \end{bmatrix}^T \\ where\ p_k^{CAV} \in \mathbb{R}, v_k^{CAV} &\in \mathbb{R}, y_k^{CAV} \in [1, 2, ..., N_{lanes}] \end{aligned} \tag{2.6}$$

The CAV states at discrete time instant k in (2.6) are longitudinal position $p[m]$, longitudinal velocity $v\left[\frac{m}{s}\right]$ and lane number y. On a road, the lane numbering starts at the rightmost lane and increases with each lane until the leftmost lane. There are also two inputs to the system:

$$\begin{aligned} \mathbf{u}_k^{CAV} &= \begin{bmatrix} a_k^{CAV} & \delta_k^{CAV} \end{bmatrix}^T \\ where\ a_k^{CAV} &\in \mathbb{R}, \delta_k^{CAV} \in [-1, 0, 1] \end{aligned} \tag{2.7}$$

The inputs in (2.7) are longitudinal acceleration $a\left[\frac{m}{s^2}\right]$ and lane change decision δ. The lane change decision will either keep the vehicle in the current lane: $\delta_k^{CAV} = 0$, change to the left lane: $\delta_k^{CAV} = 1$ or change to the right lane: $\delta_k^{CAV} = -1$. This is a simplified lane changing model in order to produce a tractable optimization problem. By means of these inputs, the CAV can control its own movements. The CAV will also indirectly control the succeeding HDVs and thus influencing the traffic behind itself.
Similarly, the states and inputs are for the jth HDV at discrete time instant k are defined:

$$\begin{aligned} \mathbf{x}_{j,k}^{HDV} &= \begin{bmatrix} p_{j,k}^{HDV} & v_{j,k}^{HDV} & y_{j,k}^{HDV} \end{bmatrix}^T \ for\ j = 1...N_{HDV} \\ where\ p_{j,k}^{HDV} &\in \mathbb{R}, v_{j,k}^{HDV} \in \mathbb{R}, y_{j,k}^{HDV} \in [1, 2, ..., N_{lanes}] \\ \mathbf{u}_{j,k}^{HDV} &= \begin{bmatrix} a_{j,k}^{HDV} & \delta_{j,k}^{HDV} \end{bmatrix}^T \ for\ j = 1...N_{HDV} \\ where\ a_{j,k}^{HDV} &\in \mathbb{R}, \delta_{j,k}^{HDV} \in [-1, 0, 1] \end{aligned} \tag{2.8}$$

The discrete time dynamical model intends to update the states in a linear fashion. As the focus of the thesis is to control the CAV in order to influence traffic behind it, the system model is kept simple. The model disregards any vehicle dynamics related equations and simply acts as a point moving through space by updating the aforementioned states based on the inputs. Thus, the system can be represented as a linear state space model:

$$\begin{aligned} \mathbf{x}_{k+1}^{CAV} &= \mathbf{A}\mathbf{x}_k^{CAV} + \mathbf{B}\mathbf{u}_k^{CAV} \\ \mathbf{x}_{j,k+1}^{HDV} &= \mathbf{A}\mathbf{x}_{j,k}^{HDV} + \mathbf{B}\mathbf{u}_{j,k}^{HDV} \\ where\ \mathbf{A} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3\times3}, \mathbf{B} &= \begin{bmatrix} \frac{\Delta t^2}{2} & 0 \\ \Delta t & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{3\times2} \end{aligned} \tag{2.9}$$

As mentioned previously, MPC is based on future time predictions based on the dynamic model used. As such, MPC considers a time horizon of length N. At each time instant k, the controller will optimize over a horizon length of N steps. As such, the states and inputs used as optimization variables are:

$$\begin{aligned} &\mathbf{x}_{k+n}^{CAV}, \mathbf{x}_{j,k+n}^{HDV} \\ &\mathbf{u}_{k+n}^{CAV}, \mathbf{u}_{j,k+n}^{HDV} \\ where\ n &= [1, ..., N] \end{aligned} \tag{2.10}$$

The current states in (2.8) and (2.6) at time k are used as initial conditions for the optimization. The states and inputs in (2.43) are used as optimization variables, where specifically the controller input a_{k+n}^{CAV} will be minimized. The state trajectory for the CAV in the optimization, according to (2.3), is then:

$$\begin{aligned}
\mathbf{x}_{k+1}^{CAV} &= \mathbf{A}\mathbf{x}_{k}^{CAV} + \mathbf{B}\mathbf{u}_{k}^{CAV} \\
\mathbf{x}_{k+2}^{CAV} &= \mathbf{A}\mathbf{x}_{k+1}^{CAV} + \mathbf{B}\mathbf{u}_{k+1}^{CAV} = \mathbf{A}^2\mathbf{x}_{k}^{CAV} + \mathbf{B}\mathbf{u}_{k}^{CAV} + \mathbf{B}\mathbf{u}_{k+1}^{CAV} \\
&\vdots \\
\mathbf{x}_{k+N}^{CAV} &= \mathbf{A}\mathbf{x}_{k+N-1}^{CAV} + \mathbf{B}u_{k+N-1}^{CAV} = \mathbf{A}^{N-1}\mathbf{x}_{k}^{CAV} + \mathbf{A}^{N-1}\mathbf{B}\mathbf{u}_{k}^{CAV} + ... + \mathbf{B}\mathbf{u}_{k+N}^{CAV}
\end{aligned} \tag{2.11}$$

In a similar manner as (2.11) the HDV trajectories are formulated.

2.2.2 Driver model

Car following models are used to describe the behaviour human-driven vehicles in a traffic situation [8]. They take information from the surroundings and try to mimic real driving behaviour. There are several different models available, simple acceleration setting based on own vehicle velocity and preceding vehicle velocity to psycho-physical models that take psychological phenomena in to account.
The goal of the MPC is to calculate an optimal input acceleration for the CAV, such that the HDVs are indirectly controlled. The CAV acceleration is thus a variable which will be minimized in the optimization problem. For the HDVs, their acceleration is determined by the driver model used. In simulation software, a complex driver model is often used in order to create a realistic scenario. The philosophy behind a MPC controller is to use simpler representations of reality in order to predict the behaviour of a more complex system.
Optimal velocity relative velocity (OVRV) [4] is a driver model which weighs the optimal velocity of own vehicle verses the relative velocity between the own vehicle and the preceding vehicle. OVRV is a piece-wise linear function which determines the own vehicles acceleration at each time instant;

$$a(t) = f(h(t), v(t), \Delta v(t)) = \alpha(V(h(t)) - v(t)) + \beta\Delta v(t) \tag{2.12}$$

In 2.12, the own vehicles longitudinal acceleration $a(t)$ is determined by a function that depends on the distance between the own vehicle and the preceding vehicle, i.e. the headway $h(t)$, the own vehicles velocity and the relative velocity between the two, $\Delta v(t)$. $V(h(t))$ is a piece-wise linear function that maps a headway to a desired velocity;

$$\begin{aligned}
V(h(t)) = \left[\bar{V}(h(t))\right]_0^{v_{max}} &= max(0, min(v_{max}, \bar{V}(h(t))) \\
where\ \bar{V}(h(t)) &= v_{max}\frac{h(t) - h_{min}}{h_{max} - h_{min}}
\end{aligned} \tag{2.13}$$

By the $max()$ operator in 2.13 it is clear that the OVRV model is piece-wise linear. In 2.13, there are tuning parameters that determine the behaviour of $\bar{V}(h(t))$. v_{max} is the maximum allowed velocity of the vehicles, h_{max} and h_{min} are the maximum and minimum allowed headway between the two vehicles.

In 2.12, the parameters α and β weigh the effect of headway based on desired velocity verses the relative velocity on the longitudinal acceleration.
Due to OVRV being an intractable and piece wise linear function, it cannot be directly used as a linear constraint in the optimization problem.
In equation 2.12, there is a function that maps headway to velocity in a piece wise linear fashion;

$$\bar{V}(h(t)) = v_{max}\frac{h(t) - h_{min}}{h_{max} - h_{min}} \tag{2.14}$$

Plotting this function reveals it's shape and gives an idea how to reformulate in to linear equations suitable for use in a convex optimization problem.

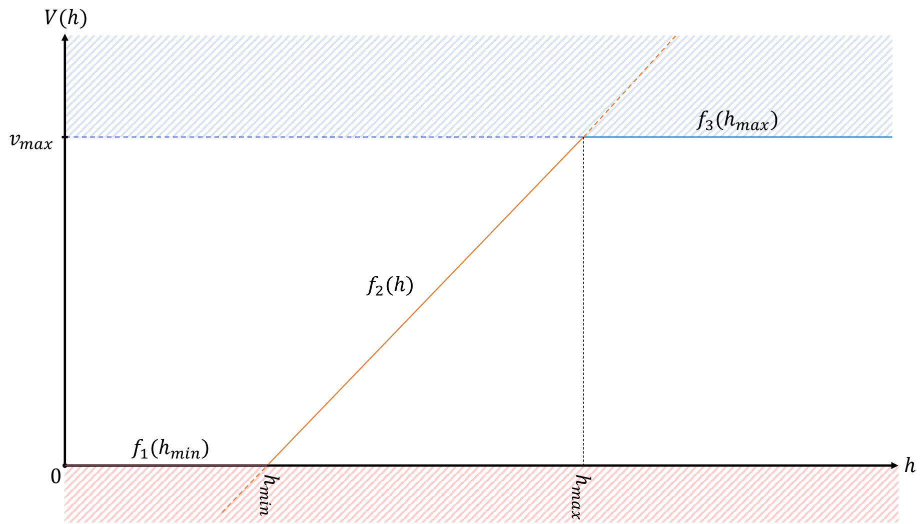


Figure 2.3: Illustration of 2.14

The function 2.14 works by taking the $min(...)$ and $max(...)$ operators in order to clip $f_2(h)$ to lie between 0 and v_{max}. This can also be accomplished by constraining $f_2(h)$ using linear constraints $f_1(h_{min})$ and $f_3(h_{max})$ in conjunction with a slack variable in $f_2(h)$. In 2.3, the blue shaded area is constraint $f_3(h_{max})$, which bounds from above. The red shaded area is constraint $f_1(h_{min})$, which bounds from below. The constraint equations for the reformulated OVRV model is then;

$$\begin{aligned} a_{j,k}^{HDV} &= \Big(\alpha(\bar{V}(h_{j,k}^{HDV}) - v_{j,k}^{HDV}\Big) - \beta\Delta v_{j,k}^{HDV} + \gamma_{j,k}^{HDV} \\ a_{j,k}^{HDV} &\geq \Big(\alpha(\bar{V}(h_{max}) - v_{j,k}^{HDV}\Big) - \beta\Delta v_{j,k}^{HDV} \\ a_{j,k}^{HDV} &\leq \Big(\alpha(\bar{V}(h_{min}) - v_{j,k}^{HDV}\Big) - \beta\Delta v_{j,k}^{HDV} \end{aligned} \tag{2.15}$$

As can be seen, the equality constraint in 2.15 has an added slack variable. In order to keep the OVRV model correct, the slack variable has to be as small as possible. This can be accomplished by minimizing the variable by adding a quadratic function as an objective function. With this added, the objective function becomes;

$$J_{slack} = \lambda \sum_{j \in \mathcal{G}_{fHDV}} \left|\left|\gamma_{j,k}^{HDV}\right|\right|_2^2 \tag{2.16}$$

In 2.16, the slack variable is a quadratic function using the squared L2-norm. The set $\mathcal{G}_{fHDV}$ contains all HDVs following the leading HDV. The weight λ controls how much the slack variable will influence the overall objective function and thus controls how much the slack variable γ is minimized. Increasing λ will cause γ to decrease. It is worth noting that the reformulation is not exact, and will vary from the true OVRV model, however minimizing the slack variable and using λ will ensure that the gap is small and that the model mismatch is low.
By the addition of slack variables for all HDVs, γ must be added to the MPC optimization variables as well.
As mentioned, the OVRV model is used for all HDVs in order to set their accelerations at each simulation step. However, since the OVRV model relies on information regarding a preceding vehicle, this model will not work for the leading HDV. The leading HDV will instead have to be accounted for by keeping the acceleration constant throughout the entire prediction horizon. This is assumed due to the leading HDV being treated as a black box, where the controller has no information regarding what is ahead of the leading vehicle. Thus, a safe way of predicting the leading vehicles movements would be to keep the acceleration constant. The velocity will also remain constant, however the velocity will never become negative. As such, every optimization will use a pre-calculated acceleration and velocity profile for the leading HDV.

2.2.3 Driving objective

The objective function is the function which the optimization solver tries to minimize. For a self driving vehicle, which should act in an altruistic manner with regards to the surrounding traffic, there are a few objectives that are of interest in this work: Traffic efficiency and driving comfort.

2.2.3.1 Traffic efficiency

Traffic efficiency can be described as how well the vehicle is keeping up with the speed limit on the road. When traffic congestion increases, the vehicle speed will decrease and thus the traffic efficiency is reduced. To describe the objective in a way that can be used in a quadratic minimization problem, one may consider to use a squared L2-norm between the desired velocity and the actual velocity of the vehicle.

$$J_{efficiency} = ||v_{actual} - v_{desired}||_2^2$$

In order to introduce altruism, a hyperparameter κ is used to weigh the importance between the efficiency of the CAV and the HDVs. Furthermore, since the MPC optimizes over the entire horizon length N, the sum of $J_{efficiency}$ is used. The full traffic efficiency objective is then;

$$J_{efficiency} = ||\mathbf{v}^{CAV} - v_{desired}||_2^2 + \kappa \sum_{j \in \mathcal{G}_i^{sHDV}} ||\mathbf{v}_j^{HDV} - v_{desired}||_2^2 \tag{2.17}$$

When minimizing the objective in 2.17, the MPC tries to reduce velocity difference between the CAV, HDVs behind the CAV and the desired velocity. By setting $\kappa \geq 0$ it is possible to influence the altruistic behaviour of the controller. Increasing leads to increasing altruism and thus sacrificing the velocity difference for the CAV in favour of the HDVs.

2.2.3.2 Comfort

Comfort is an important aspect when it comes to stop-and-go traffic, or when a lane is heavily congested. The objective to achieve comfort for passengers of both CAV and HDV is to minimize both the magnitude and the rate of change of the longitudinal acceleration. As such, the comfort objective is twofold; minimize acceleration magnitude and minimize the derivative of acceleration, i.e. jerk. Similarly to the traffic efficiency, acceleration magnitude and jerk objectives can be formulated as minimizing the squared L2-norm;

$$J_{acc.\ mag.} = ||a_{actual}||_2^2 \tag{2.18}$$

$$J_{jerk} = ||\frac{d}{dt}a_{actual}||_2^2 \tag{2.19}$$

The acceleration magnitude objective in 2.18 is also implemented in an altruistic manner using κ and takes in to account the CAV and the HDVs behind the CAV.

$$J_{acc.\ mag.} = ||\mathbf{a}^{CAV}||_2^2 + \kappa \sum_{j \in \mathcal{G}_i^{sHDV}} ||\mathbf{v}_j^{HDV}||_2^2 \tag{2.20}$$

Similarly, 2.19 is implemented. However, since the MPC controller will be using discrete time, the derivative is replaced by a first-order Euler method approximation by;

$$\frac{d}{dt}a(t) \approx \frac{a(t+\Delta t) - a(t)}{\Delta t} \tag{2.21}$$

Using 2.21, the jerk objective is implemented as;

$$J_{jerk} = \left|\left|\frac{\mathbf{a}^{CAV,pre} - \mathbf{a}^{CAV}}{\Delta t}\right|\right|_2^2 + \kappa \sum_{j \in \mathcal{G}_i^{sHDV}} \left|\left|\frac{\mathbf{a}_j^{HDV,pre} - \mathbf{a}_j^{HDV}}{\Delta t}\right|\right|_2^2 \tag{2.22}$$

The total comfort objective will then be a the sum of 2.20 and 2.22. The influence of each objective is weighted against each other using the hyperparameter w_2;

$$J_{comfort} = J_{acc.\ mag.} + w_2 J_{jerk} \tag{2.23}$$

2.2.3.3 Total objective function

The traffic efficiency and the comfort objectives can be combined to form the final MPC objective using 2.17 and 2.23. In order to tune the objective for either efficiency or comfort, the hyperparameter w_1 is used. Thus, the total objective function is;

$$J_{MPC\ objective} = J_{efficiency} + w_1 J_{comfort} + J_{slack} \tag{2.24}$$

2.2.4 Constraints

In the MPC formulation, constraints can be used to ensure that the controller adheres to certain physical bounds. As described previously, the system model is used as a constraint in order to ensure that the MPC controller will adhere to the physical bounds of the vehicles. By similar reasoning, the OVRV model is also used in order to constrain the HDVs motion so that the controller will follow the model. Other physical constraints, such as constraints on input variables and other traffic related properties are also used in order to create a desired behaviour of the controller.
One such behaviour is time headway. Time headway is a metric where the headway distance is considered a function of velocity and time. Simply, multiplication of velocity and time gives a distance, and this distance is considered a time headway. This metric captures the behaviour that when the velocity is high, the distance must be greater so that there is enough time to react to sudden events ahead of the vehicle. Thus, it is dynamic and significantly contributes to driving behaviour. As mentioned previously, the OVRV is a simple model that only takes in to account the current headway with regards to distance to the preceding vehicle. As mentioned, in the Wiedemann model, there is a time headway component. Since the MPC controller will be used in a simulator using the more advanced Wiedemann model, the controller can be upgraded with a time headway using constraints. This will help the controller to better predict the true driving behaviour of the vehicles;

$$p_{j,k}^{Preceding\ HDV} - p_{j,k}^{HDV} \geq h_{min} + t_{min}^{HDV} v_{j,k}^{HDV} \tag{2.25}$$

In 2.25, the controller accounts for the HDV headway and ensures that it is at least great than the minimum headway distance and a time headway controlled by t_{min}, e.g. 3 seconds. The MPC controller should also ensure that the CAV keeps a similar time headway;

$$p_{k}^{Preceding\ CAV} - p_{j,k}^{CAV} \geq h_{min} + t_{min}^{CAV} v_{k}^{CAV} \tag{2.26}$$

Additionally, to ensure that the CAV keeps a safe headway to vehicles in other lanes during a lane change maneuver two additional constraint are added;

$$\begin{aligned} p_{k}^{Preceding\ CAV} - p_{j,k}^{CAV} &\geq h_{safe} \\ p_{j,k}^{CAV} - p_{k}^{Succeeding\ CAV} &\geq h_{safe} \end{aligned} \tag{2.27}$$

Both 2.26 and 2.27 ensure that the CAV is controlled in a safe manner, leaving enough space in front of the CAV so that the controller will be able to brake in time. 2.27 also ensures that the succeeding vehicle in a new lane will not have to perform any emergency braking maneuvers when the CAV switches lane.
In addition to safety constraints for CAVs and HDVs, there are also constraints on CAV control inputs:

$$\begin{aligned} a_{k+n}^{CAV} &\leq a_{max} \\ a_{k+n}^{CAV} &\geq a_{min} \end{aligned} \tag{2.28}$$

The constraints in (2.29) will ensure realistic acceleration and braking behaviour of the CAV. Lastly, the CAV velocity must be bounded, as negative velocities are not desirable in the circumstances of driving on a highway:

$$v_{k+n}^{CAV} \geq 0 \tag{2.29}$$

2.3 Quadratic optimization

MPC optimization formulations are convex quadratic optimization problems.

2.3.1 Integer relaxation

The MPC formulation described in the previous section is a mixed-integer optimization problem with quadratic objective function and linear constraints. This is because the lane state in (2.6) is an integer variable due to discrete lane numbers $[1, 2, 3, \dots]$ and this can be hard to work with in practice. Generally, integer optimization problems are NP-hard, and most often an integer relaxation is made in order to solve the optimization problem. A common method for relaxing integer problems is to use floating point numbers in the optimization and then rounding to the nearest integer. However, for the purposes of this MPC optimization problem, it is possible perform integer relaxation by exploiting the way the integer numbers are used and the structure of the problem. As mentioned, the integer numbers are strictly used for lane numbers and lane changing decisions. If the lane changing control decision is offloaded from the optimization and instead is handled externally, it is possible to omit the integer states from the optimization. This may be achieved by performing 3 concurrent MPC optimizations for the lanes that may be reached in 1 step. For the HDVs, it must then be assumed that they do not change lane as the optimization will only see a prediction over a single lane. Effectively, the CAV optimizes three single lane scenarios even though the actual scenario is multi lane. The lane change decision will then be to choose the lane with the best predicted outcome:

$$\delta_k^{CAV} = \begin{cases} 1, & \text{if } J_k^{*,left} < min\left(J_k^{*,current}, J_k^{*,right}\right) \\ -1, & \text{if } J_k^{*,right} < min\left(J_k^{*,current}, J_k^{*,left}\right) \\ 0, & \text{otherwise} \end{cases} \tag{2.30}$$

In (2.30), the lane change decision δ_k^{CAV} is a conditional expression based on the optimal function values of the current lane, the lane immediately left and right of the CAV, $J_k^{*,current}$, $J_k^{*,left}$ and $J_k^{*,right}$ respectively. The lane decision is based on the best predicted outcome, i.e. the lane with the lowest predicted cost. A lane change to the left is, as before, represented by $\delta_k^{CAV} = 1$ in the lane numbering, a right lane change is $\delta_k^{CAV} = -1$ and keeping in the same lane is $\delta_k^{CAV} = 0$. The altered system states and inputs are then:

$$\begin{aligned} \mathbf{x}_k^{CAV} = \begin{bmatrix} p_k^{CAV} & v_k^{CAV} \end{bmatrix}^T, \ \mathbf{x}_{j,k}^{HDV} = \begin{bmatrix} p_{j,k}^{HDV} & v_{j,k}^{HDV} \end{bmatrix}^T \\ \mathbf{u}_k^{CAV} = \begin{bmatrix} a_k^{CAV} \end{bmatrix}^T, \ \mathbf{u}_{j,k}^{HDV} = \begin{bmatrix} a_{j,k}^{HDV} \end{bmatrix}^T \end{aligned} \tag{2.31}$$

In addition, the lane states and inputs in the system model will be removed and thus simpler state and input matrix are used:

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{2\times 2}, \mathbf{B} = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \\ 0 \end{bmatrix} \in \mathbb{R}^{3\times 1} \tag{2.32}$$

Since the alteration is just removing the lane states and inputs from the system, no other alterations are made to the system model equations in (2.2))

2.3.2 Formulating general quadratic programs

A quadratic program (QP) is a way of solving convex optimization problems on the form of;

$$\begin{aligned} \underset{x}{min} f_0(x) \\ s.t. f_i(x) \leq b_i, i = 1, ..., n \end{aligned} \tag{2.33}$$

In 2.33, the objective function $f_0(x)$ can be any quadratic function. The function is subject to constraints $f_i(x) \leq b_i$, where $f_i(x)$ is a linear function. The goal is to find a solution for the variables x such that the constraints are satisfied and the quadratic equation is minimized. In general there are no analytical solutions to convex optimization problems and thus iterative methods are used to solve such problems. A common tool for solving 2.33 is interior-point methods. Interior-point methods are used in most commercial optimization solvers.
A more common way of expressing 2.33 is to use matrix notation;

$$\begin{aligned} \underset{x}{min} \frac{1}{2} x^T P x + q^T x \\ s.t. \; Gx \leq H \\ Ax = b \end{aligned} \tag{2.34}$$

The objective function in 2.34 is simply a quadratic function on matrix form with the constant term being omitted, and is subject to both inequality constraints, $Gx \leq H$, and equality constraints, $Ax = b$. Since an equality constraint can be represented by two inequality constraints with different sign, 2.33 and 2.34 are equivalent.

2.3.3 Formulating the MPC optimization problem

While the general formulation for QPs in (2.34) is the final form of the MPC optimization problem, there are still practical considerations for creating "good" numerical optimization problems. Some reformulation of the OVRV model has already been made, and that allowed the model to be used as a constraint. However, when the optimizer solves the problem, there are always numerical issues to take in to account, e.g. there is no infinite precision for the numerical values of variables. One of these particular issues is scaling of objective and constraints. For example:

$$J(\mathbf{x}) = x_1^2 + 10^{10} x_2^2 \tag{2.35}$$

The variables x_1 and x_2 in (2.35) will hase significatly different contribution to the overall objective function value. When dealing with numerical optimization, the differences in scale of 10^{10} may cause the variable x_1 to simple be ignored. This is because by having such a different scale for x_2, this term becomes dominating and thus one may not be sure if the numerical optimum for x_1 is the near the true optimum. Simply, the contribution of x_1 may be unreliable. If x_1 is the control input for the CAV, the control input may be unusable and can have devastating effects on the system. In order to address this type of issue, the objective function and constraints are scaled down. The total objective function, including the addition of the slack variable, is:

$$J_{total} = J_{efficiency} + w_1\left[J_{acc.\ mag.} + w_2 J_{jerk}\right] + \lambda J_{slack} \tag{2.36}$$

For the total objective function (2.36), there will be scaling issues related both to the hyperparameters w_1, w_2, λ and the internal altruism weight κ present in $J_{efficiency}$, $J_{acc.\ mag.}$ and $w_2 J_{jerk}$. Furthermore, there may also be large differences in scale between the magnitude of velocities, accelerations and slack variables. Scaling problems resulting from differences in weights may be resolved by capping all weights to lie within a range $[0, 1]$ and rewriting the objective function accordingly. Issues related to the magnitude of variables can be solved by dividing by the maximum value the variable can reach. The final, re-scaled, objective function is then:

$$\begin{aligned} J_{total} = (1-\lambda)\left[(1-w_1)J_{efficiency} + w_1\left[(1-w_2)J_{acc.\ mag.} + \dots\right.\right. \\ \left.\left. w_2 J_{jerk}\right]\right] + \lambda J_{slack} \\ \textit{where } w_1 \in [0,1]\ , w_2 \in [0,1]\ , \lambda \in [0,1] \end{aligned} \tag{2.37}$$

The individual objective functions are then also modified to accommodate the altered altruism hyperparameter κ and scaling down the variables:

$$\begin{aligned} J_{efficiency} = (1-\kappa)\frac{1}{v_{max}^2}||\mathbf{v}^{CAV} - \mathbf{v}_{desired}||_2^2 + \dots \\ \kappa\frac{1}{N_{sHDV} v_{max}^2}\sum_{j\in\mathcal{G}^{sHDV}} ||\mathbf{v}_j^{HDV} - \mathbf{v}_{desired}||_2^2 \\ \\ J_{acc.\ mag.} = (1-\kappa)\frac{1}{a_{max}^2}\left|\left|\mathbf{a}^{CAV}\right|\right|_2^2 + \kappa\frac{1}{N_{sHDV} a_{max}^2}\sum_{j\in\mathcal{G}^{sHDV}}\left|\left|\mathbf{a}_j^{HDV}\right|\right|_2^2 \\ \\ J_{jerk} = (1-\kappa)\frac{1}{a_{max}^2}\left|\left|\frac{1}{\Delta t}(\mathbf{a}^{CAV,pre} - \mathbf{a}^{CAV})\right|\right|_2^2 + \dots \\ \kappa\frac{1}{N_{sHDV} a_{max}^2}\sum_{j\in\mathcal{G}^{sHDV}}\left|\left|\frac{1}{\Delta t}(\mathbf{a}_j^{HDV,pre} - \mathbf{a}_j^{HDV})\right|\right|_2^2 \\ J_{slack} = \sum_{j\in\mathcal{G}_{fHDV}}\frac{1}{\gamma_{max}^2 N_{fHDV}}\left|\left|\gamma_j^{HDV}\right|\right|_2^2 \\ \textit{where } \kappa \in [0,1]\ , N_{sHDV} = \#\mathcal{G}^{sHDV},\ N_{fHDV} = \#\mathcal{G}_{fHDV} \end{aligned} \tag{2.38}$$

Note that the slack variable is present for all HDVs except for the leader, meaning that we divide by $(N_{HDV} - 1)$ for the slack objective. The objective is now further simplified by expanding the squared L2-norm in to more recognizable quadratic equations:

$$\begin{aligned} J_{efficiency} = \sum_{i=1}^{N} \Bigg(\frac{(1-\kappa)}{v_{max}^2} (v_i^{2,CAV} - 2v_i^{CAV} v_{desired} + v_{desired}^2) + \dots \\ \frac{\kappa}{N_{sHDV} v_{max}^2} \sum_{j \in \mathcal{G}^{sHDV}} (v_{i,j}^{2,HDV} - 2v_{i,j}^{HDV} v_{desired} + v_{desired}^2) \Bigg) \end{aligned} \tag{2.39}$$

$$J_{acc.\ mag.} = \sum_{i=1}^{N} \left(\frac{(1-\kappa)}{a_{max}^2} a_i^{2,CAV} + \frac{\kappa}{N_{sHDV} a_{max}^2} \sum_{j \in \mathcal{G}^{sHDV}} a_{i,j}^{2,HDV} \right) \tag{2.40}$$

$$\begin{aligned} J_{jerk} = \sum_{i=1}^{N} \Bigg(\frac{(1-\kappa)}{a_{max}^2 \Delta t^2} \left((a_i^{CAV,pre})^2 - 2a_i^{CAV,pre} a_i^{CAV} + (a_i^{CAV})^2 \right) + \dots \\ \frac{\kappa}{N_{sHDV} a_{max}^2 \Delta t^2} \sum_{j \in \mathcal{G}^{sHDV}} \left((a_{i,j}^{HDV,pre})^2 - 2a_{i,j}^{HDV,pre} a_{i,j}^{HDV} + (a_{i,j}^{HDV})^2 \right) \Bigg) \end{aligned} \tag{2.41}$$

$$J_{slack} = \sum_{j \in \mathcal{G}_{fHDV}} \sum_{i=1}^{N} \frac{1}{\gamma_{max}^2 N_{fHDV}} (\gamma_{j,i}^{HDV})^2 \tag{2.42}$$

The objective functions are now on an easily recognizable quadratic format.
As seen in the general quadratic optimization formulation on matrix form (2.34), there are several matrices that contain the specific objective functions and constraints. In (2.34), the optimization variable not only contain the system states and inputs, but it must also include the slack variables introduced in (2.16). Firstly, all optimization states are concatenated in to a single vector:

$$\begin{aligned} \mathbf{x}_{opt} = \Big[\mathbf{x}_{k+1}^{CAV} \dots \mathbf{x}_{k+N}^{CAV}\ \mathbf{x}_{1,k+1}^{HDV} \dots \mathbf{x}_{1,k+N}^{HDV} \dots \mathbf{x}_{N_{HDV},k+1}^{HDV} \dots \mathbf{x}_{N_{HDV},k+N}^{HDV} \dots \\ \dots \mathbf{u}_{k}^{CAV} \dots \mathbf{u}_{k+N}^{CAV}\ \mathbf{u}_{1,k}^{HDV} \dots \mathbf{u}_{1,k+N}^{HDV} \dots \mathbf{u}_{N_{HDV},k}^{HDV} \dots \mathbf{u}_{N_{HDV},k+N}^{HDV} \\ \gamma_{1,k}^{HDV} \dots \gamma_{1,k+N}^{HDV} \gamma_{N_{HDV}-1,k}^{HDV} \dots \gamma_{N_{HDV}-1,k+N}^{HDV} \Big]^T \in \mathbb{R}^{N_{vars} \times 1} \\ where\ N_{vars} = N_{states} + N_{slack} + N_{inputs},\ N_{states} = 3(N + N_{HDV} N), \\ N_{inputs} = 2(N + N_{HDV} N),\ N_{slack} = (N_{HDV} - 1)N \end{aligned} \tag{2.43}$$

The vector in (2.43) structure is simple. The CAV states come first, then the states for all N_{HDV} HDVs over the horizon are concatenated together. Afterwards, the CAV inputs and HDV inputs are concatenated. Finally the slack variables are added to form the final optimization variable vector. For example, in the scenario in figure 2.4 there are 4 vehicles in total, 1 CAV and 3 HDVs. Setting $N = 50$ time steps, this would result in $N_{vars} = 1100$ optimization variables. The initial conditions are also combined in to a vector:

$$\begin{aligned} \mathbf{x}_{0,opt} = \Big[\mathbf{x}_{k}^{CAV}\ \mathbf{x}_{1,k}^{HDV} \dots \mathbf{x}_{N_{HDV},k}^{HDV} \Big]^T \in \mathbb{R}^{N_0 \times 1} \\ where\ N_0 = 3(1 + N_{HDV}) \end{aligned} \tag{2.44}$$

It is important to note that $\mathbf{x}_{0,opt}$ in (2.44) is considered as a vector of constant terms, since the initial condition vehicle states are fixed for each MPC optimization and only change between each optimization step. In order to give insight how the optimization problem is formulated, a simple example will now be made. The example involves 2 HDVs and 1 CAV, where there is one leading HDV, HDV_2, one CAV which follows HDV_2, and lastly a HDV that follows the CAV, i.e. HDV_1. To simplify the scenario, the horizon length is only $N = 2$, meaning that the MPC controller predict 2 time steps in to the future, e.g. 0.2 *sec*. Figure 2.4 shows this simple scenario and how the prediction steps may be interpreted.
Firstly, the HDV and CAV states in this example are as follows:

$$\begin{aligned}
\mathbf{x}_0^{CAV} &= \begin{bmatrix} p_0^{CAV} \\ v_0^{CAV} \end{bmatrix},\ \mathbf{x}_1^{CAV} = \begin{bmatrix} p_1^{CAV} \\ v_1^{CAV} \end{bmatrix},\ \mathbf{x}_2^{CAV} = \begin{bmatrix} p_2^{CAV} \\ v_2^{CAV} \end{bmatrix} \\
\mathbf{x}_0^{HDV_1} &= \begin{bmatrix} p_0^{HDV_1} \\ v_0^{HDV_1} \end{bmatrix},\ \mathbf{x}_1^{HDV_1} = \begin{bmatrix} p_1^{HDV_1} \\ v_1^{HDV_1} \end{bmatrix},\ \mathbf{x}_2^{HDV_1} = \begin{bmatrix} p_2^{HDV_1} \\ v_2^{HDV_1} \end{bmatrix} \\
\mathbf{x}_0^{HDV_2} &= \begin{bmatrix} p_0^{HDV_2} \\ v_0^{HDV_2} \end{bmatrix},\ \mathbf{x}_1^{HDV_2} = \begin{bmatrix} p_1^{HDV_2} \\ v_1^{HDV_2} \end{bmatrix},\ \mathbf{x}_2^{HDV_2} = \begin{bmatrix} p_2^{HDV_2} \\ v_2^{HDV_2} \end{bmatrix}
\end{aligned} \tag{2.45}$$

The inputs and slack variables are:

$$\begin{aligned}
&\mathbf{u}_0^{CAV} = a_0^{CAV},\ \mathbf{u}_1^{CAV} = a_1^{CAV} \\
&\mathbf{u}_0^{HDV_1} = a_0^{HDV_1},\ \mathbf{u}_1^{HDV_1} = a_1^{HDV_1},\ \gamma_1^{HDV_1},\ \gamma_2^{HDV_1} \\
&\mathbf{u}_0^{HDV_2} = a_0^{HDV_2},\ \mathbf{u}_1^{HDV_2} = a_1^{HDV_2}
\end{aligned} \tag{2.46}$$

The optimization variable vector is then

$$\begin{aligned}
\mathbf{x}_{opt} = \Big[\mathbf{x}_1^{CAV}\ \mathbf{x}_2^{CAV}\ \mathbf{x}_1^{HDV_1}\ \mathbf{x}_2^{HDV_1}\ \mathbf{x}_1^{HDV_2}\ \mathbf{x}_2^{HDV_2}\ \dots \\
\mathbf{u}_0^{CAV}\ \mathbf{u}_1^{CAV}\ \mathbf{u}_0^{HDV_1}\ \mathbf{u}_1^{HDV_1}\ \mathbf{u}_0^{HDV_2}\ \mathbf{u}_1^{HDV_2}\ \gamma_1^{HDV_1}\ \gamma_2^{HDV_1}\Big] \in \mathcal{R}^{20\times 1}
\end{aligned} \tag{2.47}$$

By using the states and variables in (2.46) and forming them in to the optimization variable vector (2.43), the objective in (2.38) can be formulated on matrix form using quadratic matrix equations:

$$\begin{aligned}
&J_{total} = \frac{1}{2}\mathbf{x}_{opt}^T P \mathbf{x}_{opt} + q^T \mathbf{x}_{opt} \\
&where\ P = diag\left[\tfrac{(1-\lambda)(1-w_1)}{v_{max}^2} P_{efficiency} \quad \tfrac{(1-\lambda)w_1}{a_{max}^2 \Delta t^2} P_{comfort} \quad \tfrac{\lambda}{\gamma_{max}^2 N_{fHDV}} P_{slack}\right] \in \mathcal{R}^{20\times 20} \\
&q = -2v_{desired}\left[0 \quad (1-\kappa) \quad 0 \quad (1-\kappa) \quad 0 \quad \tfrac{\kappa}{N_{sHDV}} \quad 0 \quad \tfrac{\kappa}{N_{sHDV}} \quad 0 \dots \quad 0\right]^T \in \mathcal{R}^{20\times 1} \\
&and\ P_{efficiency} = diag\left[0 \quad (1-\kappa) \quad 0 \quad (1-\kappa) \quad 0 \quad \tfrac{\kappa}{N_{sHDV}} \quad 0 \quad \tfrac{\kappa}{N_{sHDV}} \quad 0 \quad 0 \quad 0 \quad 0\right] \in \mathcal{R}^{12\times 12} \\
&P_{comfort} = \begin{bmatrix} (1-\kappa)(1+w_2) & -2w_2(1-\kappa) & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\kappa(1+w_2)}{N_{sHDV}} & \frac{-2\kappa w_2}{N_{sHDV}} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathcal{R}^{6\times 6} \\
&P_{slack} = I_{2\times 2} \in \mathcal{R}^{2\times 2}
\end{aligned} \tag{2.48}$$

The constraints are also reformulated in to matrix form, where they are split in to equality and inequality constraints according to the general quadratic optimization problem (2.34).

$$\begin{aligned} &A_{eq}\mathbf{x}_{opt} = b \\ &where\ A_{eq} = \begin{bmatrix} A_{VM} \\ A_{OVRV} \\ A_{MPC\ pred.\ heuristic} \end{bmatrix},\ b_{eq} = \begin{bmatrix} b_{VM} \\ b_{OVRV} \\ b_{MPC\ pred.\ heuristic} \end{bmatrix} \end{aligned} \tag{2.49}$$

In (2.56), A_{VM} is the matrix of the vehicle model presented in (2.9). The conents are the entire trajectories for each vehicle, which was presented in (2.4). b_{VM} are all the constant terms from these equations, i.e. the matrix of initial conditions from (**??**). For this example, A_{VM} and b_{VM} are formulated as

$$\begin{aligned} &A_{VM} = \begin{bmatrix} I_{12\times12} & B_{VM} & \mathbf{0}_{12\times2} \end{bmatrix} \in \mathcal{R}^{12\times20} \\ &where\ B_{VM} = diag\begin{bmatrix} B_{BM} & B_{BM} & B_{BM} \end{bmatrix} \in \mathcal{R}^{12\times6},\ B_{BM} = \begin{bmatrix} \frac{\Delta t^2}{2} & 0 \\ \Delta t & 0 \\ \frac{3\Delta t^2}{2} & \frac{\Delta t^2}{2} \\ \Delta t & \Delta t \end{bmatrix} \in \mathcal{R}^{4\times2} \end{aligned} \tag{2.50}$$

$$b_{VM} = diag\begin{bmatrix} A_{BM} & A_{BM} & A_{BM} \end{bmatrix} \mathbf{x}_{0,opt} \in \mathcal{R}^{12\times6},\ A_{BM} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 2\Delta t \\ 0 & 1 \end{bmatrix} \in \mathcal{R}^{4\times2} \tag{2.51}$$

The OVRV model constraints, presented in (2.15) contains both equality and inequality constraints, and the equality part is represented in matrix form as A_{OVRV} and the constant terms in b_{OVRV}.

$$\begin{aligned} &A_{OVRV}\ \in \mathcal{R}^{2\times20} = \\ &\begin{bmatrix} -\rho & \beta & 0 & 0 & \rho & (\alpha-\beta) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\rho & \beta & 0 & 0 & \rho & (\alpha-\beta) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \\ &where\ \rho = \frac{\alpha V_{max}}{h_{max} - h_{min}} \end{aligned} \tag{2.52}$$

$$b_{OVRV} = \rho \begin{bmatrix} -h_{min} \\ -h_{min} \end{bmatrix} \in \mathcal{R}^{2\times1} \tag{2.53}$$

As mentioned, the MPC prediction heuristics for the leading HDV is simply keeping the acceleration constant. So, $A_{MPC\ pred.}$ and $b_{MPC\ pred.}$ are

$$\begin{aligned} &A_{MPC\ pred.} \in \mathcal{R}^{2\times20} = \\ &\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{aligned} \tag{2.54}$$

$$b_{MPC\ pred.} = \begin{bmatrix} a_0^{lead} \\ a_0^{lead} \end{bmatrix} \in \mathcal{R}^{2\times1} \tag{2.55}$$

Finally, the inequality constraint also follow the general quadratic optimization format where

$$\begin{aligned} &G_{ineq}\mathbf{x}_{opt} = H_{ineq} \\ &where\ G_{ineq} = \begin{bmatrix} G_{OVRV} \\ G_{safety} \\ G_{CAV\ input} \\ G_{velocity} \end{bmatrix},\ H_{ineq} = \begin{bmatrix} H_{OVRV} \\ H_{safety} \\ H_{CAV\ input} \\ H_{velocity} \end{bmatrix} \end{aligned} \tag{2.56}$$

Firstly, the remaining OVRV constraints are contained in G_{OVRV} and the respective constant terms in H_{OVRV}.

$$G_{OVRV} \in \mathcal{R}^{4\times20} = \begin{bmatrix} 0 & -\beta & 0 & 0 & 0 & (-\alpha+\beta) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\beta & 0 & 0 & 0(-\alpha+\beta) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 & 0 & (\alpha-\beta) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta & 0 & 0 & 0(\alpha-\beta) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{2.57}$$

$$H_{OVRV} = \begin{bmatrix} \alpha V_{max} \\ 0 \\ \alpha V_{max} \\ 0 \end{bmatrix} \in \mathcal{R}^{4\times1} \tag{2.58}$$

The safety constraints from (2.26) and (2.25) are formulated in matrix form using G_{safety} and the constant terms are considered in H_{safety}.

$$G_{safety} \in \mathcal{R}^{4\times20} = \begin{bmatrix} -1 & 0 & 0 & 0 & 1 & t_{min}^{HDV} & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & t_{min}^{HDV} & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & t_{min}^{CAV} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & t_{min}^{CAV} & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & \dots & 0 \end{bmatrix} \tag{2.59}$$

$$H_{safety} = \begin{bmatrix} -h_{min} \\ -h_{min} \\ -h_{min} \\ -h_{min} \end{bmatrix} \in \mathcal{R}^{4\times1} \tag{2.60}$$

The limits on acceleration input for the CAV in (2.29) are put in matrix form by $G_{CAV\ input}$ and $H_{CAV\ input}$.

$$G_{CAV\ input} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix} \in \mathcal{R}^{4\times20} \tag{2.61}$$

$$H_{CAV\ input} = \begin{bmatrix} -a_{min} \\ a_{max} \\ -a_{min} \\ a_{max} \end{bmatrix} \tag{2.62}$$

The velocity for the CAV and the leading HDV are also constrained, and in matrix from these constraints are represented by $G_{velocity}$ and $H_{velocity}$.

$$G_{velocity} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & \dots & 0 \end{bmatrix} \in \mathcal{R}^{4\times 20} \tag{2.63}$$

$$G_{velocity} = \mathbf{0}_{4\times 1} \in \mathcal{R}^{4\times 1} \tag{2.64}$$

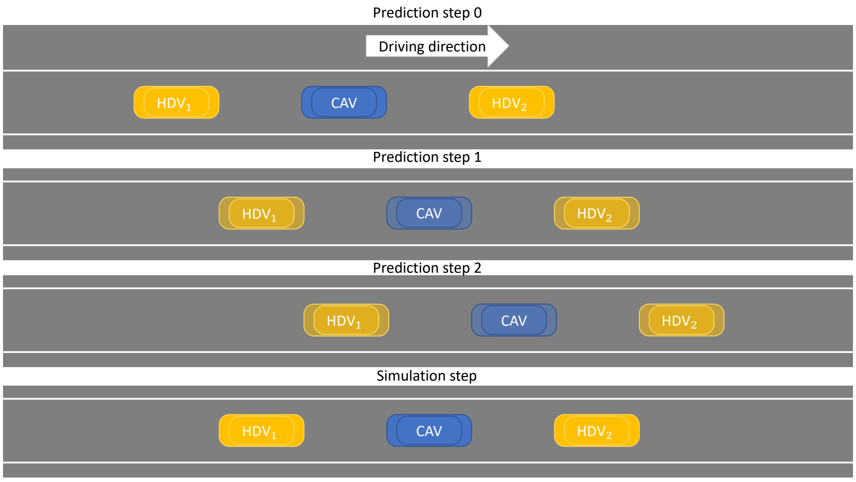


Figure 2.4: Example of MPC problem formulation for simple 4 vehicle scenario. Prediction step 0 show the initial states from the real environment which are used for predicting in to the future. Prediction step 1 and 2 are performed by the MPC controller to optimize the vehicle trajectories. The simulation step is again in the real environment, where the inputs used to go from prediction step 0 to prediction step 1 are applied by the controller on to the CAV in the simulation environment. This procedure occurs at each simulation step, albeit with longer time horizon for better predictions.

3

Experiments

This chapter covers how the experiments were implemented, how the experiments were set-up and the results along with a discussion.

3.1 Implementation

This section gives insights into the different aspects of implementing the MPC controller and how the traffic simulator was set-up for the experiments. Firstly, the traffic simulator is introduced and covers how the simulations work, what driving behaviour is used as well as how the simulations are designed. Secondly, the Python programming implementation is briefly covered, where the programming structure and functions are explained along with the optimization toolbox that is used

3.1.1 Traffic simulator - PTV VISSIM

PTV VISSIM is a high fidelity, microscopic traffic simulator. The simulator is able to accurately simulate real driving behaviour of vehicles on a variety of traffic conditions and road types, from congested highway traffic to highly complex intersections with multiple vehicle types. The simulator is highly customizable and allows one to design road networks from scratch, or from map data. The traffic driving behaviours can also be fully customized so that different aggressiveness levels of drivers can be simulated. For the purposes of the thesis, the simulator is used to gain insight into how the altruistic controller behaves in life-like conditions.
In the thesis, the traffic simulator is first used to design scenarios for testing the controller behaviour. The scenarios includes both design of road network and design of driving behaviours in traffic. As the focus is to control a CAV on a highway with multiple lanes. The simulations should be simple yet representative of steady highway traffic in differing conditions. Thus, the Ehra Leissen test track in Germany was used as a base for modeling the highway scenarios. In order to reduce the programmatic complexity of the simulator interaction with the MPC controller, and that it was sought after to use straight road sections, the Ehra Leissen test track was ideal. The long straight road sections ensure that the traffic is not affected by sharp turns, which helps with repeatability of tests. The very wide turns also aid in this regard.
There is also the ability to adjust the driving behaviour of vehicles in the simulator. The simulator uses the Wiedemann99 (W99) driver model. The W99 model is a psycho-physical driver model that not only accounts for headway and relative veloc-

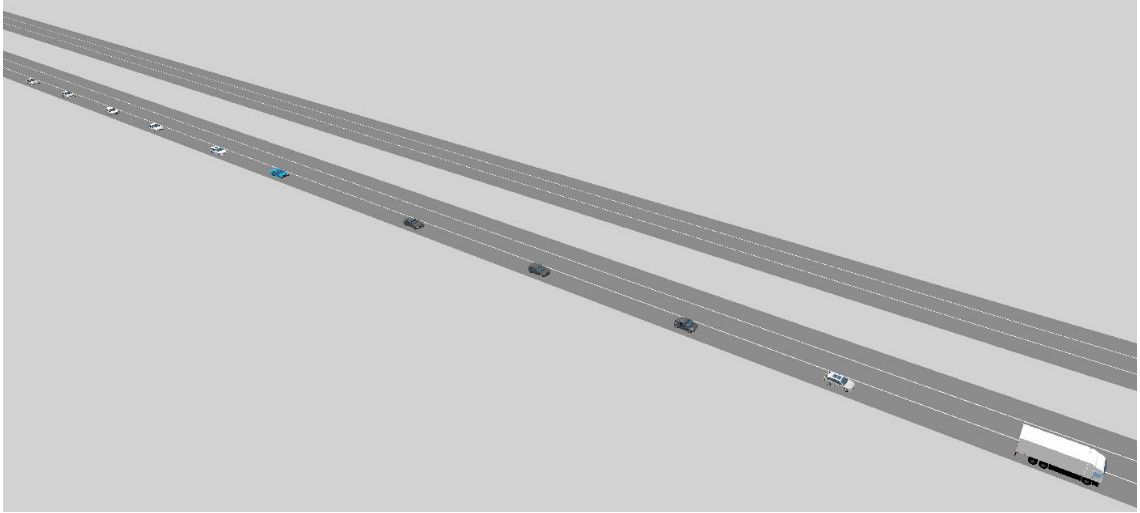


Figure 3.1: The CAV (in blue) during a single lane traffic simulation

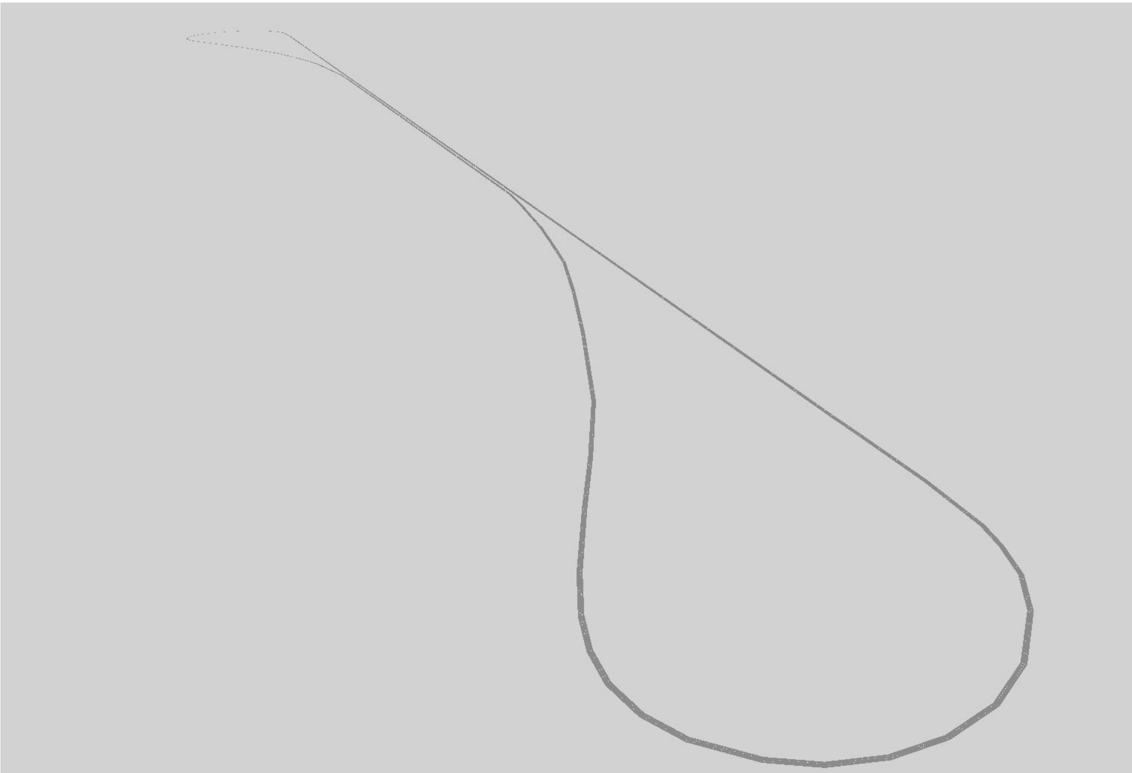


Figure 3.2: The Ehra Leissen test track model in PTV VISSIM

ity, like OVRV, but also introduces psychological elements to the behaviour of the vehicle. Unlike the OVRV model, W99 is also stochastic. This means that the driver behaviour will have some level of randomness to the actions taken. Furthermore, W99 is a non-tractable model consisting of both function and logical expressions, meaning that no analytic function exists.

The W99 model has several parameters, and Figure 3.7 shows the W99 parameters that may be customized in Vissim. CC0 is the desired vehicle standstill distance,

CC1 is the headway time (in seconds) that the vehicle wants to keep. CC0 and CC1 define the safe vehicle headway by $h_{safe} = CC0 + CC1\dot{v}$ where v [m/s] is the vehicle velocity. CC2 controls the following variation in meters by defining the oscillation boundaries by $h_{safe} \geq h \geq h_{safe} + CC2$. CC3 is a threshold parameter defining when the vehicle recognized a slower vehicle ahead of it, starts to slow down and entering a following state. CC4 and CC5 controls the speed differences during the follwing state for deceleration and acceleration respectively where smaller values increases the vehicles sensitivity to accelerations and deceleration of the preceding vehicle. CC6 controls the speed dependency of oscillations, where larger values lead to larger velocity with increasing distance while the vehicle is in the following state. CC7, CC8 and CC9 controls the acceleration during oscillation, at standstill and at 80 km/h respectively.
In the Vissim simulations, the following parameters for W99 [9] are used, CC0= 1 [m], CC1= 0.9 [s], CC2= 1 [m], CC3= -8, CC4= -0.05, CC5= 0.05, CC6= 1, CC7= 10 [m/s^2], CC8= 10 [m/s^2], CC9= 10 [m/s^2]. Furthermore, the look ahead distance is set to 150 m with 2 vehicles observed at most and the look back distance is set to 100 m. For single lane scenarios, there are 5 HDVs behind the CAV. For multi lane scenarios, there are 15 HDVs behind the CAVs with 3 CAVs in total.

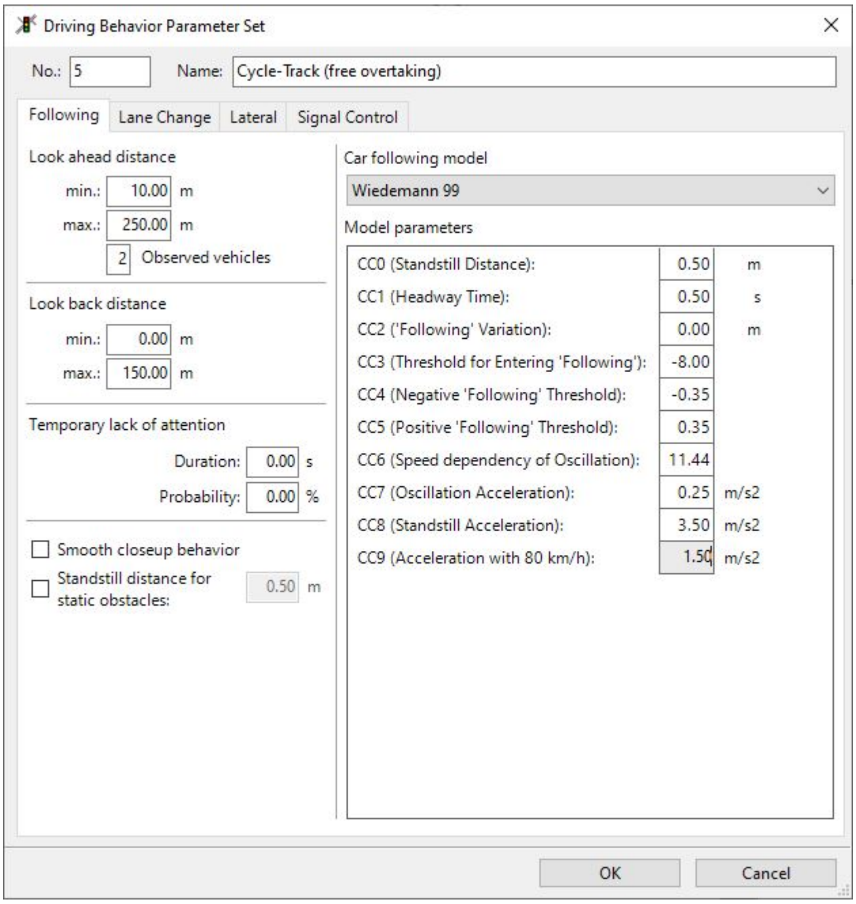


Figure 3.3: Customizable driver model parameters in PTV VISSIM

For the purposes of the experiments, the driving behaviour that is important is the level of aggressiveness of drivers. The parameters that correspond to aggressive driving behaviour are the standstill distance (CC0), headway time (CC1), following variation (CC2), oscillation acceleration (CC7), standstill acceleration (CC8) and acceleration at 80 km/h. Furthermore, the look ahead distance and number of observed vehicles are crucial parameters for aggressive driving behaviour, as these will indicate how late reactions occur and how much the driver is able to see in front. To show the effects of altruism more clearly, these parameters are set very aggressive. This means that the vehicles will be following at close distance at high speed, not many vehicles will be observable and the traffic oscillations will be very high.

In order to evaluate the altuistic behaviour of the controller, a few scenarios are designed. Firstly, to validate that the optimization is correct, a simple simulation using the OVRV model is implemented for a single-lane scenario. Then, a similar scenario is implemented in the simulator so that the model-mismatch between OVRV and W99 may be evaluated. Both of these simulations are single lane scenarios only, which reduces the variability to only the driver model used. The final scenarios are multi-lane freeway driving. Here, the altruistic behaviour is extended by also including the lane-changing behaviour of both CAVs and HDVs. More details are covered in the scenario descriptions in the result section.

3.1.2 Computer program

PTV VISSIM uses a Microsoft COM-Server API, through which all aspects of VISSIM may be controlled. Since COM-Servers are widely used in most popular programming languages, Python was chosen as the language to implement the MPC controller and simulations. This was due to the simplicity of implementation using Python and also that there are many readily available packages for optimization, especially high performance optimization software. The three main functions of the software implementation were the reading the surrounding vehicles in the simulator, performing the MPC optimization and then applying the optimal control input to the CAV in the simulation.

VISSIM is built around a link and connection type of structure, similar to edges and nodes in graph theory. Vehicles are then categorized in to which link(road segment) or connection(connecting road segments to road segments) they are currently on. By the way VISSIM deals with these links and connections, the distance between two vehicles on different links or connections is not trivial to calculate. This is because the internal distance measurement in VISSIM only considers how far along a link vehicles have traveled. Thus, the road network was considered as a bi-directional graph structure in order to easily calculate distances. The shortest path between the two vehicles was calculated from a simple graph search and the distance was calculated based on the link lengths between the two vehicles and the distance they traveled on their current link. Algorithm 1 shows a simple algorithm that will calculate the headway between the CAV and all nearby vehicles.

Algorithm 1: Calculate headway between HDVs and CAV

Result: List of headways

Init: Get link and lane number for HDVs in range;

for i = *HDVs in range* **do**

  $L_{links,\ forwards}$ =graph search

  $D_{forwards} = (P_{CAV} - L_{CAV,link}) + L_{links,\ forwards} + HDV_{pos}$;

  $D_{backwards} = P_{CAV} + L_{links,\ backwards} + (P_{HDV} - L_{HDV,link})$;

  $H[i] = min(D_{forwards}, D_{backwards})$

end

In order to speed up calculation times for the MPC controller, the workload is offloaded to a remote cloud server and the result is then downloaded to the simulation in order to control the vehicle. The simulator sends data regarding HDVs initial positions, velocities and accelerations. The server then receives this information and builds the optimization problem on matrix form according to the structure in **??**. The optimization problem is then solved by the optimization toolbox CVX-OPT in Python, where the quadratic solver MOSEK uses an algorithm incorporating interior-point methods to perform the optimization efficiently. The server then returns the optimal control inputs, i.e. a^*_{CAV} and the optimal function value J^* for each lane. Algorithm 2 shows how this was implemented.

Algorithm 2: Remote MPC computation

Result: CAV lane change and acceleration input

Init: Get surrounding HDVs data;

Simulator: Transmit HDV data to server for the 3 lanes, wait for response;

Server: Recieve HDV data;

Server: Spawn parallel processes to concurrently handle optimizations on each lane;

for i = *3* **do**

  Server: Set up optimization matrices, constraints and objective function;

  Server: Perform optimization using CVX-OPT with MOSEK solver;

end

Server: Return optimal accelerations $\left(a_k^{*,left}, a_k^{*,right}, a_k^{*,current}\right)$
and objective function values $\left(J_k^{*,left}, J_k^{*,right}, J_k^{*,current}\right)$;

Simulator: Receive acceleration inputs and objective function values ;

Simulator: Make the lance change decision:

$$\delta_k^{CAV} = \begin{cases} 1, & \text{if } J_k^{*,left} < min\left(J_k^{*,current}, J_k^{*,right}\right) \\ -1, & \text{if } J_k^{*,right} < min\left(J_k^{*,current}, J_k^{*,left}\right) \\ 0, & \text{otherwise} \end{cases}$$

The final computer program is then presented in Algorithm 3, where the above components are used to perform the most critical tasks.

Algorithm 3: Main program

Result: Simulation data
Init: Get altruism parameter κ, simulation length and scenario set-up;
Spawn vehicles on the road; **for** k = *simulation length* **do**
  Record HDV and CAV data at time step k;
  Function: Calculate headway between HDVs and CAV;
  Function: Remote MPC computation Apply the optimal control input to the CAV;
  Take step in simulator;
end

3.2 Results

The result section covers the experiments conducted for different scenarios in the simulator. Firstly, a baseline simulation using only the OVRV model is covered. Then, the VISSIM simulator is used to obtain results for a single lane scenario. Lastly, the VISSIM simulator is again used for two multi lane scenarios. The results of all scenarios are then discussed.

3.3 Single-lane OVRV

The single lane OVRV simulations can be considered as a baseline for the MPC controller and is used to validate that altruism can affect the traffic efficiency and vehicle comfort. The single lane OVRV scenario concerns vehicle placements as seen in figure 3.4.

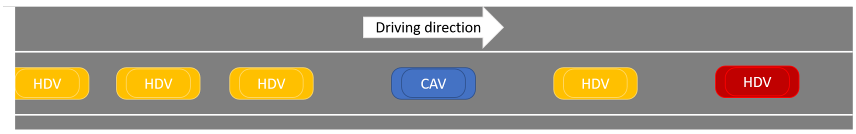


Figure 3.4: Single lane OVRV scenario

In figure 3.4, the blue vehicle is the CAV, the yellow vehicles are HDVs that drive using the OVRV driver model. The red vehicle is also a HDV, but it has a predetermined sinusoidal acceleration profile. The red HDV will cause disturbances on to the following vehicles, where the CAV will try to optimize it's own movements and thus influence the yellow HDVs behind it.

3.4 Single-lane VISSIM

The single lane VISSIM simulations use an identical scenario set-up as the previous OVRV simulations. This may help to give insights in to differences between the two models, W99 and OVRV.

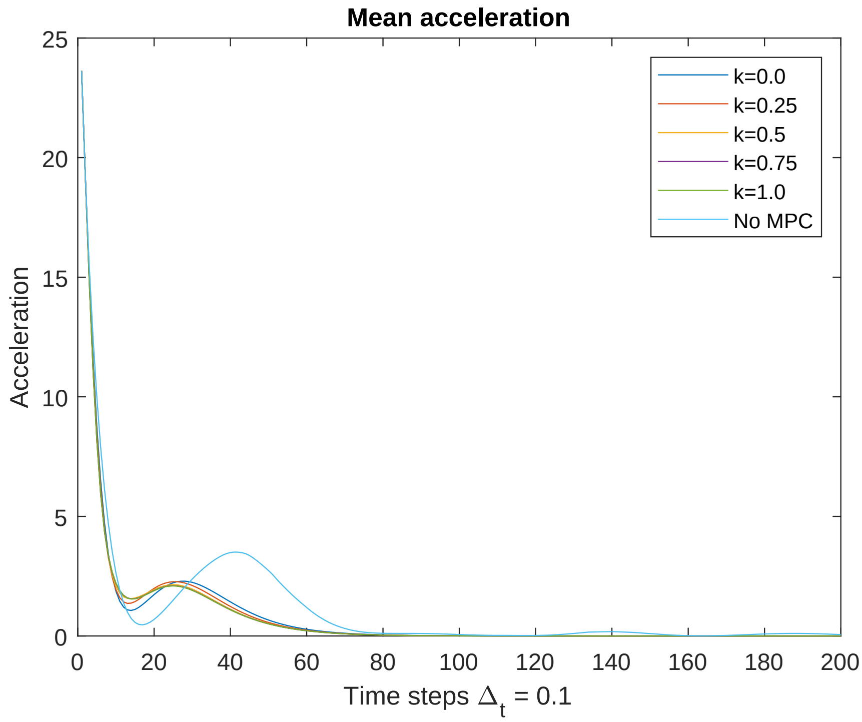


Figure 3.5: OVRV mean acceleration

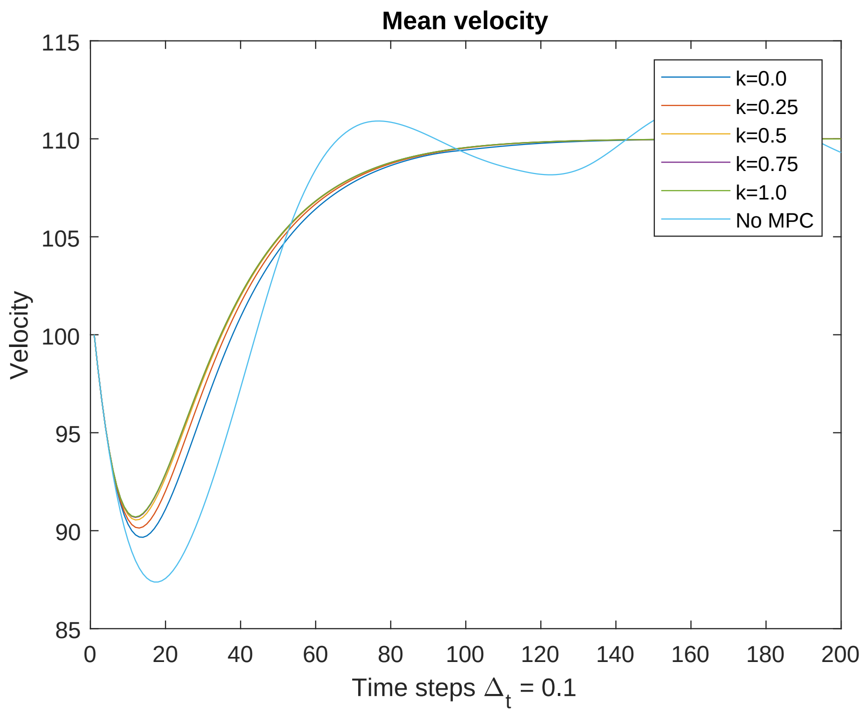


Figure 3.6: OVRV mean velocity

In figure 3.8, the CAV and the yellow HDVs are influenced by the sinusoidal acceleration profile of the leading, red HDV. The red HDVs acceleration and braking behaviour causes oscillations throughout the following traffic, which the CAV tries to mitigate.

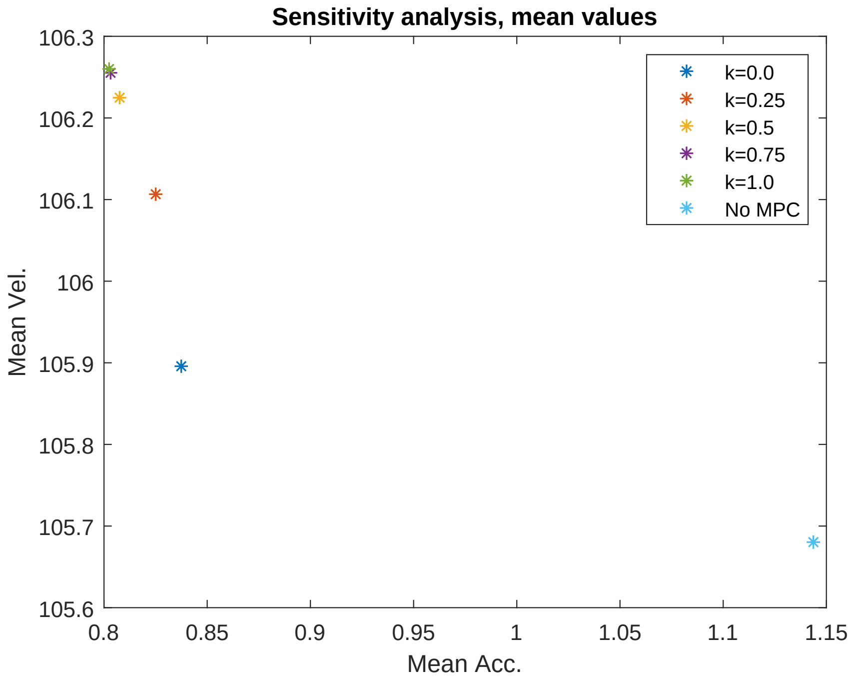


Figure 3.7: OVRV sensitivity analysis

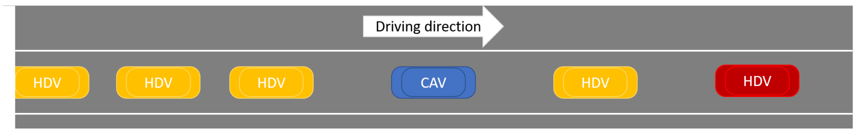


Figure 3.8: Single lane VISSIM scenario

3.5 Multi-lane VISSIM

For the multi lane VISSIM simulations, there are two scenarios. The first one concerns a multi lane scenario similar to the single lane ones above. Here, the single CAV tries to mitigate the oscillations by leading HDVs on all three lanes. The idea is to have a scenario where the traffic congestion is high and distances between HDVs is rather small. This creates a very harsh traffic scenario.

3.5.1 Scenario 1

In figure 3.12, there are several yellow HDVs behind and infront of the CAV on all lanes. These HDVs use the W99 model and are controlled by the simulator. The CAV calculates the optimal control input for all lanes and change to the lowest function value. On each lane, there are also leading HDVs in red. Their acceleration profiles are again sinusoidal so that they create oscillations in the traffic.
The data to be evaluated is how different levels of altruism affects the mean acceleration and velocity of all yellow HDVs in the simulation. Figure 3.13 and 3.14 presents results of the mean accelerations and velocities respectively. Since these results are difficult to interpret, figure 3.15 is instead used as a way of visualizing how altruism affects the two metrics.

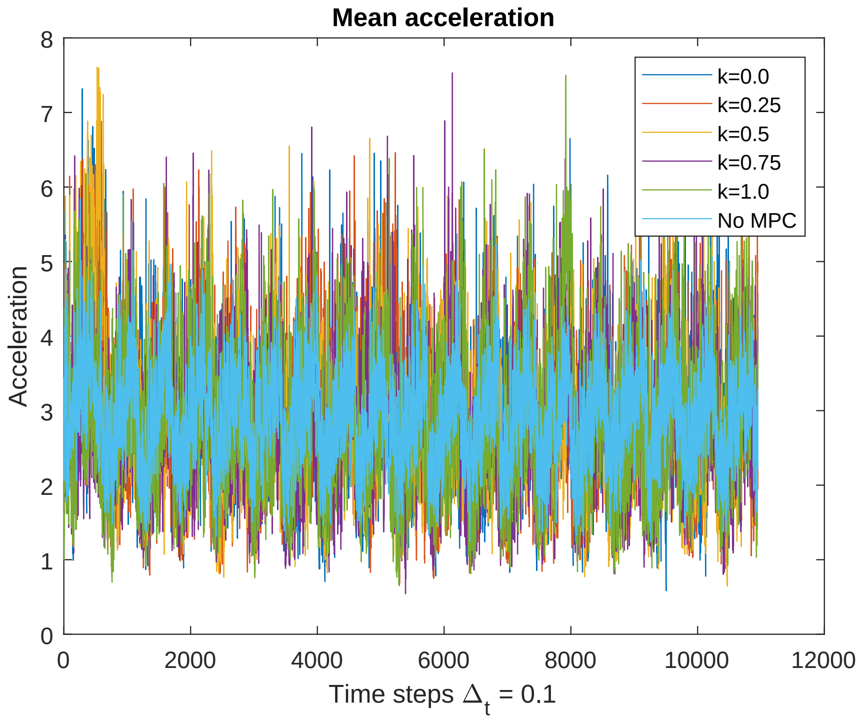


Figure 3.9: VISSIM single lane: Mean acceleration of all yellow HDVs

3.5.2 Scenario 2

The second multi lane scenario is depicted in figure 3.16. Here, the scenario is designed to show the lane changing behaviour of the CAV under complete altruistic and selfish control strategy. There are again three lanes with yellow HDVs behind the CAV in each lane. On the lane to the left of the CAV, there is a leading HDV in red. The leading HDV causes oscilations on that lane and thus the traffic will be worse. The middle lane is free driving, meaning that the leading yellow HDV behaves according to the W99 model. The lane to the right of the CAV is free. A red lane change maneuver represents a selfish strategy, where this control action would reduce the acceleration of the selfish CAV. Simply, the CAV will not need to decelerate at all and keep its own efficiency in an empy lane. A green lane change maneuver represents the altruistic control strategy where the CAVs objective is to reduce the traffics overall accelerations and increase the efficiency. By switching to that lane, the idea is that the traffic will, overall, perform better than if the control strategy is selfish. The results are presented as mean acceleration of all vehicles in figure 3.17, mean velocity in figure 3.18 and a sensitivity analysis in figure 3.19. Note that in this case, only selfish driving ($\kappa = 0$) and fully altruistic driving ($\kappa = 1$) are evaluated.

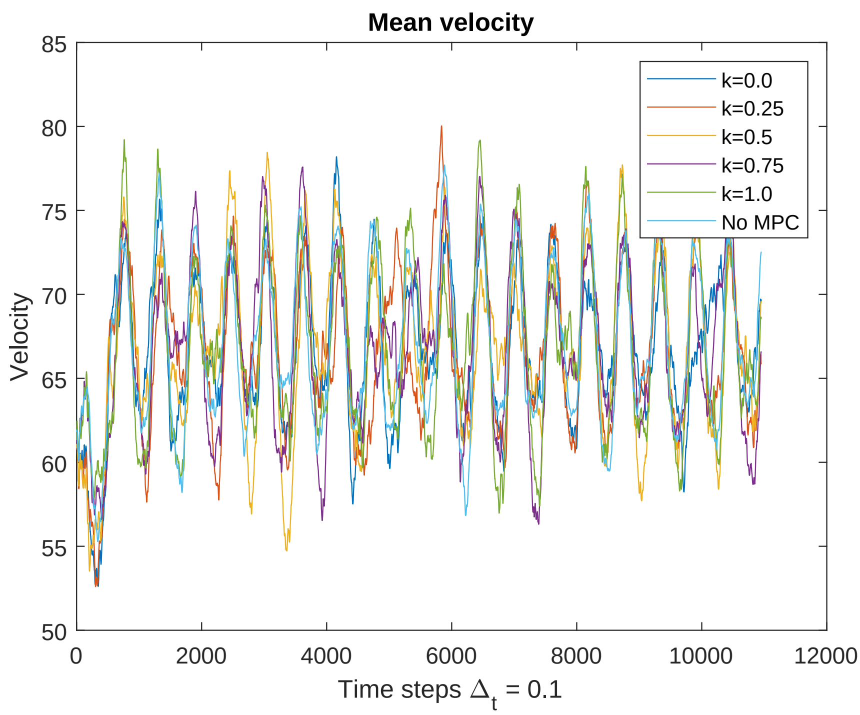


Figure 3.10: VISSIM single lane: Mean velocity of all yellow HDVs

3.6 Discussion

The implementation of the MPC controller was rather simple, and using Python along with a highly optimized optimization toolbox and quadratic solver helped to speed up simulation times. However, there was difficulty to validate the problem set-up due to the large size of coefficient matrices for constraints and objective function, along with the large number of optimization variables. Significant time had to be spent to ensure that the matrices were correct and that the optimization makes sense. Furthermore, since the optimization problem is quite large and will be calculated for each simulation step, some effort had to be made in order to optimize code and increase performance. Therefore the choice of using cloud computing was critical. For a multi-lane simulation, the program needs to solve three concurrent optimizations, one of each lane, and then pick the lowest cost among these three. This means that the program may be ran in parallel, and thus computation time is reduced if these are ran as separate processes. Each process then ran the optimization problem using the MOSEK solver. This solver is also multi-threaded and is able to fully utilize multiple CPU cores. This means that in order to efficiently run a large number of simulation batches, high CPU core count was critical. Fortunately, cloud computing solutions widely offer computational servers that can be used, which sped up simulation times significantly.

With regards to implementation of the MPC control strategy on a real-life vehicle, the inherent use of multi-processing is highly beneficial. We believe that by utilizing parallel computing for gathering information from sensors and running the highly efficient optimization solver, along with switching to more high performance and optimized C code, we will be able to achieve near real-time performance of the al-

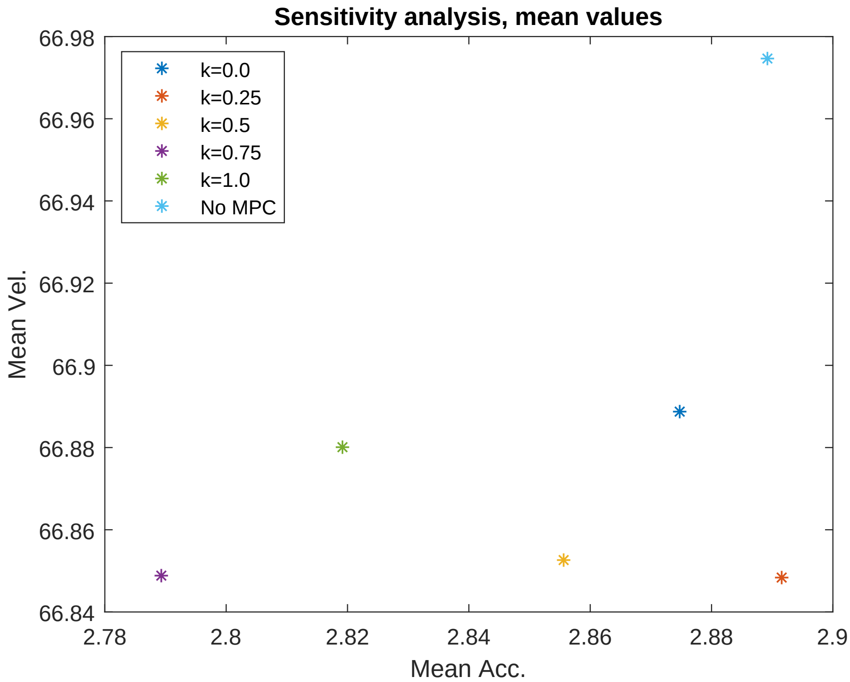


Figure 3.11: VISSIM single lane: Sensitivity analysis for differing altruism levels

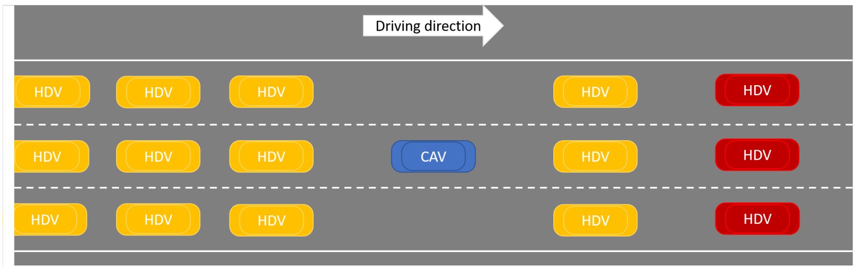


Figure 3.12: Harsh multi lane VISSIM scenario

gorithm. This means that there may be vehicles in the traffic run these controllers in a decentralized way and thus latency problems of centralized control are eliminated, creating a much more responsive control system. Furthermore, modern high performance computing hardware is faster than server grade hardware, and thus the controller implementation is very feasible.

Another controller and simulation related implication was the parameter tuning. It was quite difficult to choose suitable parameters for both the controller and the traffic simulator. The thought behind using the OVRV simulations was to allow for controller tuning in a more controller way. However, the model mismatch and stochastic behaviour of the W99 model in the simulator proved to be difficult for the controller that was tuned on the OVRV model. Instead, everything had to be tuned together. Since the simulation times were long, even when using parallel processing on high core count server CPUs, the tuning process was lengthy and difficult. And the controller may need even further refinement. The first issue was how to tune the OVRV models parameters to predict the vehicles in the simulator, where it seemed that the headway parameters were important for the smoothness of the controller.

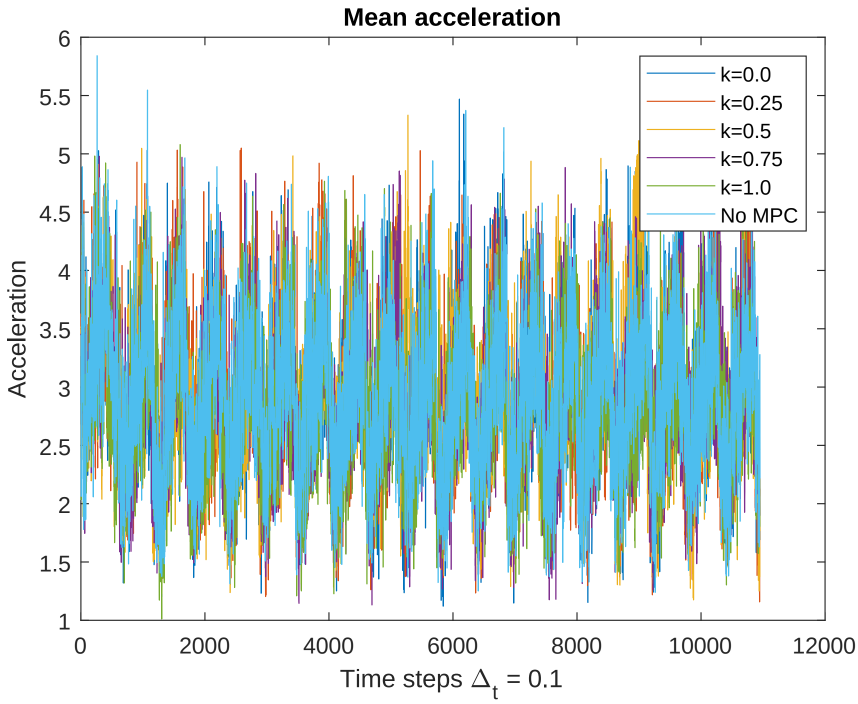


Figure 3.13: Mean acceleration of all yellow HDVs

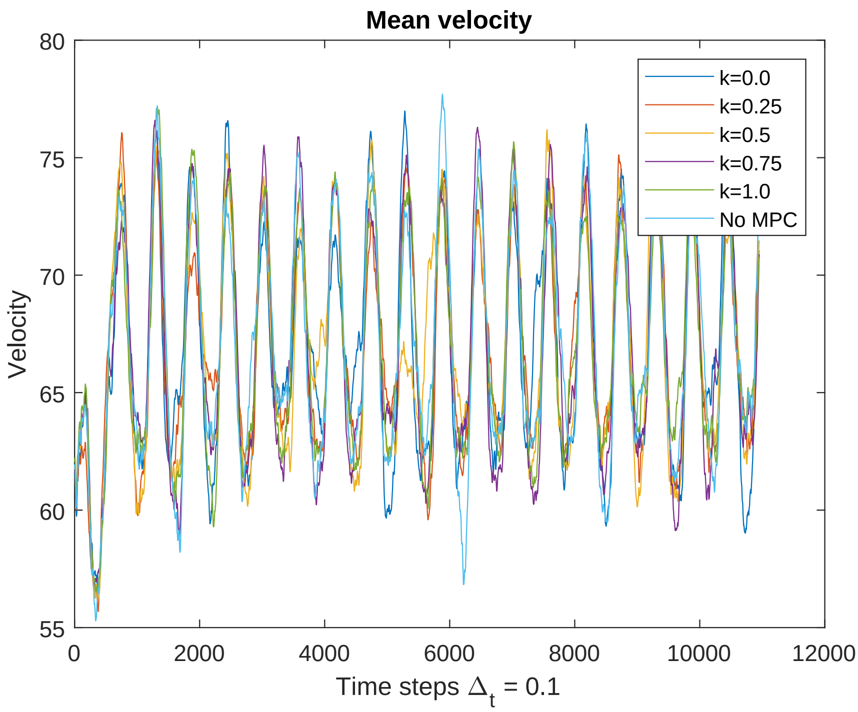


Figure 3.14: Mean velocity of all yellow HDVs

Another issue was the simulator itself, where there were many parameters that could be tweaked. Since the purpose of the control strategy is to improve highly congested and aggressive traffic scenarios, the W99 model had to be modified to achieve this. Thus, it was difficult to achieve a aggressive yet life-like behaviour of the vehicles. By

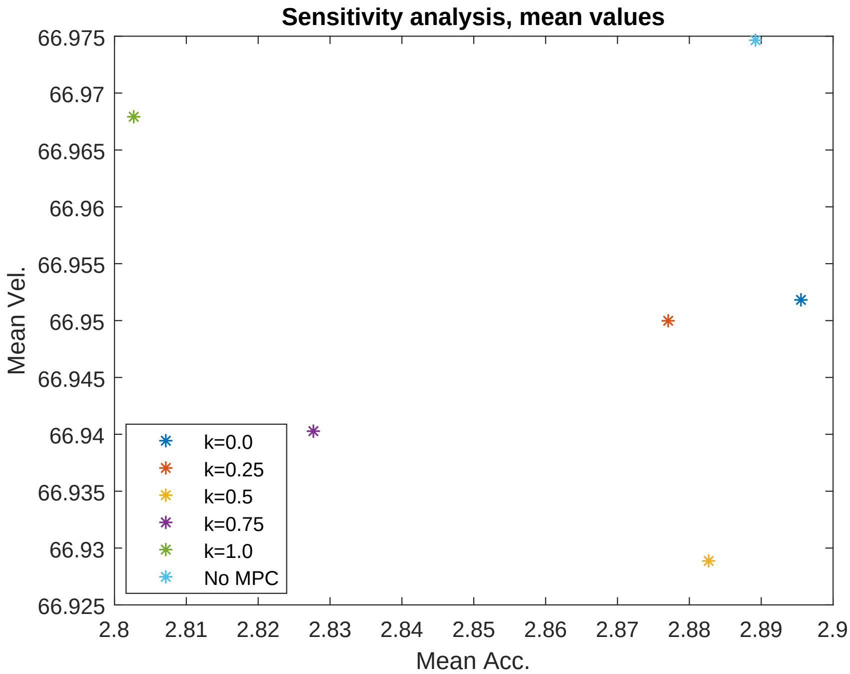


Figure 3.15: Sensitivity analysis for differing altruism levels

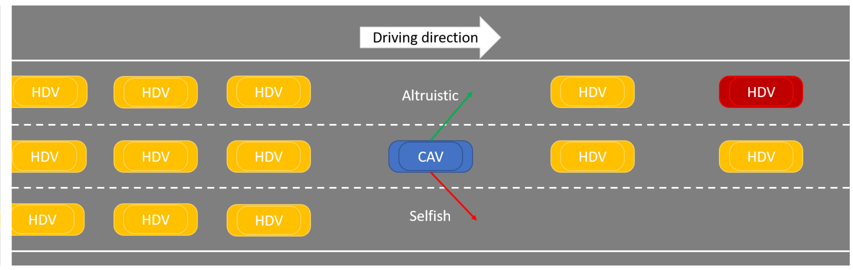


Figure 3.16: Harsh multi lane VISSIM scenario

this tuning, the model mismatch may also have been further increased, rendering the predictions by the OVRV model less reliable. For future work on this topic, it would be interesting to investigate other possible control strategies, e.g. robust control methods. It may be possible to account more for this model mismatch by going to robust control, thus allowing for better control inputs.

As mentioned, the MPC controller was tested on both the simpler OVRV simulations and the more complex Vissim simulations. As a starting point for validating altruism and the MPC controller, the OVRV model shows great initial potential. The altruism works well and the results are distinctively improving when altruism increases, which from a theoretical standpoint is correct. This then means that the controller is valid and that the MPC formulation works as intended. When moving to more complex simulations, there is less distinctiveness in the results. Looking at the single lane simulations of both OVRV and Vissim, it is evident that the benefits of altruism in the Vissim scenario is not as clear. However, there is a general trend, where the higher levels of altruism are clustered together. This seems to indicate that the actual altruism number is less important, but the effect of the κ parameter

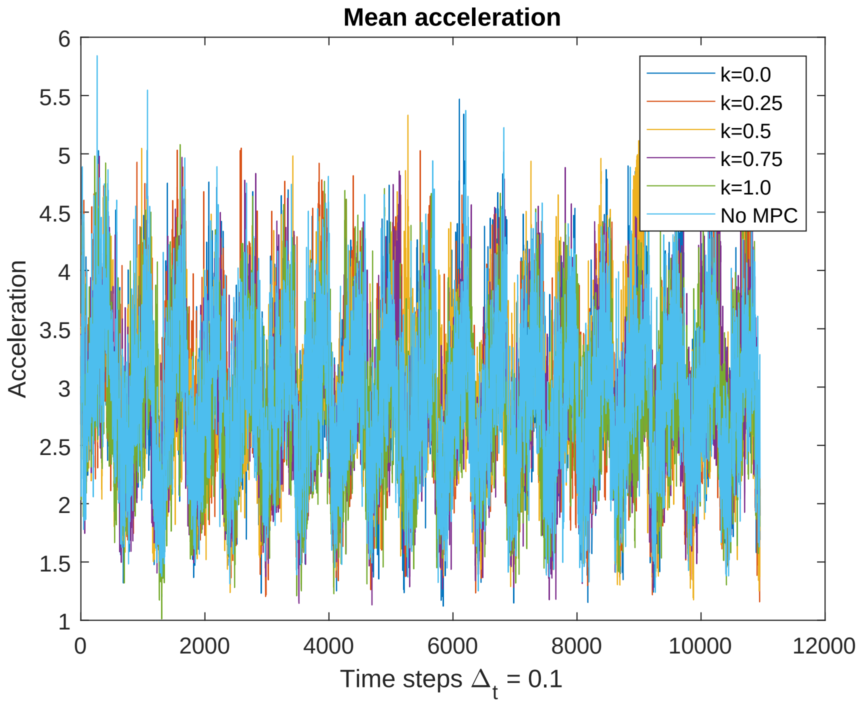


Figure 3.17: Mean acceleration of all yellow HDVs

on the objective function is more important, i.e. the scaling. The results also seem to indicate that either a lower altruism level ≤ 0.5 or a higher altruism level ≥ 0.75 are more meaningful as representing the selfish and altruistic behaviour respectively. rather than the actual κ value. Furthermore, comparing OVRV and Vissim, it is clear that the *No MPC* cases differ significantly. Again, since the model mismatch is quite significant between OVRV and W99, this is to be expected, and W99 even performs on the same level as a selfish MPC controller.

These behaviours are again present in the multi-lane scenarios. A more interesting case is the second multi lane scenario. In comparison to having all lanes in high congestion, this scenario was aimed to showcase that a altruistic CAV will sacrifice itself for the benefit of the other HDVs. This behaviour is clearly shown in the result, as the decrease in average acceleration is significantly improved when being altruistic. The 7% improvement can be compared with the roughly 3% improvement in multi lane scenario 1. These results suggest that the altruistic improvement comes more from lane change decisions rather than purely controlling acceleration input. It's quite clear also that if all lanes are highly congested, the selfish goal and the altruistic goal of reducing acceleration is similar. If the selfish CAV reduces it's own acceleration, by e.g. leaving a larger headway, this must mean that the HDVs will see a positive improvement as well. Conversely, if the CAV is altruistic, it may also leave a larger headway, resulting in similar improvements to itself and the following vehicles. Thus, in scenario 1, the improvements are marginal and that is what one may expect. In multi lane scenario 2, since the vehicles will act much more differently, the improvements are greater. Thus, for future work, scenario design has to be taken in to consideration so that the real benefits of altruism can be more clear. Furthermore, for future work it is important to realize what kind of traffic

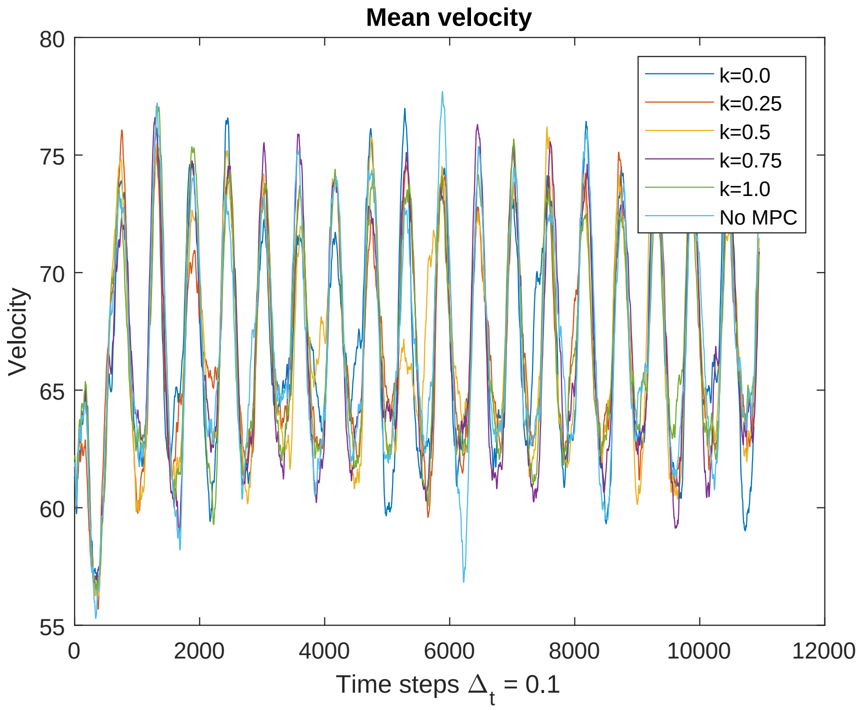


Figure 3.18: Mean velocity of all yellow HDVs

situations that benefit from altruism. This may hopefully lead to a more situational aware controller, which will adapt the altruism based on the traffic situation and possible benefits of altruism.

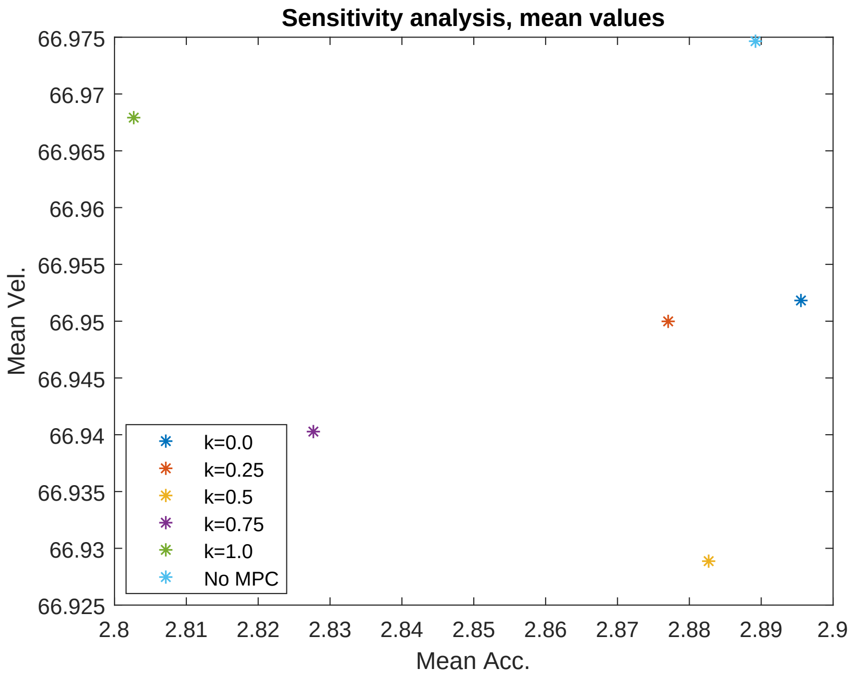


Figure 3.19: Sensitivity analysis for differing altruism levels

4
Conclusion

The altruistic control strategy makes sense in theory, and in simulated environments there is a benefit of acting altruistic. These benefits are not very distinctive in the scenarios showcased, however there is a clear improvement when there is a lane choice to be made, which was shown in scenario 2. The altruistic controller is the best performing controller overall, and it does beat the aggressive W99 driver model, meaning that there is some improvements that will be made by replacing a HDVs with CAVs in a traffic network. What needs to be researched further is how an increased number of CAVs may improve the traffic flow further. From the simulation results, it is also quite clear that the controller may need more time for tuning and a wider range of parameters have to be investigated. The implementation of the controller is another important aspect, as this will ensure that the controller will be feasible and may be applied in reality. By using cloud computing, the optimization times were significantly reduced, which indicates that if the controller was implemented using more efficient code and better computational hardware, real-time performance can be achieved. Another conclusion that can be made is the large model mismatch between OVRV and W99, and this is clearly shown in the difference between simulations of the two. It's clear that if a more realistic model is used, the gap will be less and better predictions could be made. However, still the MPC controller performs better than the W99 model when being altruistic, and it performs on the same level when being selfish.

Bibliography

[1] Yuki Sugiyama, Minoru Fukui, Macoto Kikuchi, Katsuya Hasebe, Akihiro Nakayama, Katsuhiro Nishinari, Shin-ichi Tadaki, and Satoshi Yukawa. Traffic jams without bottlenecks—experimental evidence for the physical mechanism of the formation of a jam. *New journal of physics*, 10(3):033001, 2008.

[2] M. A. S. Kamal, J. Imura, T. Hayakawa, A. Ohata, and K. Aihara. Smart driving of a vehicle using model predictive control for improving traffic flow. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):878–888, April 2014.

[3] M. A. S. Kamal, S. Taguchi, and T. Yoshimura. Efficient driving on multilane roads under a connected vehicle environment. *IEEE Transactions on Intelligent Transportation Systems*, 17(9):2541–2551, Sep. 2016.

[4] N. Wang, X. Wang, P. Palacharla, and T. Ikeuchi. Cooperative autonomous driving for traffic congestion avoidance through vehicle-to-vehicle communications. In *2017 IEEE Vehicular Networking Conference (VNC)*, pages 327–330, Nov 2017.

[5] *PTV Vissim website.* `http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/`. Accessed: 2020-01-25.

[6] Y. Zhou, M. E. Cholette, A. Bhaskar, and E. Chung. Optimal vehicle trajectory planning with control constraints and recursive implementation for automated on-ramp merging. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–12, 2018.

[7] James Blake Rawlings and David Q Mayne. *Model predictive control: Theory and design.* Nob Hill Pub. Madison, Wisconsin, 2009.

[8] A. Tapani J. Janson Olstam. Comparison of car-following models. 2004.

[9] Yu Gao. *Calibration and comparison of the VISSIM and INTEGRATION microscopic traffic simulation models.* PhD thesis, Virginia Tech, 2008.

Appendix A

This appendix includes the submitted journal paper "Pro-social Control of Connected Automated Vehicles in Mixed-Autonomy Multi-Lane Highway Traffic" which is a direct result of this master thesis work. The journal paper has been submitted to the IEEE Transactions on Intelligent Transportation Systems (T-ITS) special issue "Deployment of Connected and Automated Vehicles in Mixed Traffic Environment and the Implications on Traffic Safety and Efficiency".

Pro-social Control of Connected Automated Vehicles in Mixed-Autonomy Multi-Lane Highway Traffic

Jacob Larsson, Musa Furkan Keskin, *Member, IEEE*, Bile Peng, *Member, IEEE*, Balázs Kulcsár, Henk Wymeersch, *Senior Member, IEEE*

***Abstract*—We propose pro-social control strategies for connected automated vehicles (CAVs) to mitigate jamming waves in mixed-autonomy multi-lane traffic, resulting from car-following dynamics of human-driven vehicles (HDVs). Different from existing studies, which focus mostly on ego vehicle objectives to control CAVs in an individualistic manner, we devise a pro-social control algorithm. The latter takes into account the objectives (i.e., driving comfort and traffic efficiency) of both the ego vehicle and surrounding HDVs to improve smoothness of the entire observable traffic. Under a model predictive control (MPC) framework that uses acceleration and lane change sequences of CAVs as optimization variables, the problem of individualistic, altruistic, and pro-social control is formulated as a non-convex mixed-integer nonlinear program (MINLP) and relaxed to a convex quadratic program via penalty based reformulation of the optimal velocity with relative velocity (OVRV) car-following model. Low-fidelity simulations using the OVRV model and high-fidelity simulations using PTV Vissim simulator show that pro-social and altruistic control can provide significant performance gains over individualistic driving in terms of efficiency and comfort on both single- and multi-lane roads.**

***Index Terms*—Altruistic control, pro-social control, traffic disturbance, model predictive control, connected automated vehicles, stop-and-go waves.**

I. Introduction

In urban transportation systems, traffic jams pose a significant threat to vehicle safety, exhaust gas emission, fuel economy and passenger comfort, especially in dense traffic scenarios with stop-and-go waves. Disturbances such as accidents, lane restrictions or random braking may propagate backwards through the traffic as a result of the car-following dynamics of human-driven vehicles (HDVs), which leads to *moving traffic jams* [1]–[3]. To reduce such disturbance propagation, connected automated vehicles (CAVs) can be applied to control traffic flow and smooth out stop-and-go waves [2]–[5]. To circumvent jamming waves on single-lane roads, predictive control of CAV acceleration has recently been a popular strategy [3], [4], [6], while, for multi-lane highways, high-level lane changing decisions can be incorporated as additional degrees of freedom that can be optimized jointly with low-level acceleration inputs [7].

This work has partially been supported and funded by the Transport Area of Advance. *(Corresponding author: Balázs Kulcsár.)*

J. Larsson, M. F. Keskin, B. Kulcsár, and H Wymeersch are with the Department of Electrical Engineering, Chalmers University of Technology, 412 96 Gothenburg, Sweden (e-mail: {furkan,kulcsar,henkw}@chalmers.se).

Bile Peng is with Institute for Communications Technology, TU Braunschweig, 38106 Braunschweig, Germany (e-mail: peng@ifn.ing.tu-bs.de)

Historically in the literature, the emphasis of autonomous driving has been focused on CAVs' own (selfish) driving objectives [5], [7], [8], while ignoring the traffic-smoothing properties of CAVs [3]. In [5], a model predictive control (MPC) framework is proposed, where efficiency, comfort and safety of the CAV is improved by optimizing the acceleration and lane changes in a multi-lane traffic scenario. However, the focus of [5] is solely on the driving objectives of the CAV and not the surroundings, making such a driving strategy *selfish*. Another MPC-based approach is presented in [8], where a mixed-integer quadratic programming problem is set-up to optimize longitudinal velocity and lane-change maneuvers of the CAV. An entirely different selfish control strategy using reinforcement learning (RL) is presented in [7], where a multi-agent (multi-vehicle) RL algorithm is trained to achieve coordination between multiple CAVs in a highway scenario.

Altruistic agents have been considered in a variety of fields, including linear quadratic Gaussian (LQG) control [9], traffic route management [10], [11], microscopic traffic control [12], water resource planning via Markov decision processes (MDP) [13] and uncertain dynamic games [14]. In [10], a macroscopic routing perspective is presented to compare the total driving times of vehicles in a network obtained by a *selfish* user equilibrium (UE) model and an *altruistic* social optimum (SO) model. Similarly, the work in [11] provides a game-theoretic analysis of altruistic autonomy from a vehicle routing perspective and investigates its effect on traffic latency under varying degrees of altruism of CAVs. Regarding microscopic control, a cooperative altruistic driving strategy is developed in [12], where traffic jamming on highways is resolved by coordinating a group of CAVs using vehicle-to-vehicle (V2V) communications.

In light of the existing literature on traffic control, implementing purely individualistic (selfish) and/or purely altruistic behaviour have partially been investigated before. While a vehicle that behaves selfish/individualistic allocates control input to actions to reward only itself, pure altruistic behaviour allocates input in a way to reward only others (and disregard its own rewards or benefits). These forms of expected CAV behaviours can often be conflicting, even though they may be beneficial in different traffic scenarios. Therefore, one possible way to overcome the behaviour dilemma (selfish or altruistic?) is to use both at the same time. As shown in Fig. 1, the key insight of the altruistic controller is that performing altruistic lane change maneuvers help to dissolve jamming waves while improving comfort and efficiency. In [15], [16], socially compliant central coordination algorithms are suggested to solve a traffic coordination problem. In particular, these studies propose a large variety of Social Value Orientation (SVO) algorithms via the definition of two independent reward metrics: reward to self and reward to others. In case of intersection crossing, in [15], pro-social (combination of selfish and altruistic) behaviour provides most of the benefits in terms of wait time reduction. [17] proposes a decentralized intersection coordination mechanism using principles similar to [15]: topological braids capture selfish-altruistic modes. In the latter, however, elimination of unsafe trajectories is the goal. SVOs are mapped via proper weighting strategies, emphasizing the relative importance of altruistic or selfish objectives. The aim with the above state-of-the-art methods to influence/control/coordinate automated vehicles is the same: adapt their behaviour in mixed traffic conditions. As in mixed-autonomy traffic scenarios, both HDVs and CAVs have to co-exist and CAVs may be coordinated following social values known for human drivers.

In this paper, extending our preliminary work in [18] with comprehensive low- and high-fidelity simulation results, we propose in a comparative environment *altruistic* and *pro-social* control/coordination strategies where CAVs mitigate traffic jams by optimizing the driving objectives of the *overall traffic* as well as of their own *selfish* trajectories. We define the optimization problem that empowers CAVs with social behaviours as a model based, finite horizon, multi-objective optimization problem. Therefore, the contribution of this paper is twofold:

- We develop solutions to incentivize the pro-social coordination of mixed autonomy vehicles with model based predictive optimization algorithms. To that end, we use proper weighting strategies to map SVOs into vehicle control solutions (traffic efficiency and comfort). The key idea is to emphasize the relative importance of altruistic versus selfish objectives and reach pro-social behavior.
- We evaluate the proposed methods on both low- and high-fidelity traffic simulators, which helps us quantify the benefits/drawbacks of SVOs in terms of fuel economy, ride comfort, trajectory alignment, etc.

More precisely, in this work, we propose an MPC-based selfish, altruistic, and pro-social coordination algorithm, where CAVs in mixed traffic scenarios model HDVs with the Optimal Velocity with Relative Velocity (OVRV) car following model [19]. Then, a finite horizon time prediction window is created in which CAVs can select their speed and lanes to minimize the multi-objective cost function. The result is a non-convex mixed-integer nonlinear program, which is relaxed to a simpler convex quadratic program via penalty based reformulation of the OVRV model. Simulations with low fidelity simulations by the OVRV model, and with high fidelity simulations by PTV Vissim microscopic traffic simulator are carried out to investigate the impact of different social behaviour triggered by CAVs.

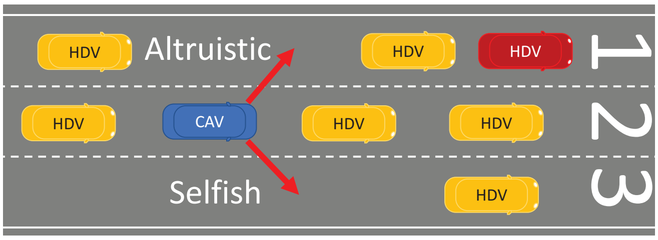


Fig. 1. Exemplary multi-lane highway scenario with CAVs (blue) and HDVs (yellow) and decelerating HDVs (red). An altruistic strategy would involve turning left in to lane 1 and try to mitigate the the traffic jamming caused by the red HDV and help improve overall traffic smoothness. A selfish decision is to take a right on to lane 3 and avoid the jamming, improving smoothness for the CAV but reducing the overall traffic smoothness and jamming.

II. System Model

In Fig. 1, a mixed autonomy multi-highway traffic scenario is depicted, with both CAVs and HDVs. It is assumed that the CAVs obtain position and speed information from surrounding HDVs via vehicle-to-vehicle (V2V) communications [20], [21]. Although there may be any number of CAVs in the traffic, the focus here lies on individual automated driving, i.e., the information at each CAV is self-contained and no additional data is obtained from the other CAVs. In the scenario, the objective of individual CAVs is to obtain optimal control input sequences in terms of vehicle acceleration input and lane change decisions. The CAV applies an altruistic MPC controller and thus the optimization objective is to maximize the entire traffic objectives in terms of comfort, efficiency and indirectly, emissions.

A. Vehicle States

The state vector of ith CAV at discrete time k, with sampling time Δt, is defined as

$$\mathbf{x}_{i,k}^{\text{CAV}} = \left[p_{i,k}^{\text{CAV}} \; v_{i,k}^{\text{CAV}} \; y_{i,k}^{\text{CAV}}\right]^T \tag{1}$$

for $i = 1, \ldots, N_{\text{cav}}$, where $p_{i,k}^{\text{CAV}} \in \mathbb{R}$ and $v_{i,k}^{\text{CAV}} \in \mathbb{R}$ are, respectively, the longitudinal position and velocity of the vehicle, and $y_{i,k}^{\text{CAV}} \in \mathcal{L} \triangleq \{1, 2, \ldots, N_{\text{lane}}\}$ represents the lane number of the vehicle. Similarly, the state vector of the jth HDV at time k is expressed as

$$\mathbf{x}_{j,k}^{\text{HDV}} = \left[p_{j,k}^{\text{HDV}} \; v_{j,k}^{\text{HDV}} \; y_{j,k}^{\text{HDV}}\right]^T \tag{2}$$

for $j = 1, \ldots, N_{\text{hdv}}$, where $p_{j,k}^{\text{HDV}}, v_{j,k}^{\text{HDV}} \in \mathbb{R}$ and $y_{j,k}^{\text{HDV}} \in \mathcal{L}$.

B. CAV Control Inputs

The control input vector of the ith CAV at time k is given by

$$\mathbf{u}_{i,k}^{\text{CAV}} = \left[a_{i,k}^{\text{CAV}} \; \delta_{i,k}^{\text{CAV}}\right]^T \tag{3}$$

where $a_{i,k}^{\text{CAV}} \in \mathbb{R}$ is the longitudinal acceleration and $\delta_{i,k}^{\text{CAV}}$ represents the lateral movement, i.e., the lane change decision, defined as

$$\delta_{i,k}^{\text{CAV}} \in \mathcal{L}_{\Delta} \triangleq \{-1, 0, 1\} \; . \tag{4}$$

Here, 0 denotes a lane-keeping decision and $1/{-1}$ represents a left/right lane change decision. To reduce modelling complexity in controller design step (prediction), it is assumed that the lane change is instantaneous and is thus completed in a single time step [22]. Note, more complex and dynamic lane change maneuvers can be added if needed. Finally, in the numerical case study, we use Vissim that has a continuous and dynamic lane change model.

C. Car-Following Behavior of HDVs

The longitudinal dynamics of HDVs is described using a car-following model as in [23], [24]

$$a_{j,k}^{\mathrm{HDV}} = f(h_{j,k}^{\mathrm{HDV}}, v_{j,k}^{\mathrm{HDV}}, \Delta v_{j,k}^{\mathrm{HDV}}) \tag{5}$$

where $a_{j,k}^{\mathrm{HDV}} \in \mathbb{R}$ is the longitudinal acceleration of the jth HDV at time k, $h_{j,k}^{\mathrm{HDV}}$ is the headway and $\Delta v_{j,k}^{\mathrm{HDV}}$ the velocity difference between the jth HDV and the preceding vehicle, written as

$$h_{j,k}^{\mathrm{HDV}} = p_{j,k}^{\mathrm{HDV,pre}} - p_{j,k}^{\mathrm{HDV}} \ , \tag{6}$$

$$\Delta v_{j,k}^{\mathrm{HDV}} = v_{j,k}^{\mathrm{HDV,pre}} - v_{j,k}^{\mathrm{HDV}} \ , \tag{7}$$

with $p_{j,k}^{\mathrm{HDV,pre}}$ and $v_{j,k}^{\mathrm{HDV,pre}}$ representing the position and speed of the vehicle preceding the jth HDV at time k on the same lane. The car-following dynamics are represented by the Optimal Velocity with Relative Velocity (OVRV) model [19]

$$f(h, v, \Delta v) = \alpha(V(h) - v) + \beta \Delta v \tag{8}$$

where the velocity function $V(h)$ is a piecewise-linear function of headway h (driver perceived optimal and headway based velocity), defined as [25]

$$V(h) = \left[\widetilde{V}(h)\right]_0^{v_{\max}}, \widetilde{V}(h) = v_{\max}\frac{h - h_{\min}}{h_{\max} - h_{\min}} \ , \tag{9}$$

with $[v]_0^{v_{\max}} \triangleq \max(0, \min(v_{\max}, v))$, and α, β, $h_{\min}$, $h_{\max}$ and $v_{\max}$ are driver-dependent model parameters. Furthermore, it is also assumed that HDVs keep the same lane over the entire MPC prediction horizon [5], i.e., $\delta_{j,k+n}^{\mathrm{HDV}} = 0$ for $n = 0, 1, \ldots, N_{\mathrm{p}} - 1$, where N_{p} denotes the prediction horizon. In Fig. 1, the described car-following behaviour is shared among the yellow HDVs. However, the red HDV causing disturbance on lane 1 is considered a leading HDV on that lane. These HDVs are handled differently, where we implement MPC prediction heuristics in the form of constant acceleration over the horizon.

D. Discrete-Time Vehicle Dynamics

The dynamics of the ith CAV can be expressed as

$$\mathbf{x}_{i,k+1}^{\mathrm{CAV}} = \mathbf{A}\mathbf{x}_{i,k}^{\mathrm{CAV}} + \mathbf{B}\mathbf{u}_{i,k}^{\mathrm{CAV}} \tag{10}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} \Delta t^2/2 & 0 \\ \Delta t & 0 \\ 0 & 1 \end{bmatrix} . \tag{11}$$

In a similar fashion, the dynamics of the jth HDV can be written as

$$\mathbf{x}_{j,k+1}^{\mathrm{HDV}} = \mathbf{A}\mathbf{x}_{j,k}^{\mathrm{HDV}} + \mathbf{B}\mathbf{u}_{j,k}^{\mathrm{HDV}} \tag{12}$$

where the input is defined as

$$\mathbf{u}_{j,k}^{\mathrm{HDV}} = \left[a_{j,k}^{\mathrm{HDV}} \ \delta_{j,k}^{\mathrm{HDV}}\right]^T . \tag{13}$$

III. MPC Formulation for Individual Altruistic Driving

This section covers the problem of how altruism can be reached with carefully selecting CAV control input. The constraints for inputs and states are provided for a multi-lane traffic scenario, and then the optimal CAV control problem is formulated in the MPC framework.

A. Constraints

For the optimal CAV control problem, the following constraints are imposed on the vehicle inputs and states.

1) Acceleration Bounds: The following constraints bound the longitudinal acceleration, i.e.,

$$a_{\min} \leq a_{i,k+n}^{\mathrm{CAV}} \leq a_{\max} \ , \ n = 0, 1, \ldots, N_{\mathrm{p}} - 1 \tag{14}$$

where N_{p} is the horizon length.

2) Lateral Safety Constraints: At the nth prediction step, a lane change occurs when $\left|\delta_{i,k+n}^{\mathrm{CAV}}\right| = 1$. Here, the ith CAV should keep the headway h_{safe} to the closest vehicle in the new lane, i.e.,

$$p_{i,k+n}^{\mathrm{CAV,bg}} - p_{i,k+n}^{\mathrm{CAV}} \geq h_{\mathrm{safe}}, \ p_{i,k+n}^{\mathrm{CAV}} - p_{i,k+n}^{\mathrm{CAV,sm}} \geq h_{\mathrm{safe}} \tag{15}$$

where $p_{i,k+n}^{\mathrm{CAV,bg}}$ and $p_{i,k+n}^{\mathrm{CAV,sm}}$ are the longitudinal positions of the vehicles on the new lane that are closest to the ith CAV at time $k+n$ with $p_{i,k+n}^{\mathrm{CAV,bg}} \geq p_{i,k+n}^{\mathrm{CAV}} \geq p_{i,k+n}^{\mathrm{CAV,sm}}$. Furthermore, we also limit the number of lane changes over the horizon to at most 1,

$$\sum_{n=1}^{N_p-1} \left|\delta_{i,k+n}^{\mathrm{CAV}}\right| \leq 1 \tag{16}$$

3) Longitudinal Safety Constraints: In order to avoid collisions and keep a safe minimum headway to preceding vehicles on the lane, CAVs and HDVs are constrained to a dynamic headway [4]. Hence, for the ith CAV, we have

$$p_{i,k+n}^{\mathrm{CAV,pre}} - p_{i,k+n}^{\mathrm{CAV}} \geq h_{\min} + t_{\min} v_{i,k+n}^{\mathrm{CAV}} \ , \ n = 1, \ldots, N_{\mathrm{p}} \tag{17}$$

where $p_{i,k+n}^{\mathrm{CAV,pre}}$ is the position of the vehicle preceding the ith CAV at time $k+n$ and $t_{\min}$ denotes the minimum time headway. Similar constraints also bound all HDVs during the prediction in order to augment the OVRV model behaviour with a dynamic and realistic headway behaviour.

B. Objectives

For the optimal CAV control problem, we apply two categories of objectives, namely, traffic efficiency and driving comfort. In addition, we split the driving comfort into acceleration and jerk components.

1) Traffic Efficiency: Traffic efficiency can be defined as the objective of maintaining a desired velocity V^* for the ith CAV and for the overall traffic including observable HDVs and other CAVs:

$$\begin{aligned} &\mathcal{J}_i^{\mathrm{eff}}(\mathbf{u}_{i,k:k+N_{\mathrm{p}}-1}^{\mathrm{CAV}}) \\ &= \sum_{n=0}^{N_{\mathrm{p}}-1} \left[(v_{i,k+n}^{\mathrm{CAV}} - V^*)^2 + \kappa \sum_{j \in \mathcal{G}_{i,k}^{\mathrm{HDV}}} (v_{j,k+n}^{\mathrm{HDV}} - V^*)^2 \right] \end{aligned} \tag{18}$$

where $\mathcal{G}_{i,k}^{\mathrm{HDV}}$ is the set of indices of those HDVs succeeding the ith CAV at time k and that are also observable by it[1], and κ is a constant variable that indicates the *level of altruism*, i.e., a weight that controls the CAV's prioritization between its own selfish driving objectives and the surrounding traffics objectives[2].

2) Driving Comfort - Acceleration Magnitude: This driving objective aims to reduce the discomfort associated with large magnitudes of acceleration:

$$\begin{aligned}&\mathcal{J}_i^{\mathrm{mag}}(\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}})\\&=\sum_{n=0}^{N_\mathrm{p}-1}\left[\left(a_{i,k+n}^{\mathrm{CAV}}\right)^2+\kappa\sum_{j\in\mathcal{G}_{i,k}^{\mathrm{HDV}}}\left(a_{j,k+n}^{\mathrm{HDV}}\right)^2\right]\end{aligned}\tag{19}$$

where the dependency of HDV accelerations on CAV control inputs is through (5)–(7).

3) Driving Comfort - Jerk: Another component of discomfort associated with acceleration is the jerky behaviour due to rapid changes in the acceleration derivative:

$$\begin{aligned}\mathcal{J}_i^{\mathrm{jerk}}(\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}})=\sum_{n=0}^{N_\mathrm{p}-1}&\left[\left(\frac{a_{i,k+n+1}^{\mathrm{CAV}}-a_{i,k+n}^{\mathrm{CAV}}}{\Delta t}\right)^2\right.\\&\left.+\kappa\sum_{j\in\mathcal{G}_{i,k}^{\mathrm{HDV}}}\left(\frac{a_{j,k+n+1}^{\mathrm{HDV}}-a_{j,k+n}^{\mathrm{HDV}}}{\Delta t}\right)^2\right]\end{aligned}\tag{20}$$

which is an approximation of the functions derivative obtained by Euler's method.

4) Driving Comfort - Total Objective: We define the total driving comfort as a two-component objective, where the individual parts above are weighted against each other. The total driving comfort objective for the ith CAV at time k is defined as

$$\begin{aligned}&\mathcal{J}_i^{\mathrm{comf}}(\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}})\\&=\mathcal{J}_i^{\mathrm{mag}}(\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}})+w_2\,\mathcal{J}_i^{\mathrm{jerk}}(\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}})\end{aligned}\tag{21}$$

where w_2 is a weight to indicate the importance of the jerk component on to the overall comfort objective.

5) Total Objective Function: For the ith CAV, the total objective function at time k with prediction horizon length N_p is defined as

$$\begin{aligned}&\mathcal{J}_i^{\mathrm{tot}}(\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}})\\&=\mathcal{J}_i^{\mathrm{eff}}(\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}})+w_1\,\mathcal{J}_i^{\mathrm{comf}}(\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}})\end{aligned}\tag{22}$$

where w_1 is a weight that balances the impact between efficiency and comfort objectives. We note that safety is taken into consideration as hard physical constraints through (16) and (17) in the MPC formulation.

[1]The CAV is only able to control succeeding vehicles, i.e., *Lagrangian control scheme* [3]), utilizing state information from both preceding and succeeding vehicles.

[2]We note that both CAV and HDV velocities in (18) depend on CAV control inputs $\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}}$ through (5)–(7) and (10). From the car-following behavior in (5)–(7), HDV acceleration is a function of the position and speed of the preceding vehicle, which implies that the effect of CAV control actions can be propagated downstream towards HDVs moving on the same lane and affect the traffic efficiency in (18).

C. MPC Prediction Heuristics

Over the MPC prediction horizon, we assume a constant acceleration heuristic [26]–[29] for predicting the leading HDV trajectory[3]. At time k, we have

$$a_{j,k+n}^{\mathrm{HDV}}=\widehat{a}_{j,k}^{\mathrm{HDV}},\ j\in\mathcal{F}_{i,k}^{\mathrm{HDV}}\tag{23}$$

for $n=0,1,\ldots,N_\mathrm{p}-1$, where $\mathcal{F}_{i,k}^{\mathrm{HDV}}$ is the set of indices for leading HDVs that are observed by the ith CAV, and $\widehat{a}_{j,k}^{\mathrm{HDV}}$ is the measured acceleration of the jth leading HDV at time k. To ensure the non-negativity of the velocity over the prediction horizon in (23), $a_{j,k+n}^{\mathrm{HDV}}$ is set to zero for $n>\widetilde{n}$ if $v_{j,k+\widetilde{n}}^{\mathrm{HDV}}<0$. In other words, we prioritize constraining the velocity over (23) in the prediction horizon.

D. Problem Formulation

Given the initial internal states of the ith CAV at time k and the initial states of HDVs observed by the ith CAV at time k, the MPC problem over a prediction horizon of length N_p can be formulated as follows:

$$\begin{aligned}\underset{\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}}}{\text{minimize}}\quad&\mathcal{J}_i^{\mathrm{tot}}(\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}})\\\text{subject to}\quad&\text{(Prediction of Leading HDVs) (23)}\\&\text{(HDV Car-Following Model) (5)–(9)}\\&\text{(CAV-HDV Dynamics) (10)–(13)}\\&\text{(Vehicle/Traffic Constraints) (14)–(17)}\ .\end{aligned}\tag{24}$$

The formulation in (24) is a mixed-integer non-linear programming problem (MINLP). The non-linearity (and non-convexity) stem from the the car-following dynamics of the HDVs in (8), where the range policy (9) is a piecewise-linear function. The integer variables come from the lane change sequence variables $\delta_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}}$.

IV. Optimization Strategies for Altruistic Driving

In this section, we propose an optimization strategy for handling the MINLP in (24). This is achieved by firstly reformulating the non-linear car following dynamics in (8) as linear constraints. To handle the integer lane change variables in $\delta_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}}$, we decompose the optimization problem into lower-level subproblems on each reachable lane. The solutions of these problems can then be combined to form a final lane-change decision.

A. Transformation of Piecewise-Linear Car-Following Constraints

To avoid the intractability of the piecewise-linear car-following dynamics in (5)–(9), we take a penalty based approach for constraint reformulation. Precisely, we transform the piecewise-linear equality constraint in (5) to a linear equality constraint, two linear inequality constraints that bound

[3]Accelerations of leading HDVs on each lane cannot be determined using a car-following function as in (5). Thus, we assume the accelerations are available through on-board tracking filters on CAVs and therefore the accelerations can be used to predict leading HDVs trajectories.

V

the function from above and below, along with a penalty term in the objective function. The car-following dynamics constraint in (5) with the OVRV model in (8), given by

$$a^{\mathrm{HDV}}_{j,k+n} = \alpha(V(h^{\mathrm{HDV}}_{j,k+n}) - v^{\mathrm{HDV}}_{j,k+n}) + \beta \Delta v^{\mathrm{HDV}}_{j,k+n} \tag{25}$$

for $n = 0, 1, \ldots, N_{\mathrm{p}} - 1$ and $j \in \mathcal{H}^{\mathrm{HDV}}_{i,k}$, can be rewritten by introducing slack variables $\boldsymbol{\gamma}_j = \begin{bmatrix}\gamma_{j,N_{\mathrm{p}}-1} \ldots \gamma_{j,0}\end{bmatrix}^T$ as follows[4]. We begin by introducing a penalty term in the objective function (24) using the ℓ_2-norm as

$$\mathcal{J}^{\mathrm{tot}}_i(\mathbf{u}^{\mathrm{CAV}}_{i,k:k+N_{\mathrm{p}}-1}) + \lambda\, \mathcal{J}^{\mathrm{slack}} \tag{26}$$

where

$$\mathcal{J}^{\mathrm{slack}} = \sum_{j \in \mathcal{H}^{\mathrm{HDV}}_{i,k}} \left\| \boldsymbol{\gamma}_j \right\|^2 . \tag{27}$$

and $\boldsymbol{\gamma}_j = \begin{bmatrix}\gamma_{j,0} \ldots \gamma_{j,N_{\mathrm{p}}-1}\end{bmatrix}^T$. Here, λ is a weight that controls the tightness of car-following dynamics in (25) and $\mathcal{J}^{\mathrm{slack}}$ enforces the driving models' slack variable to be minimized. Secondly, we reformulate (25) by introducing one equality constraint and two inequality constraints that bound the function from above and below, as

$$a^{\mathrm{HDV}}_{j,k+n} = \alpha(\widetilde{V}(h^{\mathrm{HDV}}_{j,k+n}) - v^{\mathrm{HDV}}_{j,k+n}) + \beta \Delta v^{\mathrm{HDV}}_{j,k+n} + \gamma_{j,n} \tag{28}$$

$$a^{\mathrm{HDV}}_{j,k+n} \leq \alpha(\widetilde{V}(h_{\max}) - v^{\mathrm{HDV}}_{j,k+n}) + \beta \Delta v^{\mathrm{HDV}}_{j,k+n} \tag{29}$$

$$a^{\mathrm{HDV}}_{j,k+n} \geq \alpha(\widetilde{V}(h_{\min}) - v^{\mathrm{HDV}}_{j,k+n}) + \beta \Delta v^{\mathrm{HDV}}_{j,k+n} \tag{30}$$

for $n = 0, 1, \ldots, N_{\mathrm{p}} - 1$ and $j \in \mathcal{H}^{\mathrm{HDV}}_{i,k}$.

B. Optimization Subproblem for Fixed Lane Change Decision

In order to handle the integer lane change variables in (24), we create three separate subproblems for each reachable lane for the ith CAV, where each subproblem corresponds to a fixed lane changing decision $\delta^{\mathrm{CAV}}_{i,k} \in \mathcal{L}_\Delta$. This means that each lane change decision is optimized only for the initial time of the MPC control problem, while the acceleration control inputs are still obtained for N_{p} steps forward in time on each lane[5]. With this reformulation in (26)–(30), the MPC optimization subproblem for each reachable lane of (24) for a given lane change decision can be written as:

$$\begin{aligned} \underset{a^{\mathrm{CAV}}_{i,k:k+N_{\mathrm{p}}-1}, \{\boldsymbol{\gamma}_j\}}{\text{minimize}} \quad & \mathcal{J}^{\mathrm{tot}}_i(\mathbf{u}^{\mathrm{CAV}}_{i,k:k+N_{\mathrm{p}}-1}) + \lambda\, \mathcal{J}^{\mathrm{slack}} \\ \text{subject to} \quad & \text{(Prediction of Leading HDVs) (23)} \\ & \text{(HDV Car-Following Model) (28)–(30)} \\ & \text{(CAV-HDV Dynamics) (10)–(13)} \\ & \text{(Vehicle/Traffic Constraints) (14)–(17)} . \end{aligned} \tag{31}$$

Note that (31) is a convex optimization problem, with convex quadratic objective function and linear constraints. Thus, the optimization problem can be solved efficiently using interior-point methods [30]. The solutions of (31) for $\delta^{\mathrm{CAV}}_{i,k} \in \mathcal{L}_\Delta$ can be merged to obtain the optimal lane change decision and its corresponding acceleration sequence $a^{\mathrm{CAV}}_{i,k:k+N_{\mathrm{p}}-1}$. This is achieved by choosing the solution with the lowest cost $\mathcal{J}^{\mathrm{tot}}_i(\mathbf{u}^{\mathrm{CAV}}_{i,k:k+N_{\mathrm{p}}-1})$.

[4] $\mathcal{H}^{\mathrm{HDV}}_{i,k}$ is a set of indices for HDVs observed by the ith CAV, except for the leading HDVs on each lane.

[5] Intuitively, the algorithm tends to select the lane with the highest probability of congestion in the horizon.

V. Experiments

We perform experiments at two levels of simulation fidelity. First, we carry out low-fidelity simulations where the simpler OVRV model in (8) is used to simulate the car-following dynamics of HDVs in evaluating the performance of the proposed MPC controller in (31) on a single lane scenario. Then, the high-fidelity microscopic multi-modal traffic simulator PTV Vissim is deployed to verify the controller in realistic single- and multi-lane settings. Whilst with the OVRV model we validate the proof of concept in simplified traffic scenarios, with the multi-lane scenario Vissim tools, we partially test the nominal controller in uncertain (car following model mismatch) and more complex and realistic traffic environments.

A. Simulation Setup and Parameters

We consider three simulation scenarios with 2 different setups. Firstly, the OVRV and Vissim single lane scenarios use a setup where there is a leading HDV driving with a sinusoidal acceleration profile. Following this leading vehicle there is directly a CAV that tries to mitigate these disturbances. HDVs driving with the OVRV model or the Vissim W99 model are then following this CAV, and these HDVs are being controlled by the CAV.

For the third scenario, which is multi-lane Vissim, there is a second setup. The road consists of three lanes and the first row of vehicles contain HDVs following a sinusoidal acceleration profile on all three lanes. Following these leading HDVs, there is the second row of another three HDVs, behaving according to the OVRV or W99 driving model, on all three lanes. Succeeding these HDVs is a single CAV in the middle lane. After the CAV comes several rows of HDVs depending on the penetration rate. For all setups, the headway distance for all vehicles is 40 meters.

Both the low-fidelity OVRV and high-fidelity Vissim simulations have several parameters that determine the behaviour of the MPC controller and HDV trajectories. In the appendix, we briefly cover the parameters pertaining to the driver model and the MPC optimization, used for the different simulation scenarios. See Appendix A for specific optimization parameters and Appendix B for specific simulation parameters.

B. Evaluation Metrics

For assessing the performance of the controller and the impact of the different levels of altruism (parameter κ), we evaluate three areas: overall traffic acceleration and velocity directly impacted by the objective functions along with vehicle emissions, which is an indirect byproduct of acceleration and velocity via the vehicle emission model VT-Micro.

1) Vehicle Emission Model: Since vehicle emissions are not directly optimized in the MPC controller, we use a vehicle emission model to calculate the emissions of carbon monoxide (CO), hydrocarbons (HC), nitrogen oxide (NOx) and fuel consumption (FC). We adopt the VT-Micro model [31] to calculate these metrics from the acceleration and velocity data gathered from simulation. In addition, it is assumed that the

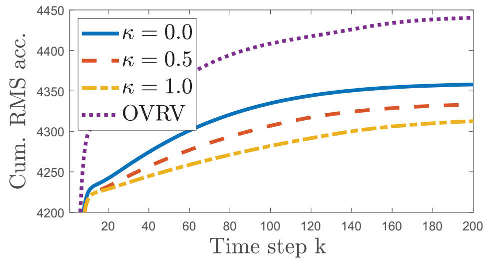
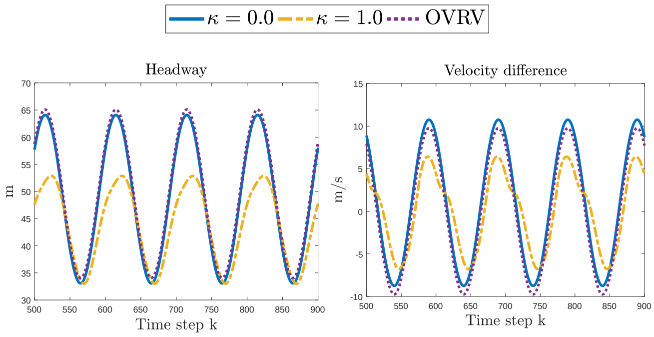


Fig. 2. Single lane OVRV simulation results of cumulative RMS acceleration with varying altruism parameter κ.



(a) (b)

Fig. 3. Headway and velocity difference between the HDV following the CAV for each controller

traffic consists solely of gasoline passenger vehicles. Under this setting, emissions can be calculated as

$$\mathcal{J}_k^y(\mathbf{v}_{i,k}, \mathbf{a}_{i,k}) = \exp(\mathbf{v}_{i,k}^T \mathbf{P}_y \mathbf{a}_{i,k}) \tag{32}$$

where $\mathcal{J}_k^y(\mathbf{v}_{i,k}, \mathbf{a}_{i,k})$ is the prediction of the variable $y \in \{\mathrm{CO}, \mathrm{HC}, \mathrm{NOx}, \mathrm{FC}\}$ at every simulation step k of the i-th vehicle (CAV or HDV), $\mathbf{P}_y \in \mathbb{R}^{4\times 4}$ is a parameter matrix for each variable y,

$$\mathbf{v}_{i,k} = \begin{bmatrix} 1 & v_{i,k}^\star & (v_{i,k}^\star)^2 & (v_{i,k}^\star)^3 \end{bmatrix}^T \tag{33}$$

$$\mathbf{a}_{i,k} = \begin{bmatrix} 1 & a_{i,k}^\star & (a_{i,k}^\star)^2 & (a_{i,k}^\star)^3 \end{bmatrix}^T \tag{34}$$

are the velocity and acceleration vectors for the i-th vehicle at time k, respectively, and $\star \in \{\mathrm{CAV}, \mathrm{HDV}\}$. Emission and fuel consumption rates are given in the units kg/s and l/s, respectively.

C. Results

In this section, we present the experimental results for the OVRV model and the Vissim W99 model in single- and multi-lane scenarios.

1) Single Lane - OVRV: To evaluate the performance of the proposed altruistic control strategy in a single-lane road, Fig. 2 showcases the cumulative RMS acceleration obtained via the OVRV model and the proposed approach with different values of the altruism parameter $\kappa = [0, 0.5, 1]$. It is observed that cumulative acceleration decreases with an increasing degree of altruism. The selfish controller improves upon the OVRV driving model by reducing cumulative accelerations by 3.4% while the altrustic controller with $\kappa = 1$ further improves on the results with an additional 2.1% compared to the selfish case with $\kappa = 0$. This indicates the potential of altruistic control to mitigate traffic disturbances and improve driving comfort and safety.

As an illustration of how stop-and-go waves are dampened via the proposed altruistic strategy, Fig. 3a shows with respect to time the headway of the HDV that follows the CAV. Comparing the OVRV driving model with the selfish case, $\kappa = 0$, there is only a minuscule smoothing effect that is barely noticeable. However, when the altruistic controller is applied, with $\kappa = 1$, the smoothing effect is more pronounced, i.e., headway fluctuations are significantly reduced, which proves the effectiveness of the altruistic strategy against the disturbances caused by the leading HDV. In Fig. 3b, we investigate the velocity fluctuations, i.e., the velocity difference between the HDV that follows the CAV and the CAV itself. It is seen that the amplitude of fluctuations decreases for the altruistic case in comparison to the OVRV driving model and the selfish controller, which again demonstrates the smoothing effect provided by the altruistic CAV. These results evidence that an altruistic CAV can reduce the oscillations stemming from the leading vehicle, leading to more stable velocities (i.e., efficiency) and lower accelerations (i.e., comfort) experienced by the vehicles.

Fig. 4a–4d illustrate the benefits of altruism on the cumulative vehicle emissions, where altruism reduces the total emissions of all metrics. Similarly, Fig. 4e–4h demonstrate the emission rate as distributions for the entire simulation, where the pure altruistic driving strategy exhibits a lower mean value and variance in comparison to other altruism levels.

2) Single Lane - W99: In the previous subsection, no car following model uncertainties have been considered, i.e. the MPC used the OVRV to create predictions. In PTV Vissim, the car following model W99 aims at capturing significantly more complex driver behaviour than the OVRV. We therefore tested the MPC algorithm (based on OVRV predictions) in a single lane context by emulating the real environment with W99. First, the model mismatch between the OVRV and W99 has been appropriately handled by the altuistic controller, showing clear signs of robustness. We can also report some degradation of the performance (compared to the nominal case in the previous subsection). Fig. 5a-5b show the benefits of the controller in general, as the mean acceleration is reduced and the mean velocity is increased for all altruism levels in comparison to the coordination control free driving model. Furthermore, the proposed MPC controller indicates lower mean and variance leading to consistency in the driving and reducing oscillations.

Fig. 6a–6d depict the cumulative emissions, which suggests that the controller can provide significant reductions in emissions in comparison to W99. However, there is no noticeable difference between the altruism levels. For emission distributions in Fig. 6e–6h, the pure altruistic driving strategy exhibits a slight reduction in the variance in comparison to the other altruism levels and the W99 model. Mean values for all altruism are similar and all altruism levels show lower mean

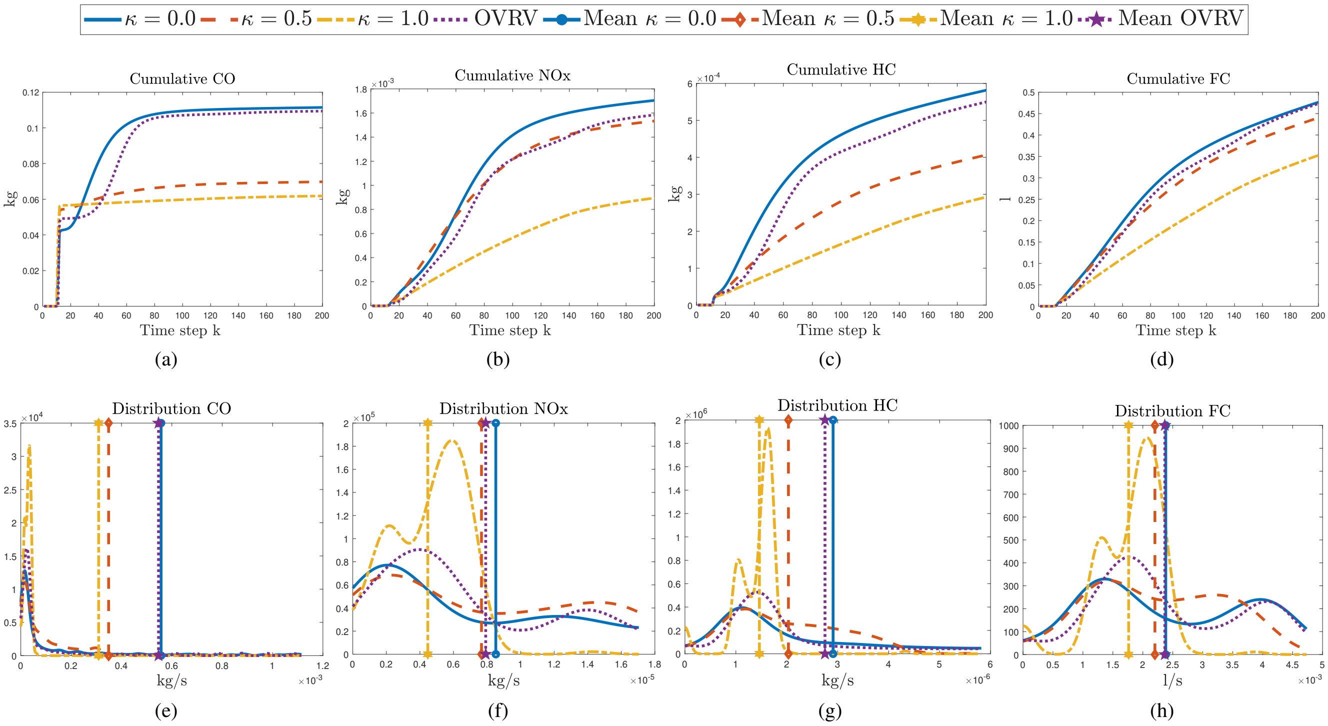


Fig. 4. Single lane OVRV simulation results of vehicle emissions with varying altruism parameter κ.

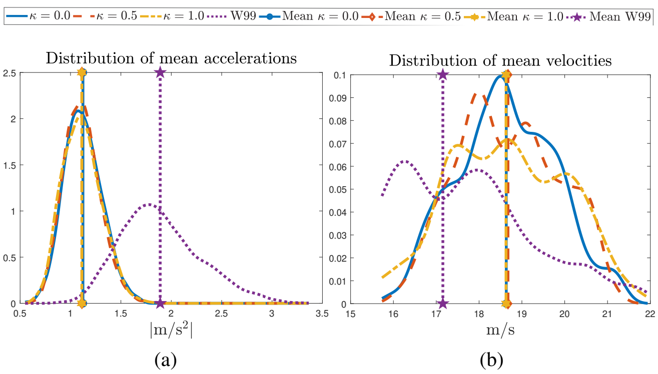


Fig. 5. Single lane Vissim simulation results of mean acceleration and velocity distribution with varying altruism parameter κ.

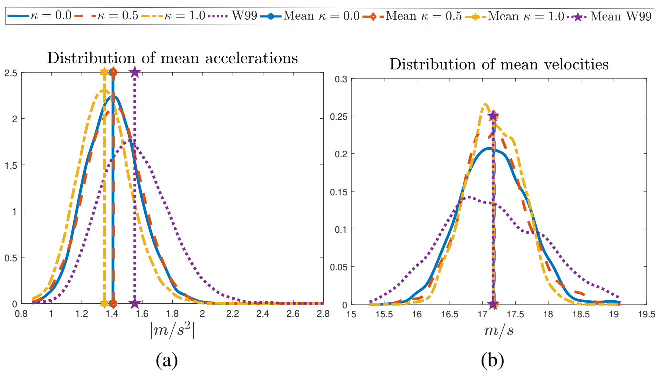


Fig. 7. Multi lane Vissim simulation results of mean acceleration and velocity distribution with varying altruism parameter κ.

values in comparison to uncontrolled W99 models.

3) Multi-Lane - W99: For multi-lane simulations, the benefits of altruism are more pronounced compared to single-lane scenarios. These simulations involve the optimization of longitudinal acceleration along with lane changing decisions, meaning that the full control strategy is utilized in this scenario. Furthermore, in this simulation scenario, we are running 3 CAVs in a decentralized control strategy. In other words, CAVs do not coordinate their actions, and they view the other CAVs as regular HDVs.

VIII

In Fig. 7a, we observe that the mean accelerations are reduced by adopting the pure altruistic driving strategy, while the controller in general outperforms the W99 driving model in both mean values and variance. The variance between the different altruism levels is similar. For the velocity distribution in Fig. 7b, all altruism levels and the W99 model perform similarly in terms of mean values; however, the variance differs. Pure altruism exhibits the lowest variance, while the other altruism levels provide slightly higher variance. The W99 model performs the worst when it comes to variance, indicating that the traffic efficiency is negatively impacted.

To investigate vehicle emissions in multi-lane Vissim scenarios, Fig. 8a–8d illustrate the cumulative emissions with respect to time. As seen from the figures, the altruistic driving strategy significantly outperforms the W99 driving model, with the performance gap increasing with higher levels of altruism. In addition, Fig. 8e–8h show the histograms of the

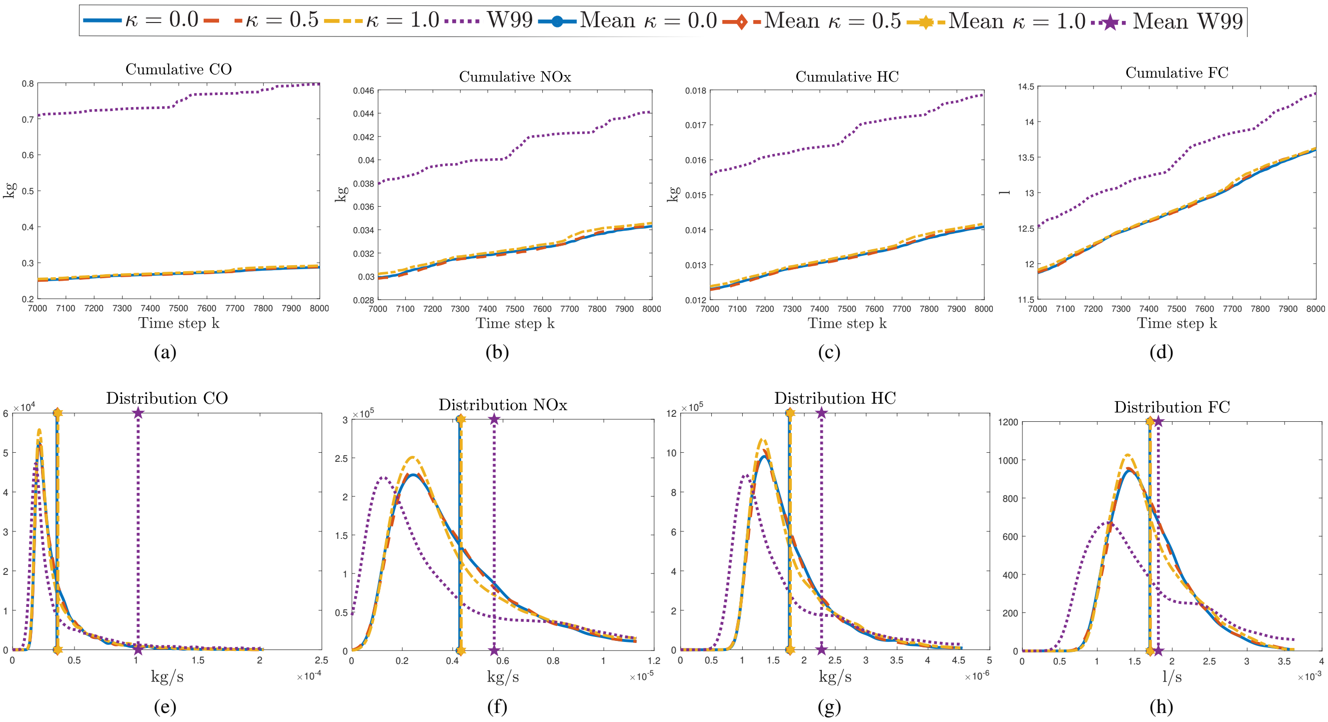


Fig. 6. Single lane Vissim simulation results of vehicle emissions with varying altruism parameter κ.

fuel consumption metrics, which exhibit similar trends, i.e., the pure altruistic driving strategy achieves lower mean emission values than the other controllers.

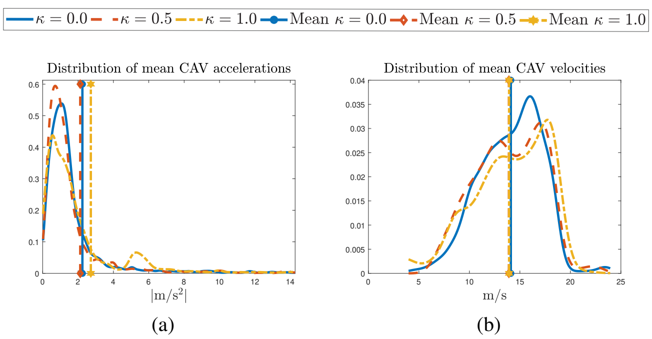


Fig. 9. Multi lane Vissim simulation results of mean acceleration and velocity distribution with varying altruism parameter κ for the CAVs.

To explore the CAV-specific results, Fig. 9 shows the distribution of acceleration and velocities for only the CAVs in the simulation. As expected, the highest mean CAV acceleration is obtained in the case of pure altruism (i.e., $\kappa = 1$) since an altruistic CAV sacrifices its own driving objectives for the sake of overall traffic smoothness. The performance difference between the altruistic and selfish controllers in terms of mean acceleration is 20%, while the mean velocities are very close to each other. In addition, the variance in velocity is slightly lower for the altruistic case and the distribution is skewed slightly towards higher velocities as well, giving an indication that the traffic efficiency for HDVs is improved at almost no cost of the efficiency of the CAVs.

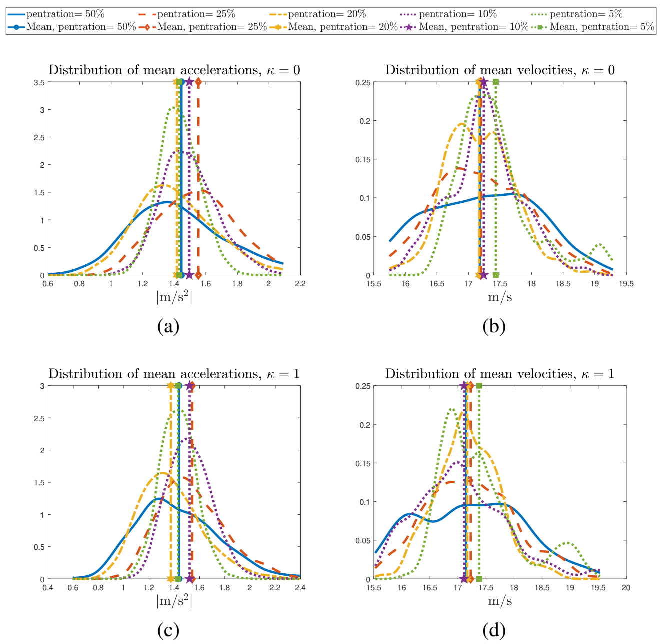


Fig. 10. Multi lane Vissim simulation results of mean acceleration and velocity distribution with varying CAV penetration rate from 10% to 50%.

Fig. 10 showcases the selfish controller, $\kappa = 0$, altruistic controller $\kappa = 1$ and the penetration rate of CAVs, i.e. the ratio between HDVs and CAVs. For penetration rates between 50% to 25%, the performance is lacking due to the larger mismatch between driving models, which results from decentralized control strategy and that the CAVs assume all vehicles it sees are HDVs. The penetration rate of 20% is the best performing one in terms of mean acceleration, and the penetration rates

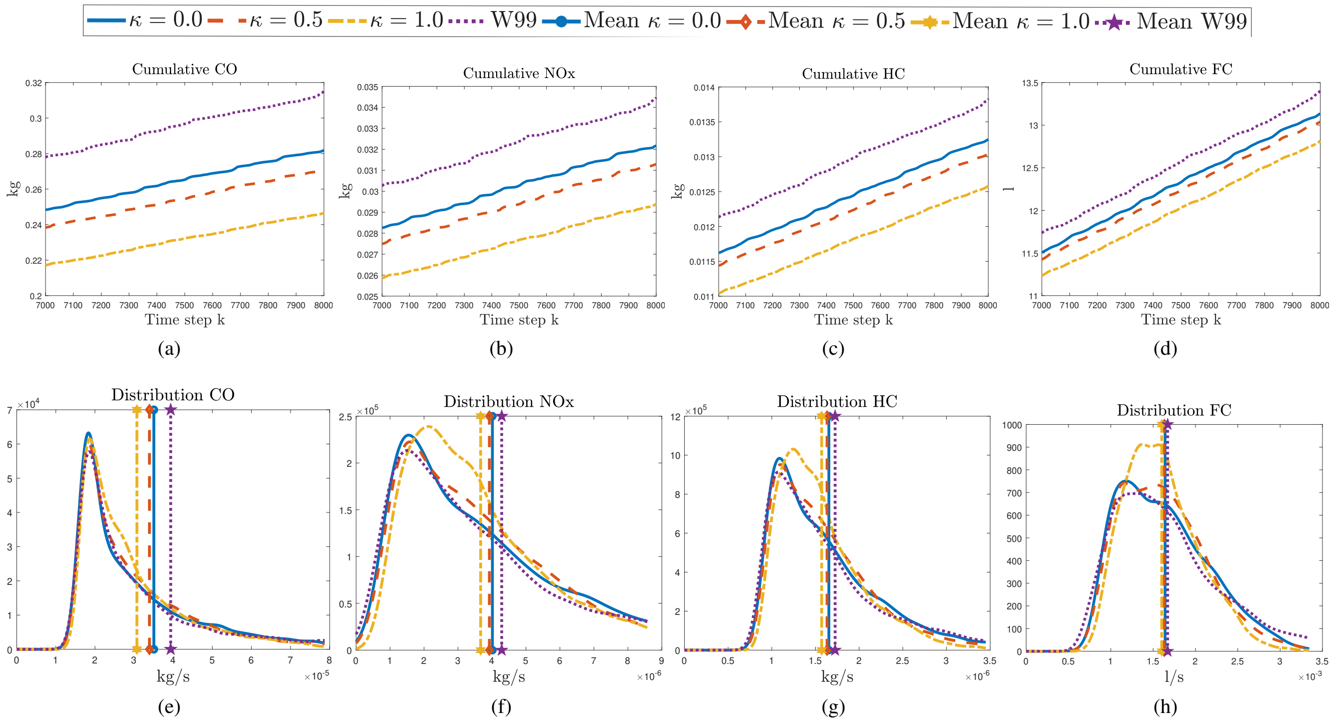


Fig. 8. Multi lane Vissim simulation results of vehicle emissions with varying altruism parameter κ.

of 10% and 5% do have lower variance than the rest. This pattern is also present in the velocity distributions, where lower penetration rates show a lower variance meanwhile the mean values for all penetration rates are similar, with a slight increase for the 5% case. This indicates that even a low penetration rate will provide the benefits of altruism to the traffic.

It is worth emphasizing the importance of the model used in predictive control, which influences the performance reached by the altruistic controller. We therefore envisage the benefits of (i) using more sophisticated car-following models in the MPC design or (ii) robustifying the nominal MPC control algorithm, which may help to reach better closed loop performance.

VI. Conclusion

We proposed rolling horizon, model based control methodologies to coordinate connected automated vehicles (CAV) in multi-lane highway traffic conditions. A set of realistic traffic scenarios are defined where automated and human driven vehicles (HDV) have to co-exist. The objective function of the control algorithms for CAVs is formulated in a such a way that selfish and altruistic goals can be addressed separately or simultaneously. In this way, pro-social human behaviour can be replicated via the control of CAVs. Simulation results in cumulative and empirical distribution metrics showed that traffic efficiency and comfort metrics could be significantly improved by applying an altruistic driving strategy. We pointed out that pro-social behaviour could be triggered with changing the relative weights of the overall cost function, i.e., changing the weightings between the two objectives, namely, the objective function to self and objective function to others. Future research directions may involve changing the proposed objective functions with the number of vehicles, i.e., traffic density dependent weightings (e.g., no altruism is needed in free-flow conditions). In addition to the decentralized control strategies proposed in this paper, more complex MPC controllers utilizing centralized control of CAVs on highways, or cooperative CAV driving, can be designed to incorporate inter-CAV communication for coordinating CAV driving and improved controllability of the traffic flow.

Appendix A
MPC Optimization Parameters

The optimization parameters involve the constant terms in the objective function and constraints in (31) used in the simulation sequences. Since we are evaluating the altruistic parameter, all parameters except κ remain static. The static optimization parameters are set as follows: $\alpha = 2$, $\beta = 2$, $\mathrm{h_{max}} = 70$ [m], $\mathrm{h_{min}} = 10$ [m], $\mathrm{a_{max}} = 5$ [m/s^2], $\mathrm{a_{min}} = -5$ [m/s^2], $\mathrm{v_{max}} = 30.5$ [m/s], $\mathrm{t_{min}} = 0.25$ [s], $\mathrm{w_1} = 0.75$, $\mathrm{w_2} = 0.5$, $\lambda = 0.99$ and $\mathrm{N_p} = 40$. For numerical stability in optimization, we perform scaling of the objective function in (31), as described in the following.

A. *Scaling the Objective Function in* (31)

While not critical for the purposes of formulating the MPC controller, there are some practical issues to consider when optimizing the objective function. In order to avoid the practical issues related to differing scales of the numerical components of the objective function, we normalize each objective in (18), (19) and (20) and the slack objective in (26) to lie within the

X

interval of $[0,\ 1]$ by dividing each component by its maximum value. Additionally, we also modify the weights $w_1,\ w_2,\ \kappa$ and λ to lie within the same interval. Furthermore, we adopt a decentralized control strategy, i.e., each optimization concerns only a single CAV. However, there may be multiple CAVs on the road, in which case the optimized CAV treats the other CAVs as regular HDVs in predictive optimization.

We modify the efficiency objective in (18) by

$$\mathcal{J}_i^{\mathrm{eff,scaled}}(\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}}) = \sum_{n=0}^{N_\mathrm{p}-1}\Bigg[\kappa\Bigg(\frac{v_{k+n,-}^{\mathrm{CAV}}V^*}{v_{\max}}\Bigg)^2 + (1-\kappa)\sum_{j\in\mathcal{G}_{i,k}^{\mathrm{HDV}}}\Bigg(\frac{v_{j,k+n}^{\mathrm{HDV}}-V^*}{v_{\max}}\Bigg)^2\Bigg] \tag{35}$$

where $\{\kappa \in \mathbb{R} \mid 0 \le \kappa \le 1\}$. Similarly, the acceleration magnitude objective (19) and jerk objective (20) now becomes

$$\mathcal{J}_i^{\mathrm{mag,scaled}}(\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}}) = \sum_{n=0}^{N_\mathrm{p}-1}\Bigg[\kappa\Bigg(\frac{a_{i,k+n}^{\mathrm{CAV}}}{a_{\max}}\Bigg)^2 + (1-\kappa)\sum_{j\in\mathcal{G}_{i,k}^{\mathrm{HDV}}}\Bigg(\frac{a_{j,k+n}^{\mathrm{HDV}}}{a_{\max}}\Bigg)^2\Bigg] \tag{36}$$

$$\mathcal{J}_i^{\mathrm{jerk,scaled}}(\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}}) = \sum_{n=0}^{N_\mathrm{p}-1}\Bigg[\kappa\Bigg(\frac{a_{i,k+n+1}^{\mathrm{CAV}}-a_{i,k+n}^{\mathrm{CAV}}}{a_{\max}\Delta t}\Bigg)^2 + (1-\kappa)\sum_{j\in\mathcal{G}_{i,k}^{\mathrm{HDV}}}\Bigg(\frac{a_{j,k+n+1}^{\mathrm{HDV}}-a_{j,k+n}^{\mathrm{HDV}}}{a_{\max}\Delta t}\Bigg)^2\Bigg] \tag{37}$$

The objective function concerning the slack variable is also modified as

$$\mathcal{J}^{\mathrm{slack,scaled}} = \sum_{j\in\mathcal{H}_{i,k}^{\mathrm{HDV}}}\left\|\frac{\boldsymbol{\gamma}_j}{\max_j \boldsymbol{\gamma}_j}\right\|^2 \tag{38}$$

The scaled version of the objective function in (31) is then given by

$$\begin{aligned}&\mathcal{J}_i^{\mathrm{tot,scaled}}(\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}}) \\ &= (1-\lambda)\Bigg[(1-w_1)\mathcal{J}_i^{\mathrm{eff,scaled}}(\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}}) \\ &+ w_1\Bigg((1-w_2)\mathcal{J}_i^{\mathrm{mag,scaled}}(\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}}) \\ &+ w_2\mathcal{J}_i^{\mathrm{jerk,scaled}}(\mathbf{u}_{i,k:k+N_\mathrm{p}-1}^{\mathrm{CAV}})\Bigg)\Bigg] \\ &+ \lambda\mathcal{J}^{\mathrm{slack,scaled}}\ .\end{aligned} \tag{39}$$

Appendix B
Simulation Parameters

While the OVRV simulation results use the same parameters as the driving model in the MPC formulation, the Vissim Wiedemann99 (W99) model does not. The W99 model has several parameters: CC0 is the desired vehicle standstill distance, CC1 is the headway time (in seconds) that the vehicle wants to keep. CC0 and CC1 define the safe vehicle headway by $\mathrm{h_{safe}} = \mathrm{CC0}+\mathrm{CC1}\dot{\mathrm{v}}$ where v $[\mathrm{m/s}]$ is the vehicle velocity. CC2 controls the following variation in meters by defining the oscillation boundaries by $\mathrm{h_{safe}} \ge \mathrm{h} \ge \mathrm{h_{safe}} + \mathrm{CC2}$. CC3 is a threshold parameter defining when the vehicle recognized a slower vehicle ahead of it, starts to slow down and entering a following state. CC4 and CC5 controls the speed differences during the follwing state for deceleration and acceleration respectively where smaller values increases the vehicles sensitivity to accelerations and deceleration of the preceding vehicle. CC6 controls the speed dependency of oscillations, where larger values lead to larger velocity with increasing distance while the vehicle is in the following state. CC7, CC8 and CC9 controls the acceleration during oscillation, at standstill and at 80 $\mathrm{km/h}$ respectively.

In the Vissim simulations, the following parameters for W99 [32] are used, CC0$=1$ [m], CC1$=0.9$ [s], CC2$=1$ [m], CC3$=-8$, CC4$=-0.05$, CC5$=0.05$, CC6$=1$, CC7$=10$ [m/s^2], CC8$=10$ [m/s^2], CC9$=10$ [m/s^2]. Furthermore, the look ahead distance is set to 150 m with 2 vehicles observed at most and the look back distance is set to 100 m. For single lane scenarios, there are 5 HDVs behind the CAV. For multi lane scenarios, there are 15 HDVs behind the CAVs with 3 CAVs in total.

References

[1] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S.-i. Tadaki, and S. Yukawa, "Traffic jams without bottlenecks—experimental evidence for the physical mechanism of the formation of a jam," *New journal of physics*, vol. 10, no. 3, p. 033001, 2008.

[2] M. Wang, W. Daamen, S. P. Hoogendoorn, and B. van Arem, "Cooperative car-following control: Distributed algorithm and impact on moving jam features," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1459–1471, May 2016.

[3] R. E. Stern, S. Cui, M. L. D. Monache, R. Bhadani, M. Bunting, M. Churchill, N. Hamilton, R. Haulcy, H. Pohlmann, F. Wu, B. Piccoli, B. Seibold, J. Sprinkle, and D. B. Work, "Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments," *Transportation Research Part C: Emerging Technologies*, vol. 89, pp. 205 – 221, 2018.

[4] M. A. S. Kamal, J. Imura, T. Hayakawa, A. Ohata, and K. Aihara, "Smart driving of a vehicle using model predictive control for improving traffic flow," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 878–888, April 2014.

[5] M. A. S. Kamal, S. Taguchi, and T. Yoshimura, "Efficient driving on multilane roads under a connected vehicle environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, pp. 2541–2551, Sep. 2016.

[6] R. A. Dollar and A. Vahidi, "Efficient and collision-free anticipative cruise control in randomly mixed strings," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 4, pp. 439–452, Dec 2018.

[7] C. Yu, X. Wang, X. Xu, M. Zhang, H. Ge, J. Ren, L. Sun, B. Chen, and G. Tan, "Distributed multiagent coordinated learning for autonomous driving in highways based on dynamic coordination graphs," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2019.

[8] M. Bahram, "Interactive maneuver prediction and planning for highly automated driving functions," Ph.D. dissertation, Technische Universität München, 2017.

[9] M. Huang, P. E. Caines, and R. P. Malhamé, "Social dynamics in mean field LQG control: Egoistic and altruistic agents," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 3140–3145.

[10] N. Levy and E. Ben-Elia, "Emergence of system optimum: A fair and altruistic agent-based route-choice model," *Procedia computer science*, vol. 83, pp. 928–933, 2016.

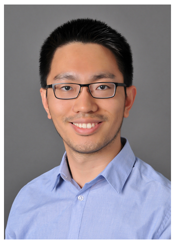
[11] E. Bıyık, D. Lazar, R. Pedarsani, and D. Sadigh, "Altruistic autonomy: Beating congestion on shared roads," *arXiv e-prints*, p. arXiv:1810.11978, Oct 2018.

[12] N. Wang, X. Wang, P. Palacharla, and T. Ikeuchi, "Cooperative autonomous driving for traffic congestion avoidance through vehicle-to-vehicle communications," in *2017 IEEE Vehicular Networking Conference (VNC)*, Nov 2017, pp. 327–330.

[13] G. Desquesnes, G. Lozenguez, A. Doniec, and E. Duviella, "Distributed MDP for water resources planning and management in inland waterways," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 6576–6581, 2017.
[14] E. Camponogara, H. Zhou, and S. Talukdar, "Altruistic agents in uncertain dynamic games," *Journal of Computer and Systems Sciences International*, vol. 45, no. 4, pp. 536–552, 2006.
[15] N. Buckman, A. Pierson, W. Schwarting, S. Karaman, and D. Rus, "Sharing is caring: Socially-compliant autonomous intersection negotiation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 6136–6143.
[16] A. Pierson, W. Schwarting, S. Karaman, and D. Rus, "Weighted buffered voronoi cells for distributed semi-cooperative behavior," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 5611–5617.
[17] S. S. S. Christoforos Mavrogiannis, Jonathan A. DeCastro, "Implicit multiagent coordinationat unsignalized intersections via multimodalinference enabled by topological braids," *arXiv:2004.05205*, 2020.
[18] F. Keskin, B. Peng, B. Kulcsár, and H. Wymeersch, "Altruistic control of connected automated vehicles in mixed-autonomy multi-lane highway traffic," in *2020 IFAC World Congress*, 2020.
[19] R. E. Wilson and J. A. Ward, "Car-following models: fifty years of linear stability analysis–a mathematical perspective," *Transportation Planning and Technology*, vol. 34, no. 1, pp. 3–18, 2011.
[20] L. Zhang and G. Orosz, "Beyond-line-of-sight identification by using vehicle-to-vehicle communication," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 1962–1972, 2018.
[21] S. S. Avedisov, G. Bansal, and G. Orosz, "Impacts of connected automated vehicles on freeway traffic patterns at different penetration levels," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2020.
[22] N. Kheterpal, K. Parvate, C. Wu, A. Kreidieh, E. Vinitsky, and A. Bayen, "Flow: Deep reinforcement learning for control in sumo," 2018.
[23] G. Orosz, R. E. Wilson, and G. Stépán, "Traffic jams: dynamics and control," 2010.
[24] G. Orosz, R. E. Wilson, R. Szalai, and G. Stépán, "Exciting traffic jams: nonlinear phenomena behind traffic jam formation on highways," *Physical review E*, vol. 80, no. 4, p. 046205, 2009.
[25] L. Zhang and G. Orosz, "Motif-based design for connected vehicle systems in presence of heterogeneous connectivity structures and time delays," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1638–1651, June 2016.
[26] S. H. Hamdar, M. Treiber, H. S. Mahmassani, and A. Kesting, "Modeling driver behavior as sequential risk-taking task," *Transportation research record*, vol. 2088, no. 1, pp. 208–217, 2008.
[27] A. Kesting, M. Treiber, and D. Helbing, "Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4585–4605, 2010.
[28] M. Treiber and A. Kesting, "Modeling human aspects of driving behavior," in *Traffic flow dynamics*. Springer, 2013, pp. 205–224.
[29] Y. Yu, R. Jiang, and X. Qu, "A modified full velocity difference model with acceleration and deceleration confinement: calibrations, validations, and scenario analyses," *IEEE Intelligent Transportation Systems Magazine*, 2019.
[30] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004.
[31] S. Zegeye, B. De Schutter, J. Hellendoorn, E. Breunesse, and A. Hegyi, "Integrated macroscopic traffic flow, emission, and fuel consumption model for control purposes," *Transportation Research Part C*, vol. 31, pp. 158–171, Jun. 2013.
[32] Y. Gao, "Calibration and comparison of the VISSIM and INTEGRATION microscopic traffic simulation models," Ph.D. dissertation, Virginia Tech, 2008.

Jacob Larsson recieved the B.S degree in Automation and Mechatronics, the M.Sc in Product Development and the the M.Sc in Systems, Control and Mechatronics from Chalmers University of Technology, Gothenburg, Sweden, in 2016, 2018 and 2020. He is currently a research engineer at Husqvarna Construction Products Primary R&D, Jonsered, Sweden, within mechatronics and control systems. His main research interests is optimal control systems and modelling.

Musa Furkan Keskin is a researcher and a Marie Skłodowska-Curie Fellow (MSCA-IF) in the department of Electrical Engineering at Chalmers University of Technology, Gothenburg, Sweden. He obtained the B.S., M.S., and Ph.D degrees from the Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey, in 2010, 2012, and 2018, respectively. He received the 2019 IEEE Turkey Best Ph.D Thesis Award for his thesis on visible light positioning systems. His project "OTFS-RADCOM: A New Waveform for Joint Radar and Communications Beyond 5G" is granted by the European Commission through the H2020-MSCA-IF-2019 call. His current research interests include intelligent transportation systems, joint radar-communications, and positioning in 5G and beyond 5G systems.



Bile Peng received the B.S. degree from Tongji University, Shanghai, China, in 2009, the M.S. degree from the Technische Universität Braunschweig, Germany, in 2012, and the Ph.D. degree with distinction from the Institut für Nachrichtentechnik, Technische Universität Braunschweig in 2018. He has been a Postdoctoral researcher in the Chalmers University of Technology, Sweden from 2018 to 2019, a development engineer at IAV GmbH, Germany from 2019 to 2020. Currently, he is a Postdoctoral researcher in the Technische Universität Braunschweig, Germany. His research interests include wireless channel modeling and estimation, Bayesian inference as well as machine learning algorithms, in particular deep reinforcement learning, for resource allocation of wireless communication.

Dr. Peng is a major contributor to the IEEE Standard for High Data Rate Wireless Multi-Media Networks Amendment 2: 100 Gb/s Wireless Switched Point-to-Point Physical Layer (IEEE Std 802.15.3d-2017) and received the IEEE vehicular technology society 2019 Neal Shepherd memorial best propagation paper award.

Balázs Kulcsár received the M.Sc. degree in traffic engineering and the Ph.D. degree from Budapest University of Technology and Economics (BUTE), Budapest, Hungary, in 1999 and 2006, respectively. He has been a Researcher/Post-Doctor with the Department of Control for Transportation and Vehicle Systems, BUTE, the Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN, USA, and with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands. He is currently a Professor with the Department of Electrical Engineering, Chalmers University of Technology, Göteborg, Sweden. His main research interest focuses on traffic flow modeling and control.

Henk Wymeersch (S'01, M'05, SM'19) obtained the Ph.D. degree in Electrical Engineering/Applied Sciences in 2005 from Ghent University, Belgium. He is currently a Professor of Communication Systems with the Department of Electrical Engineering at Chalmers University of Technology, Sweden. He is also a Distinguished Research Associate with Eindhoven University of Technology. Prior to joining Chalmers, he was a postdoctoral researcher from 2005 until 2009 with the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. Prof. Wymeersch served as Associate Editor for IEEE Communication Letters (2009-2013), IEEE Transactions on Wireless Communications (since 2013), and IEEE Transactions on Communications (2016-2018). During 2019-2021, he is a IEEE Distinguished Lecturer with the Vehicular Technology Society. His current research interests include the convergence of communication and sensing, in a 5G and Beyond 5G context.