CHALMERS
UNIVERSITY OF TECHNOLOGY


# Safe and Efficient Vehicle Motion Control of Articulated Heavy Vehicles

## A motion control method using Model Predictive Control and Machine Learning

Master's thesis in Systems, Control and Mechatronics

LUKAS WISELL
SANDER VAN DAM

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

Master's thesis 2024

# Safe and Efficient Vehicle Motion Control of Articulated Heavy Vehicles

A motion control method using Model Predictive Control and Machine Learning

LUKAS WISELL
SANDER VAN DAM


CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
*Division of Systems and Control*
Chalmers University of Technology
Gothenburg, Sweden 2024

Safe and Efficient Vehicle Motion Control of Articulated Heavy Vehicles
A motion control method using Model Predictive Control and Machine Learning
LUKAS WISELL
SANDER VAN DAM



Supervisors:
Umur Erdinc, Volvo Group Trucks Technology
Maliheh Sadeghi Kati, Volvo Group Trucks Technology
Esteban Gelso, Volvo Group Trucks Technology

Examiner:
Nikolce Murgovski, Electrical Engineering

Master's Thesis 2024
Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Volvo FH Electric tractor with a semitrailer driving on a snow-covered road.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2024

Safe and Efficient Vehicle Motion Control of Articulated Heavy Vehicles
A motion control method using Model Predictive Control and Machine Learning
LUKAS WISELL
SANDER VAN DAM
Department of Electrical Engineering
Chalmers University of Technology

# Abstract

When braking an articulated heavy vehicle with at least one electric unit, it is best to prioritize electric braking to enhance power regeneration for the battery electric vehicle. However, this approach combined with treacherous road conditions such as low-friction high curvature roads, may lead to unsafe yaw instability modes such as jack-knifing, trailer swing, and combinational spin-out. These unsafe modes can lead to potentially fatal accidents. Therefore, achieving a safe control allocation during manoeuvring is crucial to ensure stability while maximizing the regenerated energy.

This thesis describes a method for control allocation of longitudinal forces and steering wheel input to a specified vehicle combination for vehicle motion control. The system was implemented using a non-linear Model Predictive Control (MPC) scheme, which allows the use of constraints to maintain the vehicle within safe operating conditions. To ensure a persistently feasible non-linear model predictive controller that improves vehicle safety, an approximation of the control invariant set was computed. The control invariant set is approximated using an Active Learning algorithm that utilizes support vector machines to approximate the set effectively.

Results indicated the ability of the control scheme to manage the vehicle motion under various environmental conditions in near real-time. Energy regeneration could be improved using the MPC while keeping the vehicle combination in a safe mode. The derived method enabled flexible usage of the MPC depending on the vehicle parameters and objectives. Additional extensive analysis is necessary to more accurately approximate the largest possible control invariant set through the application of machine learning.

Keywords: Heavy trucks, Articulated Heavy Vehicle, electric truck, control allocation, model predictive control, machine learning, persistent feasibility, actuator coordination, path tracking.

# Acknowledgements

We want to thank our thesis supervisors at Volvo Group Trucks Technology, Umur Erdinc, Maliheh Sadeghi Kati and Esteban Gelso for their support, kind words and trust in us throughout the thesis project. Your words of encouragement and constructive feedback helped to uplift the thesis to its final form.

We would also like to thank our examiner Nikolce Murgovski for the continued support and supportive feedback during this thesis. Your expertise and valuable insights have helped us enormously.

We would also like to thank the following people for helping us finalise the project: All employees of the Vehicle Motion Management team at Volvo Group Trucks Technology for creating a welcoming atmosphere and sharing your experience. All other thesis workers in the VMM team for their company and discussions. Lastly, we would also like to thank friends and family for their feedback and support throughout this journey.

Lukas Wisell, Gothenburg, June 2024
Sander van Dam, Gothenburg, June 2024

# List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

| | |
|---|---|
| AHV | Articulated Heavy Vehicle |
| AL | Active Learning |
| BEV | Battery Electric Vehicle |
| CA | Control Allocation |
| CoG | Centre of gravity |
| FN | False Negative |
| FP | False Positive |
| IPOPT | Interior Point Optimizer |
| ML | Machine Learning |
| MPC | Model Predictive Control |
| OCP | Optimal Control Problem |
| SOC | State of charge |
| SOE | Safe Operating Envelope |
| SQP | Sequential Quadratic Programming |
| SVM | Support Vector Machine |
| TN | True Negative |
| TP | True Positive |
| VTM | Volvo Transport Model |

# Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

## Indices

| | |
|---|---|
| $i$ | Index corresponding to the unit counted from the front unit |
| $j$ | Index corresponding to the axle group counted from the front axle on the current unit |
| $k$ | Index corresponding to the time step of the MPC |
| $n$ | Index for various counters |

## Sets

| | |
|---|---|
| $\mathbb{X}$ | State space of MPC |
| $\mathbb{X}_f$ | Terminal constraint set |
| $\mathbb{U}$ | Input space of MPC |
| $\mathcal{X}_N$ | Feasible set |
| $\mathcal{C}$ | A control invariant set |
| $\mathcal{C}_\infty$ | Maximum control invariant set |
| $\Omega_k$ | Recursive precursor set |
| $\mathcal{S}$ | Set of unlabeled samples for AL |
| $\mathcal{S}_r$ | Random subset of $\mathcal{S}$ |
| $\mathcal{L}_0$ | Initial set of labeled samples for AL |
| $\mathcal{L}_n$ | Set of labeled samples for AL at iteration n |
| $\mathcal{L}_{s,n}$ | Set of newly labeled samples for AL at iteration n |
| $\mathbb{P}$ | Parameter space |
| $\mathbb{X}_{ext}$ | State space $\mathbb{X}$ and parameter space $\mathbb{P}$ combined |

| | |
|---|---|
| $\mathcal{F}$ | Feature space for SVM |

# Parameters

| | |
|---|---|
| $m_i$ | Mass of unit $i$ |
| $J_i$ | Moment of inertia of unit $i$ |
| $a_i$ | Length between the first axle and reference point for unit $i$ |
| $b_i$ | Length between the reference point and coupling point to next unit (if applicable) $i$ |
| $c_i$ | Length between the last axle and reference point for unit $i$ |
| $C_{ij}$ | Cornering stiffness at axle $j$, unit $i$ |
| $\mu$ | Road friction coefficient |
| $N$ | Prediction horizon length of MPC |
| $c_r$ | Curvature of the road |
| $\mathbf{Q}$ | Cost matrix for MPC states |
| $\mathbf{R}$ | Cost matrix for MPC inputs |
| $\mathbf{P}$ | Target cost, cost matrix for MPC states at final time step |
| $\alpha$ | Scaling factor for target cost |
| $J_{bx}$ | Matrix in Acados mapping states to constraints |
| $J_{bx}^N$ | Matrix in Acados mapping states in the last time step to terminal constraint |
| $J_{bu}$ | Matrix in Acados mapping inputs to constraints |
| $\underline{h}$ | Lower limit on non-linear constraint in Acados |
| $\overline{h}$ | Upper limit on non-linear constraint in Acados |
| $\underline{h}^e$ | Lower limit on non-linear constraint in Acados, last time step |
| $\overline{h^e}$ | Upper limit on non-linear constraint in Acados, last time step |
| $\mathbf{W}$ | Cost matrix for states and inputs in Acados |
| $\mathbf{W_N}$ | Cost matrix for states at last time step in Acados |
| $V_x$ | Matrix in Acados mapping cost elements to states |
| $V_x^N$ | Matrix in Acados mapping cost elements to states at last time step |
| $V_u$ | Matrix in Acados mapping cost elements to inputs |
| $N_L$ | Final number of labelled samples for AL |
| $B$ | Batch size for AL |
| $\zeta$ | Balancing index for $\mathcal{L}_0$ |

| | |
|---|---|
| $N_{feas}$ | Minimum number of feasible points in $\mathcal{L}_0$ |
| $\eta$ | Balancing index for randomness in the choice of samples to label |

# Variables

| | |
|---|---|
| $F_{xij}$ | Longitudinal force at axle $j$ on unit $i$ |
| $F_{yij}$ | Lateral force at axle $j$ on unit $i$ |
| $v_{xi}$ | Longitudinal speed of unit $i$ |
| $\dot{v}_{xi}$ | Longitudinal acceleration of unit $i$ |
| $v_{yi}$ | Lateral speed of unit $i$ |
| $\dot{v}_{yi}$ | Lateral acceleration of unit $i$ |
| $v_x$ | Longitudinal speed of unit 1 |
| $v_y$ | Lateral speed of unit 1 |
| $\dot{\phi}_1$ | Yaw rate of unit $i$ at the reference point |
| $\delta_{ij}$ | Steering angle of axle $j$ on unit $i$ |
| $F_{cxi}$ | Longitudinal coupling force on unit $i$ |
| $F_{cyi}$ | Lateral coupling force on unit $i$ |
| $\theta$ | Articulation angle |
| $s_{yij}$ | Lateral slip of the tyre at axle $j$ on unit $i$ |
| $v_{xij}$ | Longitudinal velocity of wheel at axle $j$ on unit $i$ |
| $v_{yij}$ | Lateral velocity of wheel at axle $j$ on unit $i$ |
| $F_{zij}$ | Vertical force at axle $j$ on unit $i$ |
| $F_{xtot}$ | Longitudinal force at coupling point |
| $F_{ytot}$ | Lateral force at coupling point |
| $X$ | Global position of the reference point of unit 1, $x$-coordinate |
| $Y$ | Global position of the reference point of unit 1, $y$-coordinate |
| $\phi_1$ | Heading angle of the truck |
| $x_k$ | MPC state vector at time step $k$ |
| $u_k$ | MPC input vector at time step $k$ |
| $p_k$ | MPC parameter vector at time step $k$ |
| $\theta_{ref}$ | Articulation angle reference |
| $\dot{v}_{xref}$ | Acceleration reference vector |
| $x_e$ | Deviation state vector |

| | |
|---|---|
| $v_{xe}$ | Longitudinal velocity error |
| $v_{y_e}$ | Lateral velocity error |
| $\dot{\phi}_{1e}$ | Yaw rate error |
| $\phi_{1e}$ | Heading error, angle between truck and road |
| $\dot{\theta}_e$ | Articulation angle rate error |
| $\theta_e$ | Articulation angle error |
| $d_e$ | Distance between reference point on tractor and road |
| $\delta_{11e}$ | Steering angle error |
| $\phi_{road}$ | Yaw angle of road |
| $C_n$ | Classifier in AL at iteration $n$ |
| $x_l$ | Label of state $x$ |
| $x_{\mathcal{L}}$ | State $x$ combined with label $x_l$ |
| $n_-$ | Number of infeasible points in $\mathcal{L}_0$ |
| $n_+$ | Number of feasible points in $\mathcal{L}_0$ |
| $y_n$ | Label of sample $x_n$ |
| $\mathbf{w}$ | Weight vector of support vectors |
| $b$ | Bias in SVM |
| $\alpha_n$ | Lagrange multiplier for each support vector |
| $F_{prop,ij}$ | Normalised vertical force on axle $j$ on unit $i$ |

# Contents

# List of Figures

# List of Tables

# 1

# Introduction

Trucks can be coupled with different types and numbers of trailers, leading to several combinations of motion actuators. These can include steerable axles, mechanical brakes, and auxiliary brakes in combination with available powertrain actuators. This setup exceeds the number of controlled motions, making the system over-actuated and allowing for multiple feasible solutions to achieve the desired motion. One way to solve the over-actuation is by using Control Allocation (CA) within the control system. The CA can also enable corrections in the motion of the vehicle combination when unsafe actions occur or even prevent them. Therefore, the CA is an option for coordination when the system has more input signals than virtual signals controlled and can be implemented using multiple different control strategies. Using the flexibility of the CA, energy-efficient choices can be made when allocating the control signals. For example, when braking a Battery Electric Vehicle (BEV), the CA can be used to maximize regeneration of electric power to the battery [1].

## 1.1 Background

When driving an Articulated Heavy Vehicle (AHV) such as a tractor with a semi-trailer, unsafe yaw instabilities can occur. Such instabilities can include jack-knifing, where the tractor loses grip relative to the trailer, or trailer swing, where the trailer loses grip relative to the tractor. When both units lose grip, this is referred to as a combinational spin-out. Additionally, off-tracking will be considered a yaw instability since the loss of grip on the front axle of the tractor will lead to the vehicle not being able to follow the defined path. See Fig. 1.1 for an overview of the possible instabilities which will later be referred to as unsafe modes.

These unsafe modes usually arise when braking or accelerating in a turn under low-friction road conditions. Due to the over-actuation of the system, the control strategy can be further constrained to allow for changes in the control input to the vehicle combination while still meeting goals and deadlines. This can allow for better control and avoidance of these unsafe modes [2]. A method used for constraining the force allocation is using a Safe Operating Envelope (SOE) strategy, constraining the possible CA using predefined limits which can be derived using simulations of various models with fixed input values during the manoeuvre (such as steering rate

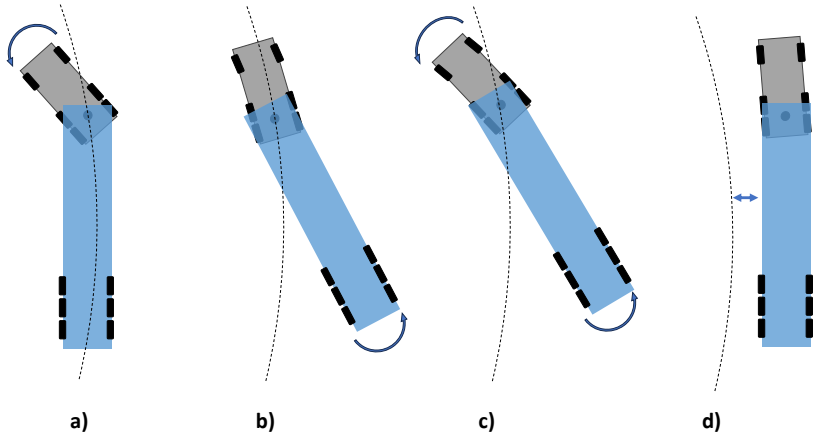



**Figure 1.1:** Top-down view of unsafe modes, with the dotted line representing the road path. a) Jack-knifing, b) Trailer swing, c) Combinational spin-out, d) off-tracking

and longitudinal forces). A uniform sampling determines force allocations for a specific set of environmental conditions and vehicle parameters. The process can be repeated for multiple scenarios to obtain multiple SOEs and can then be applied to constrain a control scheme to achieve safe CA [1], [3], [4].

Multiple control strategies can be implemented to achieve the desired CA, but for this thesis, a Model Predictive Control (MPC) scheme will be investigated due to its beneficial properties for CA, i.e., using the predictive capabilities of the MPC and constraining the state of the vehicle. Thus, the MPC will be used instead of the SOE to perform the safe CA. In automotive applications, MPC can be especially useful as mentioned due to the ability to place constraints on the system and the predictive capabilities of the control scheme, enabling the system to anticipate the ever-changing behaviour of automotive systems. Furthermore, it enables the change of control inputs (such as steering and longitudinal force) to the vehicle over time, which the current SOE strategies are not able to provide. Some disadvantages of using an MPC are the higher computational complexity of the control scheme and the need for an accurate model for reliable predictions [5], [6].

Integrating Machine learning (ML) into such a controller can offer benefits in various areas, such as computational efficiency and reduced complexity [7]. An MPC controller can require high computational power to calculate an optimal solution within a feasible region over the horizon. Using an ML model can decrease the computational complexity of the solution by making predictions about the system. ML can enable the system to use a shorter horizon while still maintaining a feasible solution, which then can be extended past the horizon of the controller, increasing the robustness of the defined control scheme. Suppose an approximation is made beyond the horizon. In that case, the goal is to limit the states at the final time step of the horizon to those that are feasible according to the approximation. The

enforced limitation at the final time step states is referred to as the terminal set.

Thus, an approximation can be made of a terminal constraint set which enables the system to remain in a feasible state past the horizon. If this is achieved, the MPC can be considered to be persistently feasible [8]. An illustration of the idea is presented in Fig. 1.2.

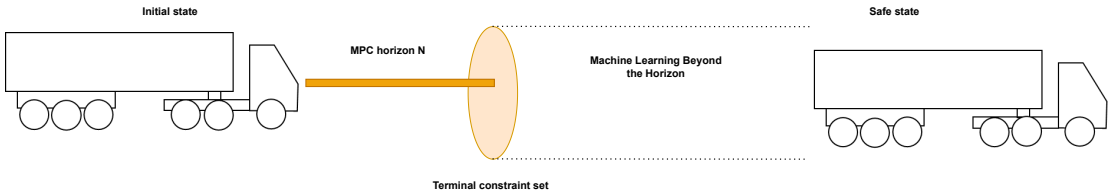



**Figure 1.2:** Illustration of using machine learning to approximate the terminal set in MPC, giving the MPC information regarding what happens after the defined horizon.

## 1.2 Purpose

The thesis will mainly focus on the longitudinal forces and steering inputs, meaning the controller should allocate longitudinal forces and determine the steering angle to follow a specified reference path and acceleration profile, as provided by the driver or autonomous driver model. It aims to achieve both a safe and energy-efficient CA. This means that it should be energy-efficient while still being in a safe state using the control scheme. Furthermore, the purpose is to investigate whether it is possible to estimate a terminal set that can ensure the safety of the system beyond the horizon. This aims to increase the chance of finding a feasible control action that keeps the vehicle safe in future time steps. In short, the end product should function in the following way: an acceleration request comes from a driver or autonomous driver model. The longitudinal forces should then be distributed between the tractor and any attached trailers while also keeping the combination on the defined reference path, which is controlled via the steering input of the tractor. The system should ensure safety while also allowing for better energy usage compared to traditional control methods.

## 1.3 Aim

This thesis aims to use an ML model that approximates a terminal constraint set for a given vehicle combination that, according to the model, can keep the system safe beyond the horizon. This will then be integrated with an MPC application that finds the optimal control to achieve energy efficiency within the safety constraints of the MPC. To achieve this, these objectives are set up:

1. Develop a vehicle model representing the dynamics of the system as a function of the vehicle inputs $u$.

2. Formulate an MPC with vehicle safety- and operational constraints, achieving safe vehicle motion throughout the horizon $N$. The system should also be suitable for online usage and maximizing energy efficiency during the manoeuvre.

3. Implement a machine learning infrastructure which can approximate a set. This includes finding an efficient way of sampling the state space.

4. Find a way to collect data of points inside the set for training the machine learning algorithm. This will allow the system to classify data points as belonging either to the set or not.

5. Integrate the approximated set as the terminal constraint set of the online MPC.

6. Investigate how well the approximated terminal constraint set improves the safety of the vehicle.

## 1.4 Scope

The scope of this thesis is defined as follows:

- The tractor-semitrailer combination has a three-axle tractor and a two-axle semi-trailer. The first axle on the tractor is steerable and the second and third axles are driven.

- All propulsion actuators on the vehicle combination are assumed to be electrical.

- All wheels on the combination are equipped with mechanical brakes and controlled per axle group.

- All parameters are assumed to be known and no uncertainties are added to the signals.

- Only deceleration scenarios will be used for training the ML model.

- Only one tractor and trailer combination per control scheme is used. If another vehicle combination is implemented, the ML model and NMPC/MPC controller must be redefined.

- The high-fidelity vehicle model will be assumed to be the ground truth.

- The developed vehicle model only models planar dynamics in the XY plane, describing the yaw motion of the vehicle.

## 1.5 Limitations

The predefined limitations of the project are listed below.

- Some parameters that in real life are hard to estimate, such as friction or slip angles, are assumed to be known.
- The developed control system is only tested in a simulation environment, not on a real vehicle.
- The estimation of the road curvature and the friction coefficient is constant along the horizon of the MPC.

## 1.6 Ethical and sustainability aspects

The main benefit of a successful project is that it will be possible to make allocations that are more energy-efficient than what is used today. From a sustainability point of view, this is beneficial since it will help to reduce the energy consumed by the vehicle, which in turn will have a positive environmental impact.

However, from an ethical point of view, it is not as obvious. The system is designed to make safe decisions which is of course a positive ethical aspect. It could prove to be more safe than the current methods used. However, since it should also make the most energy-efficient decision there could be situations where it becomes a trade-off between safe and energy-efficiency. It is therefore important that safety is prioritized and that the definition of safe is well-defined.

The project also contains functions to facilitate automated driving which can be a controversial ethical discussion when it comes to safety and decision-making. For now, the developed algorithms are modelled and tested in a virtual environment. If this developed solution is to be further tested in actual vehicle combinations, a more in-depth analysis should be conducted regarding the decision-making of autonomous systems and the ethical implications involved.

## 1.7 Disposition

Initially, the thesis introduces the deriving of the vehicle dynamics and creating a state space model in Chapter 2. The composition of the MPC is presented in Chapter 3, and a comparison between different solvers is made. Chapter 4 describes the trained ML model and corresponding sampling technique in more detail. Finally, in Chapter 5, the results of the implemented control scheme are presented and Chapter 6 contains the conclusions with the suggested future work directions.

# 2

# Vehicle modelling

To establish a control scheme, it is essential to have appropriate models that describe the dynamics of the vehicle system to be controlled. Different models can be used with different levels of complexity and with different purposes. This chapter clarifies the various models that are used in the thesis and their respective functions. For this thesis, a Volvo FH model is used with an electric power unit. The parameters of the setup can be found in Appendix A.

## 2.1 Non-linear single-track model

A non-linear model is obtained for the AHV to describe the motion dependent on the input to the system. A single-track model is used from a computational efficiency perspective since previous work shows that the performance of these single-track models is comparable in accuracy to the double-track models [9]. See Fig. 2.1 for an overview of the parameters used for modelling the system and the setup of these parameters. To describe the state of the vehicle, the longitudinal, lateral and yaw motion equilibrium of the tractor and trailer can be constructed to formulate the dynamics according to Newton's second law of motion [4], [10].

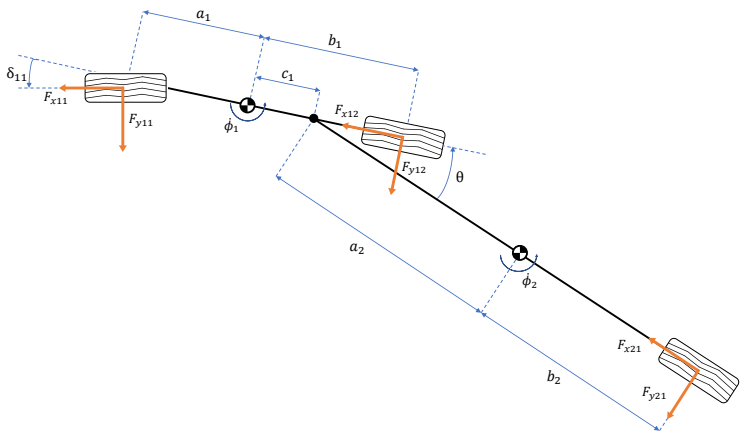



**Figure 2.1:** Illustration of the developed single-track model with according parameters and directions. All forces and moments are calculated according to the right-hand rule.

### 2.1.1 Newtonian mechanics modelling

From Fig. 2.1 and Fig. 2.2, showing the free-body diagram of the coupling point, the following equations can be derived. These include the longitudinal ($F_{xij}$) and lateral ($F_{yij}$) forces for each axle group where $i$ corresponds to the unit and $j$ the axle group on that unit, counted from the front axle. $v_{xi}$ and $\dot{v}_{xi}$ are the longitudinal speed and acceleration of unit $i$ respectively. $v_{yi}$ and $\dot{v}_{yi}$ is the lateral speed and acceleration of unit $i$, $\dot{\phi}_i$ is the yaw rate of unit $i$ and $\delta_{11}$ is the steering angle of the front axle on the tractor. $F_{cxi}$ and $F_{cyi}$ are the longidudinal and lateral coupling forces, $m_i$ is the mass and $J_i$ is the moment of inertia of unit $i$. $a_i$, $b_i$ and $c_i$ is the measurements of unit $i$ which can be seen in Fig. 2.1.

The force equilibrium along the longitudinal and lateral axes together with the moment of the yaw axis around the centre of gravity (CoG) for the tractor is given by:

$$m_1(\dot{v}_x - \dot{\phi}_1 v_y) = F_{x11}\cos\delta_{11} - F_{y11}\sin\delta_{11} + F_{x12} + F_{cx1} \tag{2.1}$$

$$m_1(\dot{v}_y + \dot{\phi}_1 v_x) = F_{x11}\sin\delta_{11} + F_{y11}\cos\delta_{11} + F_{y12} + F_{cy1} \tag{2.2}$$

$$J_1\ddot{\phi}_1 = a_1\left(F_{x11}\sin\delta_{11} + F_{y11}\cos\delta_{11}\right) - b_1F_{y12} - c_1F_{cy1} \tag{2.3}$$

Furthermore, the force equilibrium along the longitudinal and lateral axes together with the moment of the yaw axis around the CoG for the trailer is given by:

$$m_2(\dot{v}_{x2} - \dot{\phi}_2 v_{y2}) = F_{x21} + F_{cx2} \tag{2.4}$$

$$m_2(\dot{v}_{y2} + \dot{\phi}_2 v_{x2}) = F_{y21} + F_{cy2} \tag{2.5}$$

$$J_2\ddot{\phi}_2 = -b_2F_{y21} + a_2F_{cy2} \tag{2.6}$$

The force equilibrium for the coupling point is given by Equation (2.7) and (2.8) where $\theta$ is the articulation angle between tractor and trailer.

$$F_{cx1} = -F_{cx2}\cos\theta - F_{cy2}\sin\theta \tag{2.7}$$

$$F_{cy1} = F_{cx2}\sin\theta - F_{cy2}\cos\theta \tag{2.8}$$

The longitudinal velocities at the coupling point in the free-body diagram are equal to those in the CoG of the units since these are not affected by the angular velocities. The lateral velocities, on the other hand, will depend on the units angular velocities. Thus, these are calculated at each point of interest along the vehicle combination using equations (2.9).

The coupling forces on the vehicle can be expressed through the acceleration of the tractor translated to the trailer which makes it possible to reduce the number of states. The acceleration and velocity translations through the rigid body of the tractor and trailer can be formulated according to (2.9), taken from [11].

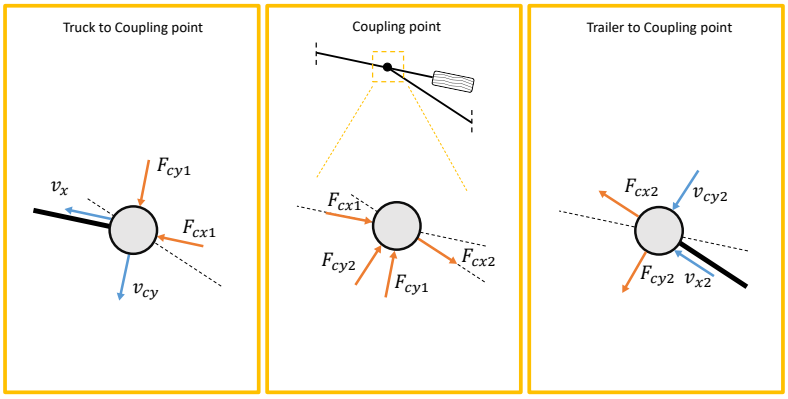



**Figure 2.2:** Free-body diagram of the coupling point with coupling forces represented in orange and the velocities in blue. The forces are modelled according to action and reaction forces.

$$\begin{aligned} \boldsymbol{v}_A &= \boldsymbol{v}_B + \dot{\boldsymbol{\phi}} \times \boldsymbol{r}_{AB} \\ \dot{\boldsymbol{v}}_A &= \dot{\boldsymbol{v}}_B + \ddot{\boldsymbol{\phi}} \times \boldsymbol{r}_{AB} + \dot{\boldsymbol{\phi}} \times (\dot{\boldsymbol{\phi}} \times \boldsymbol{r}_{AB}) \end{aligned} \tag{2.9}$$

By using the translation equations (2.9), the trailer velocities and accelerations can be expressed as:

$$v_{x2} = v_x \cos\theta - (v_y - c_1\dot{\phi}_1)\sin\theta \tag{2.10}$$
$$v_{y2} = (v_y - c_1\dot{\phi}_1)\cos\theta - a_2\dot{\phi}_2 + v_x \sin\theta \tag{2.11}$$
$$\dot{v}_{x2} = a_2\dot{\phi}_2^2 - (\dot{v}_y - c_1\ddot{\phi}_1)\sin\theta + (\dot{v}_x + c_1\dot{\phi}_1^2)\cos\theta \tag{2.12}$$
$$\dot{v}_{y2} = -a_2\ddot{\phi}_2 + (\dot{v}_y - c_1\ddot{\phi}_1)\cos\theta + (\dot{v}_x + c_1\dot{\phi}_1^2)\sin\theta \tag{2.13}$$

### 2.1.2 Tyre models

Besides forces of propulsion and braking acting on the tyres in longitudinal direction $F_{xij}$, lateral forces act on the tyres of the vehicle $F_{yij}$. The longitudinal forces are a consequence of the inputs to the vehicle system. These lateral forces are caused due to the lateral slip ($s_{yij}$) of the tyre and can be derived in multiple ways depending on which tyre model is applied [9]. The lateral slip of the tyre is calculated using the positional velocities in the $x$ and $y$ direction of the respective wheels ($v_{xij}$ and $v_{yij}$).

$$s_{y11} = \frac{v_{y11}}{v_{x11}} \tag{2.14}$$
$$s_{y12} = \frac{v_{y12}}{v_{x12}} \tag{2.15}$$
$$s_{y21} = \frac{v_{y21}}{v_{x21}} \tag{2.16}$$

The velocities are derived with respect to the velocity in the CoG of each vehicle unit as follows:

$$v_{x11} = v_x \cos\delta_{11} + (v_y + a_1\dot{\phi}_1)\sin\delta_{11} \tag{2.17}$$
$$v_{y11} = -v_x \sin\delta_{11} + (v_y + a_1\dot{\phi}_1)\cos\delta_{11} \tag{2.18}$$
$$v_{x12} = v_x \tag{2.19}$$
$$v_{y12} = v_y - b_1\dot{\phi}_1 \tag{2.20}$$
$$v_{x12} = v_{x2} \tag{2.21}$$
$$v_{y12} = v_{y2} - b_2\dot{\phi}_2 \tag{2.22}$$

Vehicle tyres typically exhibit non-linear behaviour when relating lateral slip to the lateral force produced by the tyre. When using a linear tyre model, the tyre can produce an infinite amount of lateral force proportional to the slip. A more realistic model is the Magic Formula or similar non-linear models but these are computationally less efficient and can be hard to tune [12]. In the following sections, some different tyre model equations will be introduced which can be used by the vehicle models.

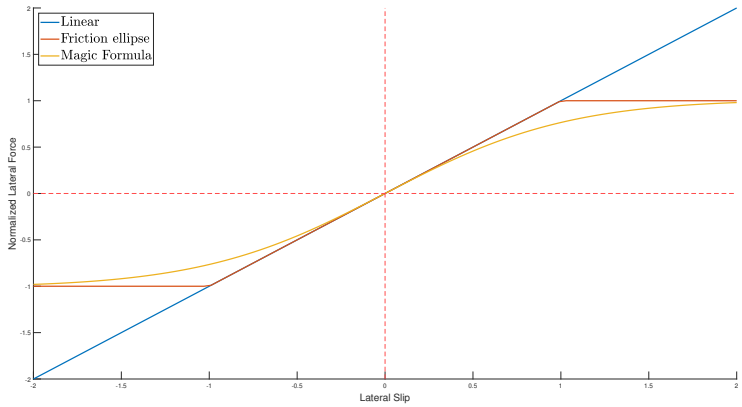



**Figure 2.3:** Comparison between different tyre models and their behaviour concerning the lateral force produced. A toy example is used to illustrate the nature of the different tyre models with a normalised lateral force produced.

#### 2.1.2.1 Linear tyre model

The linear tyre model is computationally efficient but lacks the ability to model the loss of lateral force beyond a certain lateral slip of the tyre, as seen in Fig. 2.3. The lateral force produced by the tyre is proportional to the slip using the cornering stiffness coefficient $C_{ij}$. Therefore, when the lateral slip angle is small, the model works well since the non-linear models first exhibit linear behaviour.

$$F_{y11} = -C_{11}s_{y11} \tag{2.23}$$
$$F_{y12} = -C_{12}s_{y12} \tag{2.24}$$
$$F_{y21} = -C_{21}s_{y21} \tag{2.25}$$

#### 2.1.2.2 Non-linear tyre model

To get a more realistic model, the friction circle model which combines the linear model with a calculated lateral force limit as seen in Fig. 2.3 can be used. Since the longitudinal force is known, this can be used in deriving the limit of the tyre based on the friction to the road [13]. Equation (2.26) - (2.28) does this, where $F_{zij}$ is the vertical forces and $\mu$ is the friction coefficient.

$$F_{y11}^{lim} = \mu \cdot F_{z11} \cdot \sqrt{1 - \left(\frac{F_{x11}}{\mu \cdot F_{z11}}\right)^2} \tag{2.26}$$

$$F_{y12}^{lim} = \mu \cdot F_{z12} \cdot \sqrt{1 - \left(\frac{F_{x12}}{\mu \cdot F_{z12}}\right)^2} \tag{2.27}$$

$$F_{y21}^{lim} = \mu \cdot F_{z21} \cdot \sqrt{1 - \left(\frac{F_{x21}}{\mu \cdot F_{z21}}\right)^2} \tag{2.28}$$

The values in (2.26) - (2.28) can then be used to limit the linear model only to produce the permitted amount of lateral force. This enables the use of the computationally efficient linear tyre model, with the benefit of exhibiting the grip limitations of real tyres to a certain degree. The limitations can be modelled as follows and allows for easy reformulation as constraints for the MPC later on.

$$-F_{y11}^{lim} \leq F_{y11} \leq F_{y11}^{lim} \tag{2.29}$$

$$-F_{y12}^{lim} \leq F_{y12} \leq F_{y12}^{lim} \tag{2.30}$$

$$-F_{y21}^{lim} \leq F_{y21} \leq F_{y21}^{lim} \tag{2.31}$$

### 2.1.3 State space model

The coupling forces in equations (2.7) and (2.8) are reformulated to sum up all forces acting on the vehicle due to the propulsion and vehicle cornering, which results in the following force equilibrium. The accelerations $\dot{v}_{x2}$ and $\dot{v}_{y2}$ are acquired from equation (2.12) and (2.13). The total force from the tyres acting on the coupling point can be modelled as follows.

$$F_{xtot} = F_{x11}\cos\delta_{11} - F_{y11}\sin\delta_{11} + F_{x12} - F_{x21}\cos\theta - F_{y21}\sin\theta \tag{2.32}$$

$$F_{y_{tot}} = F_{x11}\sin\delta_{11} + F_{y11}\cos\delta_{11} + F_{y12} + F_{x21}\sin\theta - F_{y21}\cos\theta \tag{2.33}$$

Using these force equations, the total force equilibrium of the coupling point can be defined, utilizing equations (2.32) and (2.33).

$$-2F_{cx1} = F_{xtot} - m_1\dot{v}_x + m_2\dot{v}_{x2}\cos\theta + m_2\dot{v}_{y2}\sin\theta \tag{2.34}$$

$$-2F_{cy1} = F_{y_{tot}} - m_1\dot{v}_y - m_2\dot{v}_{x2}\sin\theta + m_2\dot{v}_{y2}\cos\theta \tag{2.35}$$

By collecting and combining the above-described equations we get the following system of equations:

$$m_1(\dot{v}_x - \dot{\phi}_1 v_y) = F_{x11}\cos\delta_{11} - F_{y11}\sin\delta_{11} + F_{x12} + F_{cx1} \tag{2.36}$$
$$m_1(\dot{v}_y + \dot{\phi}_1 v_x) = F_{x11}\sin\delta_{11} + F_{y11}\cos\delta_{11} + F_{y12} + F_{cy1} \tag{2.37}$$
$$J_1\ddot{\phi}_1 = a_1\left(F_{x11}\sin\delta_{11} + F_{y11}\cos\delta_{11}\right) - b_1F_{y12} - c_1F_{cy1} \tag{2.38}$$
$$J_2\ddot{\theta} = J_1\ddot{\phi}_1 - a_2\left(F_{cy1}\cos\theta + F_{cx1}\sin\theta\right) + b_2F_{y21} \tag{2.39}$$
$$\dot{\theta} = \dot{\phi}_1 - \dot{\phi}_2 \tag{2.40}$$

This can then be solved for the 5 unknowns $[\dot{v}_x, \dot{v}_y, \ddot{\phi}_1, \ddot{\theta}, \dot{\theta}]$, resulting in an explicit state space model $\dot{x} = f(x, u)$ of the system with the state and input vectors as presented below. Here, an additional state is added for the steering wheel angle $\delta_{11}$, enabling the use of the steering wheel angular velocity as an input instead. The explicit formulation for this state is then $\dot{x}_6 = u_1$ where $x$ and $u$ are the updated state and input vectors given in Equation 2.41.

$$x = \begin{bmatrix} v_x \\ v_y \\ \dot{\phi}_1 \\ \dot{\theta} \\ \theta \\ \delta_{11} \end{bmatrix} \qquad u = \begin{bmatrix} \dot{\delta}_{11} \\ F_{x11} \\ F_{x12} \\ F_{x21} \end{bmatrix} \tag{2.41}$$

## 2.2 Single-track vehicle model

A low-fidelity plant model for testing the developed control scheme, a variation of the proposed single-track motion model is implemented. The motion equations are similar to those presented previously in equations (2.36) - (2.40), with some added states to gain more information about the system. Global positioning is added for information regarding the relative position of the vehicle in the world based on its velocity and heading. Thus, the states $X$, $Y$ and $\phi_1$ are added, describing the global coordinates and heading of the tractor unit. Therefore, these additional equations of motion are added to the plant model.

$$\dot{X} = v_x\cos\phi_1 - v_y\sin\phi_1 \tag{2.42}$$
$$\dot{Y} = v_x\sin\phi_1 + v_y\cos\phi_1 \tag{2.43}$$
$$\dot{\phi}_1 = x(3) \tag{2.44}$$

This extends the state vector x of the plant model to $x = [v_x, v_y, \dot{\phi}_1, \dot{\phi}_2, \theta, \phi_1, X, Y, \delta_{11}]$.

## 2.3 High fidelity vehicle model

The non-linear high-fidelity model used in this thesis is referred to as the Volvo Transport Model (VTM), developed internally in MATLAB/Simulink by Volvo

Group Trucks Technology for simulation of tractor and trailer combinations for various control development projects among others. The model consists of rigid bodies for the tractor cab, axles, frame, trailer frame and axles. These rigid bodies are connected, allowing free movement between the bodies with a torsional viscoelasticity-type connection, offering six degrees of freedom.

The VTM, when compared to real-life test data from similar scenarios, proved to show predictions adequately equal to the real system [14]. Thus, the VTM is assumed to be equal to a real vehicle system, which enables the use of the VTM for testing developed control schemes. Furthermore, it enables the gathering of data regarding the safety of the vehicle combination during simulations to evaluate the system response. VTM enables the extraction of comprehensive sensor data from multiple sensors and data buses, which enables the use of the same state vector $x$ as for the previously defined state space model in Section 2.2.

## 2.4 Single-track model validation

To validate whether the model works accordingly, a comparison is made between the developed model and the high-fidelity VTM model. The lateral performance of the single-track model is compared to that of the VTM to check the validity of the tyre models. The tyre models are analysed since these contribute most to the uncertainty of the low-fidelity model. Since the model is simpler than the high-fidelity model, some inaccuracies are expected. The single-track model also employs a lumped axle group, instead of modelling a double rear axle group on the tractor and semitrailer. Principals such as tyre scrubbing due to multiple axles will thus not be modelled. In Fig. 2.4, the lateral forces produced by the single-track model are compared to those of the VTM.

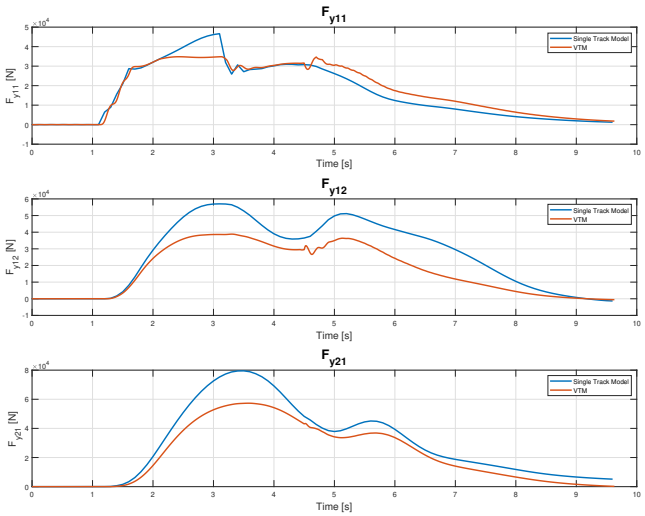



**Figure 2.4:** Comparison between the lateral forces produced by the single-track tyre model and the VTM tyre model. Some offset can be observed between the two models but the behaviour of the produced lateral force is similar with a difference in amplitude.

To achieve a good model performance, the cornering stiffness coefficients $C_{ij}$ are tuned according to the performance of the VTM to minimize the error between the lateral forces. Upon comparison, the single-track model exhibits similar behaviour in the production of lateral force compared to the VTM. During the linear region of the non-linear tyre model, this behaviour is expected with some offset. However, when the slip nears the limit of the friction circle, the forces start to diverge. This is consistent with the theory of the tyre models presented in Fig. 2.3, where the produced lateral force starts to drop off compared to the linear model near the limit. Similar behaviour can also be seen in Fig. 2.5, where the drop-off can be visualised based on simulation data from the scenario in Fig. 2.4 of the VTM and the single-track model. The simulation data points are sampled from time 0 s to 3.5 s.

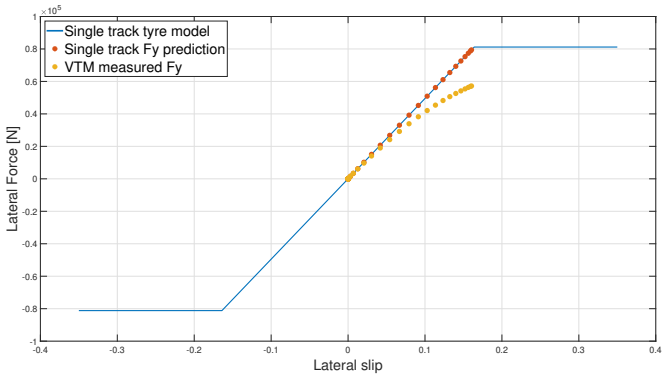



**Figure 2.5:** Comparison between the lateral forces produced by the trailer using the single-track tyre model and the VTM tyre model near the limit of the friction circle. A small divergence can be observed when nearing the limit, showing the non-linear behaviour of the VTM tyre model.

The single-track model can be seen to produce more lateral force than the VTM when nearing the limit, which creates the illusion of more grip in the single-track model. To achieve similar performance in the single-track model, it is necessary to implement a non-linear tyre model.

# 3

# Model Predictive Control

Model predictive control (MPC) holds interesting properties which can be utilized for vehicle motion control. Beyond the predictive capabilities of the control scheme, constraints can be implemented to define the limitations of the system.

In this work, the objective of the MPC is to find the control inputs (steering angle rate and longitudinal forces on each axle) that minimize energy use while keeping the vehicle safe (no unsafe modes) and following the path throughout the manoeuvre. The MPC finds the optimal control inputs using the cost function while keeping the system within the predefined constraints to ensure safety. The model introduced in the previous chapter will be used as the motion model of the MPC and in this chapter, the characteristics of the controller are described. To formulate an MPC, firstly an Optimal Control Problem (OCP) is derived to describe the behaviour of the controller. Equation (3.1) describes the OCP of the MPC with non-linear state space model $f\left(x_k, u_k, p_k\right)$ with the additional parameter vector $p_k$, which will be further explained in Section 3.1.2.

$$
\begin{aligned}
&\min_{u_k, x_k} \sum_{k=0}^{N-1} V(x_k, u_k) + V_N(x_N) \\
&\text{s.t. } x_{k+1} = f\left(x_k, u_k, p_k\right), \forall k \in [0, N-1] \\
&\quad x_0 = \hat{x}_0 \\
&\quad x_k \in \mathbb{X}, \forall k \in [0, N-1] \\
&\quad u_k \in \mathbb{U}, \forall k \in [0, N-1] \\
&\quad g(x_k, u_k, p_k) \leq 0, \forall k \in [0, N-1] \\
&\quad x_N \in \mathbb{X}_f \subseteq \mathbb{X}
\end{aligned}
\tag{3.1}
$$

Additionally, it contains the state constraint set $\mathbb{X}$, control constraint set $\mathbb{U}$, general non-linear inequality constraint $g(x_k, u_k, p_k)$ and terminal set $\mathbb{X}_f$. Furthermore, the initial state $x_0$ is constrained to be equal to the measurement $\hat{x}_0$ of the current vehicle state. The cost function $(\sum_{k=0}^{N-1} V(x_k, u_k) + V_N(x_N))$ consists of the cost up until the second to last step $V(x_k, u_k)$ and the target cost $V_N(x_N)$ of the terminal state. A detailed description of all OCP components is derived in this chapter.

## 3.1 Deviation state space model

The state space model $f(x_k, u_k, p_k)$ formulated in the OCP, i.e. Equation (3.1), is based on the derivations from Chapter 2. The state space model presented in Equation (2.41) contains the states; longitudinal and lateral speed, yaw rate of unit 1, articulation angle rate, articulation angle and the steering angle. These are sufficient to explain the motion, but for the MPC, the motion should be described in the form of state deviations instead, simplifying the procedure to track setpoints in the system. Furthermore, states are added to implement the path- and acceleration profile tracking, using error deviations between the defined path and the state of the vehicle combination. An illustration of the deviation control scheme can be seen in Fig. 3.1. Thus, the form $\dot{x} = f(x, u, p)$ is changed to $\dot{x}_e = f(x_e, u, p)$ to formulate the deviation targets for the MPC. The path and acceleration profiles are assumed to be known, with the road data consisting of a curvature and a friction coefficient that the MPC assumes to be constant over the horizon. In this thesis, the curvature provided to the MPC always represents the average curvature for 20 meters ahead of the truck. The acceleration profile is a vector of length $N$ that describes the requested acceleration, and it is assumed to not violate the longitudinal velocity constraints of the MPC (defined in Section 3.2).

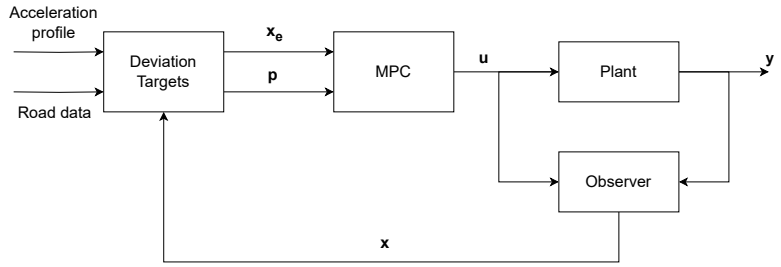
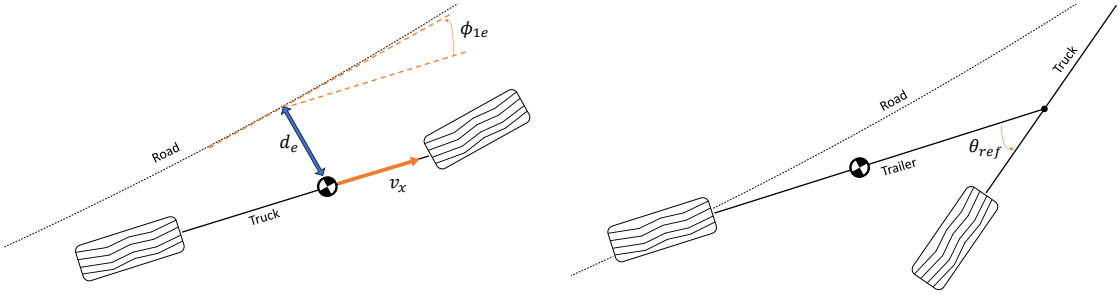



**Figure 3.1:** MPC block diagram containing the control scheme and the transformation from states to deviation states. An observer will feed forward the observed state of the plant model to be used during the next time step.

### 3.1.1 Setpoints

As illustrated in Fig. 3.2a, by adding the states of distance from the centre of the road $d_e$ and the heading error of the tractor relative to the path $\phi_{1_e}$, the possibility of following a path with the tractor is introduced. To ensure the vehicle combination follows the path, both the distance from the reference point of the tractor and the heading error to the path should be zero. The reference point can be placed freely on the vehicle (e.g. in this thesis, it is placed on the stationary, unladen CoG of the tractor). The trailer is set to follow the path through the articulation angle $\theta$ with a reference formulated such that the axle group of the trailer is on the centre of the road, as seen in Fig. 3.2b.




**(a)** The setpoints for the tractor to follow the defined road path.

**(b)** The setpoints of the articulation angle between the tractor and the trailer.

**Figure 3.2:** Illustration of the correlation between the global position of the single-track model and the defined road path.

The reference value $\theta_{ref}$ is calculated based on the current radius of the path, assuming the heading error and distance of the tractor to the path is zero. Furthermore, assuming that the curvature of the path is positive or negative based on whether the path heading is left or right, the signum of the curvature $c_r$ is used to define the sign of the reference.

$$\theta_{ref} = \left(\pi - \arcsin\frac{R}{\sqrt{R^2 + c_1^2}} - \arccos\frac{c_1^2 + a_2^2}{2a_2\sqrt{R^2 + c_1^2}}\right)\operatorname{sgn}(c_r) \tag{3.2}$$

The reference angle is shifted by $\pi$ radians to define $\theta_{ref}$ as zero when the road path radius $R$ is large. The values for the distances can be found in Appendix A.

### 3.1.2 State space model

To be able to implement the new deviation state space model to the solvers, the explicit continuous form $\dot{x}_e$ of the state space is used. The acceleration profile $\dot{v}_{xref}$ is used to define the error state of the vehicle acceleration and the angular velocity of the tractor $\dot{\phi}_1$ is set to track the angular velocity of the road. Moreover, the derivative of the deviation from the centre of the road $\dot{d}_e$ is defined. $c_r$ is the curvture of the road.

$$\dot{x}_e = f(x_e, u, p) = \begin{bmatrix} \dot{v}_{xe} \\ \dot{v}_{y_e} \\ \ddot{\phi}_{1e} \\ \dot{\phi}_{1e} \\ \ddot{\theta}_e \\ \dot{\theta}_e \\ \dot{d}_e \\ \dot{\delta}_{11e} \end{bmatrix} = \begin{bmatrix} \dot{v}_x - \dot{v}_{xref} \\ \dot{v}_y \\ \ddot{\phi}_1 \\ \dot{\phi}_1 - (v_x\cos\phi_e - v_y\sin\phi_e)\cdot c_r \\ \ddot{\theta} \\ \dot{\theta} \\ v_x\sin\phi_e + v_y\cos\phi_e \\ \dot{\delta}_{11} \end{bmatrix} \tag{3.3}$$

When discretizing, the new state space model for deviation targets, the state space model is obtained on the form $x_e(k+1) = f(x_e(k), u(k), p(k))$. Where the states, input signals and parameters are the following.

$$x_e = \begin{bmatrix} v_{xe} \\ v_{y_e} \\ \dot{\phi}_{1e} \\ \phi_{1e} \\ \dot{\theta}_e \\ \theta_e \\ d_e \\ \delta_{11e} \end{bmatrix}, \qquad u = \begin{bmatrix} \dot{\delta}_{11} \\ F_{x11} \\ F_{x12} \\ F_{x21} \end{bmatrix}, \qquad p = \begin{bmatrix} c_r \\ \dot{v}_{ref} \\ \mu \\ v_{ref} \end{bmatrix} \tag{3.4}$$

This means that the deviation state space model is expanded, and in addition to the new states, the state space model also needs the curvature $c_r$ as a parameter input $p$ to allow trajectory tracking [15]. Moreover, additional reference values are added to implement the state deviation values for the other states. These values refer to the reference values of the remaining state variables.

### 3.1.3 Converting from plant to deviation state

When receiving new data $x$ from the plant, as provided in Chapter 2, this needs to be converted to the deviation state $x_e$, as illustrated in the deviation targets block in Fig. 3.1, and it is given by the following conversion equation:

$$x_e = f(x, p) = \begin{bmatrix} v_{xe} \\ v_{y_e} \\ \dot{\phi}_{1e} \\ \phi_{1e} \\ \dot{\theta}_e \\ \theta_e \\ d_e \\ \delta_{11e} \end{bmatrix} = \begin{bmatrix} v_x - v_{xref} \\ v_y \\ \dot{\phi}_1 \\ \phi_1 - \phi_{road} \\ \dot{\theta} \\ \theta - \theta_{ref} \\ (x,y)_{CoG} - (x,y)_{road} \\ \delta_{11} \end{bmatrix} \tag{3.5}$$

The velocity reference $v_{xref}$ is calculated based on the previously defined acceleration profile, integrated over the horizon. The angle of the road path $\phi_{road}$ is calculated based on the coordinates of the closest road point to the reference point of the tractor. In the model, the angle of the tractor is calculated directly, but in a real-world scenario, the angle relative to the world is hard to calculate. Thus, in a real vehicle combination, the relative angle error $\phi_{1e}$ between the road and the tractor must be measured rather than calculated. The same argument is made for the distance error $d_e$ between the reference point of the tractor and the road. These are calculated as the difference in global position in the model, but must in real-world applications be measured, e.g. using a GPS measurement.

## 3.2 Constraints

The constraints in this section are divided into two parts, those who regard ensuring the safety of the vehicle, and those who regard the operational limitations on either physical components or on the single-track model. The combination of these forms the constraints for the MPC, applied on all constraint sets $\mathbb{X}$, $\mathbb{U}$, $\mathbb{X}_f$ and the general non-linear constraint $g(x_k, u_k, p_k) \leq 0$.

### 3.2.1 Safety constraints

Choosing the safety constraints on the states of the vehicle is not straightforward since what defines an unsafe mode depends on multiple factors. For example, a certain tractor yaw rate value can be considered unsafe in a high-speed and low-curvature scenario but safe in low-speed and high curvature. Therefore, this is a state that cannot be bounded by a simple box constraint at least. However, by putting a simple box constraint on the states $\phi_e$, $d_e$ and $\theta$, the planar motion of the vehicle can limited to following the defined road path while maintaining safe vehicle motion throughout the curvature. Moreover, a constraint is placed on the state $v_{xe}$ ensuring that the reference acceleration profile is met with some degree of freedom. These deviation parameters can restrict the vehicle to the defined path of the road whilst also constraining the relative rotation of the vehicle to match that of the reference path. The box constraints can thus be formulated as follows for each value. These constraints are then implemented as a part of the state constraint set $\mathbb{X}$.

The values used to constrain the system are based on multiple factors. The articulation angle is constrained such that the axle group of the trailer can not be more than 1 meter from the centre of the road, which is calculated as $\arcsin\left(\frac{1}{a_2}\right)$. This is based on the average width of a Swedish road structure, defined by the Swedish Transport Administration (Trafikverket) [16]. Moreover, the maximum distance from the centre of the road $d_e$ can be determined similarly, defining the maximum distance from the reference point as 1 meter.

$$\phi_e \in \left[ \begin{array}{cc} -10\,\mathrm{deg} & 10\,\mathrm{deg} \end{array} \right] \tag{3.6}$$

$$d_e \in \left[ \begin{array}{cc} -1\,\mathrm{m} & 1\,\mathrm{m} \end{array} \right] \tag{3.7}$$

$$\theta \in \left[ \begin{array}{cc} -\arcsin\left(\frac{1}{a_2}\right) & \arcsin\left(\frac{1}{a_2}\right) \end{array} \right] \tag{3.8}$$

$$v_{xe} \in \left[ \begin{array}{cc} -1.3889\,\mathrm{m\,s^{-1}} & 1.3889\,\mathrm{m\,s^{-1}} \end{array} \right] \tag{3.9}$$

Since these safety conditions should hold over the entire horizon of the controller, the terminal constraint set is also constrained in an equal manner. Furthermore, since there are no further constraints on the time of arrival within the horizon or a global position of the system with respect to time, the safety constraints can be defined equally for the terminal constraint set $\mathbb{X}_f$.

### 3.2.2 Operational constraints

The operational limitations of the vehicle can be defined based on the physical limitations of the vehicle combinations used in the controller, e.g. actuator limits or maximum lateral acceleration. Thus, the values of these constraints can change accordingly. For this, the vehicle combination actuator limitations include maximum forces per axle group and steering wheel limits. For this particular vehicle, the actuator limits are described as follows. The total braking force available to the vehicle combination is defined by the maximum friction utilization of the normal force. This can be calculated as $mg\mu = F_z\mu$. Therefore, the maximum brake force (when the friction coefficient is $\mu = 1$, is $F_z$ for each axle group. The maximum acceleration force is dependent on the capacity of the power unit of the current vehicle combination. In the case of an electric semitrailer, a non-zero limit can also be implemented on the trailer axle group, but this is not a part of this thesis. The amount of vehicle propulsion allowed is determined by the value of $F_{12}^{max} = 10\,000\,\mathrm{N\,m}$. The currently selected vehicle is chosen to a reasonable value for the maximum force needed to propel the vehicle combination but it is not directly calculated based on the properties of the Volvo FH powertrain. This is disregarded since the focus of this study was mostly on deceleration cases.

$$F_{x11} \in \begin{bmatrix} -F_{z11} & 0\,\mathrm{N\,m} \end{bmatrix} \tag{3.10}$$

$$F_{x12} \in \begin{bmatrix} -F_{z12} & F_{12}^{max} \end{bmatrix} \tag{3.11}$$

$$F_{x21} \in \begin{bmatrix} -F_{z21} & 0\,\mathrm{N\,m} \end{bmatrix} \tag{3.12}$$

The steering angle $\delta$ is also limited by the constraint in Equation (3.13) which is a part of the state constraint set $\mathbb{X}$. The constraint implemented on the steering wheel input is based on the Ackermann steering geometry. Since the path used in the thesis will have a radius no smaller than 50 meters, a steering wheel angle of a maximum of 5 degrees is needed. Thus, the steering wheel angle is allowed to be 10 degrees to allow for overcorrection by the system.

$$\delta_{11} \in \begin{bmatrix} -10\,\mathrm{deg} & 10\,\mathrm{deg} \end{bmatrix} \tag{3.13}$$

Further, constraints are enforced on the system due to the physical limitations of the state space model used for control. To get a more realistic vehicle behaviour, a non-linear tyre model is preferred to model the slip limits of real tyres. The friction ellipse model proposed in equations (2.29) - (2.31) can be used to model this behaviour, but these functions are non-smooth, creating problems for the analytic OCP solver used later. Thus, the slip limits based on the friction ellipse are implemented as state constraints instead, limiting the use of lateral grip produced by the tyres. This is also beneficial from a safety standpoint as this ensures that the calculated input to the vehicle will not cause the tyres to reach the slip limit. The limits can be

formulated as follows.

$$F_{x11}^2 + F_{y11}^2 \leq \mu^2 F_{z11}^2 \tag{3.14}$$
$$F_{x12}^2 + F_{y12}^2 \leq \mu^2 F_{z12}^2 \tag{3.15}$$
$$F_{x21}^2 + F_{y21}^2 \leq \mu^2 F_{z21}^2 \tag{3.16}$$

These constraints can be rewritten to fit in the non-linear inequality constraint formulation $g(x_k, u_k, p_k) \leq 0$ according to equation (3.17) where $F_x = \begin{bmatrix} F_{x11} & F_{x12} & F_{x21} \end{bmatrix}$, $F_y = \begin{bmatrix} F_{y11} & F_{y12} & F_{y21} \end{bmatrix}$ and $F_z = \begin{bmatrix} F_{z11} & F_{z12} & F_{z21} \end{bmatrix}$ and the squares and division is done element-wise.

$$\frac{F_x^2 + F_y^2}{\mu^2 F_z^2} - 1 \leq 0 \tag{3.17}$$

The constraints are based on the longitudinal force input $F_x$, which is a part of $u$, for each axle group and $F_y$ calculated according to equations (2.23) - (2.25). The vertical forces on each axle group are calculated based on the masses of the units $m_i$ and the vertical force transferred from the trailer to the tractor, through the coupling point $F_{2c}$.

$$F_{z11} = \frac{m_1 b_1 g + F_{2c}(b_1 - c_1)}{a_1 + b_1} \tag{3.18}$$
$$F_{z12} = \frac{m_1 a_1 g + F_{2c}(a_1 + c_1)}{a_1 + b_1} \tag{3.19}$$
$$F_{z21} = \frac{m_2 a_2 g}{a_2 + b_2} \tag{3.20}$$
$$F_{2c} = \frac{m_2 b_2 g}{a_2 + b_2} \tag{3.21}$$

## 3.3 Cost function

The cost function enables the CA within the previously defined constraints. Thus, the cost function can be used for various purposes to select an optimal input to the system, but for this project, the cost function will be used for power coordination. There are multiple sophisticated cost function implementations for power coordination of multi-unit vehicle combinations. These can handle e.g. only an actuator on the tractor, an actuator on both the tractor and the trailer (a trailer with a so-called e-axle) and multiple different actuator characteristics [17]. Using these methods allows for modelling efficiency maps of the actuators and incorporating these into the cost function for optimal control.

For this thesis, the emphasis is put on developing a functional MPC with room for adaptation to different use cases and driving scenarios. Thus, as an initial cost function implementation, an assumption is made that the vehicle combination consists of a driven BEV and a traditional semitrailer with regular service brakes.

The driven axle of the tractor should therefore primarily be used during braking to maximize energy regeneration to the battery, maximizing the State Of Charge (SOC). The cost function can therefore be defined as a weighted least square problem prioritizing the use of the driven axle. Furthermore, costs are implemented on the deviation states which will be explained further below in this section.

$$V(x_k, u_k) = ||x_k||^2_{\mathbf{Q}} + ||u_k||^2_{\mathbf{R}} = x_k^\top \mathbf{Q} x_k + u_k^\top \mathbf{R} u_k \tag{3.22}$$

In addition to this, the cost function also contains the target cost which penalises the deviations from the reference at the end of the prediction horizon.

$$V(x_N) = ||x_N||^2_{\mathbf{P}} = x_N^\top \mathbf{P} x_N \tag{3.23}$$

The weight matrix $\mathbf{R}$ can be defined as follows for this simple example with the corresponding input vector $u$. For this vehicle combination, the driven axle is the rear axle of the tractor ($F_{x12}$).

$$\mathbf{u} = \begin{bmatrix} F_{x11} \\ F_{x12} \\ F_{x11} \\ \dot{\delta}_{11} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \frac{10}{F_{x11}^{max}} & 0 & 0 & 0 \\ 0 & \frac{1}{F_{x12}^{max}} & 0 & 0 \\ 0 & 0 & \frac{10}{F_{x21}^{max}} & 0 \\ 0 & 0 & 0 & R_{\dot{\delta}_{11}} \end{bmatrix} \tag{3.24}$$

The weight matrix $\mathbf{Q}$ is defined as follows.

$$\mathbf{Q} = \begin{bmatrix} Q_{v_{xe}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & Q_{v_{ye}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & Q_{\dot{\phi}_{1e}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Q_{\phi_{1e}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & Q_{\dot{\theta}_e} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & Q_{\theta_e} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Q_{d_e} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q_{\delta_e} \end{bmatrix} \tag{3.25}$$

The cost elements in $\mathbf{Q}$ are applied to deviation state variables to minimize these states from a stability perspective. States that are critical to the safety of the vehicle combination (e.g. $d_e$, $\phi_{1e}$, $\theta_e$) can have a higher cost to let the deviation from these reference values be as small as possible. These cost elements are flexible and can therefore be implemented according to the wanted behaviour of the system and application. Therefore, specific cost element values will not be presented.

The weight matrix $\mathbf{P}$ is selected to be $\mathbf{P} = \alpha \cdot \mathbf{Q}$ where $\alpha$ is a scaling factor set to be greater than or equal to 1 ($\alpha \geq 1$). It is suggested that this improves the stability of the controller and also that the reference tracking will have a smaller error [18]. Two different values of $\alpha$ are tested and the results are presented in Chapter 5.

## 3.4 Solvers

To solve the above formulated non-linear OCP, a non-linear solver is used to calculate the optimal control solution from an initial state $x_0$. Two different solvers were implemented, Casadi and Acados, and they were compared primarily based on real-time performance. Since the derived state-space model from section 3.1 is in continuous form ($\dot{x} = f(x, u, p)$), it needs to be discretized in order to be in the discrete form ($x_{k+1} = f(x_k, u_k, p_k)$). This was done using the explicit Runge-Kutta 4 method.

### 3.4.1 Casadi

The implementation in Casadi uses the non-linear Interior Point Optimizer (IPOPT) and is implemented according to [19]. Casadi allows for a clear, high-level definition of the OCP using the MATLAB interface. The IPOPT, on the other hand, has inadequate real-time solving performance for this particular non-linear OCP. It enables the user to get good insights into the defined OCP and facilitates tuning of the solver. These properties are particularly helpful when developing an initial version of the OCP formulation to detect points of improvement. The Casadi interface is implemented with box constraint boundary variables describing the boundaries of the state space $\mathbb{X}$, and box constraint boundary variables describing the boundaries of the input space $\mathbb{U}$, defined in Section 3.2. Furthermore, the cost elements are implemented according to Section 3.3.

The Casadi solver was implemented during the initial phase of the thesis, and consequently, it lacks certain constraints and cost elements that were integrated at a later stage. The solver was not further explored due to its underperformance in real-time applications; thus, the solver was not further developed and tested. Therefore, the Casadi solver will not be further evaluated in this work but was used primarily to develop the OCP formulation and evaluate its initial performance.

### 3.4.2 Acados

To be able to achieve real-time performance with the MPC, a more efficient interface is needed between the OCP formulation and the solver, along with a more suitable solver for real-time applications. To accommodate this, Acados adopts a different solver architecture compared to Casadi, using a Sequential Quadratic Programming (SQP) approach as its primary solver. The OCP can thus be reformulated as presented in Eq. (3.26) to fit the Acados interface.

$$\begin{aligned}
\min_{u_k, x_k, p_k} \quad & \sum_{k=0}^{N-1} l(x_k, u_k, p_k) + m(x_N, p_N) \\
\text{s.t.} \quad & x_{k+1} = f\left(x_k, u_k, p_k\right), \forall k \in [0, N-1] \\
& J_{bx} x_k \in \mathbb{X}, \forall k \in [0, N-1] \\
& J_{bx}^N x_N \in \mathbb{X}_f \\
& J_{bu} u_k \in \mathbb{U}, \forall k \in [0, N-1] \\
& \underline{h} \leq h(x_k, u_k, p_k) \leq \overline{h}, \forall k \in [0, N-1] \\
& \underline{h^e} \leq h^e(x_N, p_N) \leq \overline{h^e}
\end{aligned} \tag{3.26}$$

The constrained state space and terminal space $\mathbb{X}$, $\mathbb{X}_f$ and input space $\mathbb{U}$ are as presented in section 3.2. The matrices $J_{bx}$ and $J_{bu}$ describe which states and inputs should be constrained by the OCP. Furthermore, the general non-linear constraints can now be formulated using $h$ during the horizon and as $h^e$ at the last state of the horizon.

The cost functions in Acados, presented in Equation (3.27), can be defined as linear least squares based on the weighted norm of $\mathbf{W}$ and $\mathbf{W}_N$ for the terminal set. The matrices $V_x, V_u$ and $V_x^N$ map the states and inputs to the corresponding cost elements in the cost matrices $\mathbf{W}$ and $\mathbf{W}_N$ which are constructed according to Equation (3.28).

$$\begin{aligned}
l(x_k, u_k, p_k) &= \frac{1}{2} ||V_x x_k + V_u u_k - y_{ref}||^2_{\mathbf{W}} \\
m(x_N, p_N) &= \frac{1}{2} \left|\left|V_x^N x_N - y_{ref}^N\right|\right|^2_{\mathbf{W}_N}
\end{aligned} \tag{3.27}$$

$$\mathbf{W} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \qquad \mathbf{W}_N = \mathbf{P} \tag{3.28}$$

Acados uses Casadi to easily formulate the OCP in a high-level manner which then generates the solver in C-code for increased performance. The Acados solver is shown to reduce the computation time whilst not suffering a major trade-off in sub-optimality, compared to the IPOPT solver in Casadi [20]. A comparison can be made for the computation time between the two implemented solvers, and the Acados solver improved the computation time significantly. Computation times could be reduced from seconds to around 0.1 seconds on average.

## 3.5 Observations regarding the feasibility of the NMPC

With the implemented MPC, an observation can be made when simulating the controller for a certain horizon length $N$. As depicted in Fig. 3.3, the number of feasible points in the state space $\mathbb{X}$ decreases when the horizon $N$ increases. Ideally,

the horizon of the controller is infinitely long, but since this is physically impossible, a long enough $N$ must be chosen which is dependent on the application.

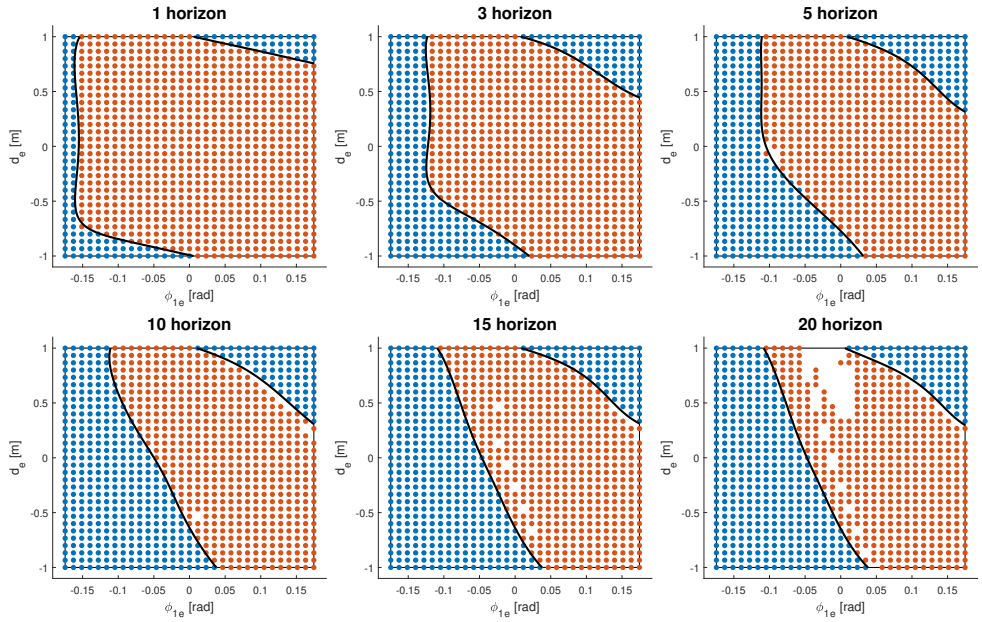



**Figure 3.3:** The difference between the number of feasible points dependent on the horizon length $N$ for a two-dimensional test case. The set can be observed to converge somewhere between horizon length $N = 15$ and $N = 20$.

The feasible area converges between horizon length $N = 15$ and $N = 20$. However, in this test case, only a two-dimensional space $\mathbb{X}$ is used with all other states being locked. This can influence the length of the horizon which is needed for the feasible area to converge. Thus, when the state space is extended to its full size, the horizon at which the convergence occurs will most probably be at a larger value for $N$.

## 3.6 Persistent feasibility

For the solution calculated by the MPC to be classified as feasible, the solution should be feasible at each time step of the control input to the system. To evaluate the feasibility of the controller, some theory regarding persistent feasibility is presented in this chapter according to [21] and [22]. Within the horizon, the feasibility can be defined as the following.

**Definition 3.6.1** (Feasibility within the horizon)**.** The controller provides a feasible solution for the motion model $x(k+1) = f(x(k), u(k))$ if there exists a predicted sequence of states and control inputs that reside within the constraints of the controller, and the final state is within the terminal constraint set. If these conditions

are met, $x(k)$ belongs to the feasible set $\mathcal{X}_N$.

$$\mathcal{X}_N = \{x(k) \mid x(k+i) \in \mathbb{X}, u(k+i) \in \mathbb{U}, x(k+N) \in \mathbb{X}_f \quad \forall i \in \left[ \begin{array}{cc} 0 & N-1 \end{array} \right]\}$$

To further evaluate the feasibility of the system, one needs to think about the next state after applying the control action. Ideally, this state $x(k+1)$, also belongs to the feasible set $\mathcal{X}_N$. This is referred to as persistent (or recursive) feasibility. Even without modelling disturbances and having a linear model, persistent feasibility is hard to assure. By introducing the concept of invariant sets, an initial definition of a persistently feasible controller can be made.

**Definition 3.6.2** (Positive invariant set)**.** An arbitrary non-empty set $\mathcal{S}$ is defined as positively invariant for a motion model $x(k+1) = f(x(k), K(x(k)))$, under a control law $K(x(k))$, if the following condition is met.

$$\mathcal{S} = \{x(0) \mid x(k) \in \mathcal{S}, \quad \forall k \in \mathbb{N}_+\}$$

Then, using the invariant set definition, a statement regarding a persistent feasible controller can be made.

**Definition 3.6.3** (Persistent (recursive) feasibility)**.** A MPC can be defined as persistently feasible if the feasible set $\mathcal{X}_N$ is a positively invariant set for the motion model $x(k+1) = f(x(k), K(x(k)))$, under a control law $K(x(k))$.

This definition of persistent feasibility is on the other hand dependent on a specified control law $K(x(k))$ by the MPC. This includes multiple factors that can influence the feasibility due to the behaviour of the MPC, rather than the primary focus of the feasibility issues due to constraint violations of the system. Changing the formulation of persistent feasibility will then generalize the previously defined definition.

Going back to the initial definition of persistent feasibility, that the next state $x(k+1)$ should also belong to the feasible set $\mathcal{X}_N$, should still hold. Starting at a feasible state $x(k) \in \mathcal{X}_N$ and applying the first control signal from this MPC iteration, one knows that the states up until $x(0 : N-1) \in \mathbb{X}$. The point of interest is then the last state which should again be part of the terminal constraint set $\mathbb{X}_f$. This means that if the system can remain within the terminal constraint set for another time step, the persistent feasibility will hold. This is defined as a control invariant set.

**Definition 3.6.4** (Control invariant set)**.** A set $\mathcal{C} \subseteq \mathbb{X}$ is control invariant of the motion model $x(k+1) = f(x(k), u(k))$ if the following condition is met.

$$\mathcal{C} = \{x(k) \mid \exists u(k) \in \mathbb{U}, x(k+1) = f(x(k), u(k)) \in \mathcal{C}\}$$

Thus, based on this definition, if the terminal set is control invariant persistent feasibility can be guaranteed. Therefore, a new version of the definition can be made.

**Definition 3.6.5** (Persistent feasibility using control invariance)**.** The defined MPC is persistently feasible if the terminal constraint set $\mathbb{X}_f$ is a control invariant set.

## 3.7 Calculating control invariant sets

As presented in Definition 3.6.5, the terminal constraint set $\mathbb{X}_f$ should be control invariant for persistent feasibility of the MPC, with there existing multiple control invariant sets in the state space $\mathbb{R}^n$. The maximum control invariant set, defined as $\mathcal{C}_\infty$, contains all subsets of control invariant sets ($\mathcal{C} \subseteq \mathcal{C}_\infty$). One way to calculate $\mathcal{C}_\infty$ is to perform the following recursion.

$$\Omega_{k+1} = \text{pre}(\Omega_k) \cap \Omega_k, \quad \Omega_0 = \mathbb{X} \tag{3.29}$$

Where the precursor set is calculated according to equation 3.30.

$$\text{pre}(\mathcal{S}) = \{x_k \in \mathbb{R}^n \mid \exists u_k \in \mathbb{U}, x_{k+1} = f(x_k, u_k, p_k) \in \mathcal{S}\} \tag{3.30}$$

The set $\Omega_k$ can here be explained as the set of initial points inside $\mathbb{X}$ where there exists a control sequence of length $k$ that keeps the system inside $\mathbb{X}$ for $k$ time steps. When the recursion in equation (3.29) has converged, i.e. $\Omega_{k+1} = \Omega_k$ for some $k$, the maximum control invariant set is found. This means that $\Omega_k = \mathcal{C}_\infty$, where the index $k$ is called the determinedness index [23]. An illustration of the described sets can be seen in Fig. 3.4.

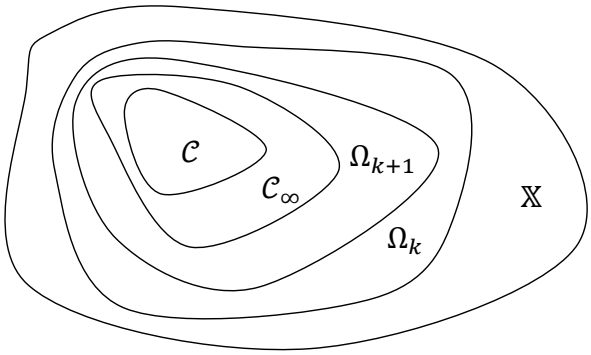



**Figure 3.4:** Calculatable sets in the state space $\mathbb{X}$. It illustrates how the precursor sets $\Omega$ shrink until converging at $\mathcal{C}_\infty$.

On the other hand, these calculations are hard to perform, especially on non-linear systems, since there is no analytical way to perform the set operations [24]. Thus, other methods are being explored to find and/or estimate these sets.

# 4

# Persistent feasibility with machine learning

Since the motion model is non-linear there is no feasible method of calculating $\Omega_k$ analytically. However, if it is assumed that an oracle $\mathcal{O}$ that can output whether a state $x \in \mathbb{X}$ is in the set $\Omega_k$ or not, one could use this to gather data points that could then be used to approximate the set $\Omega_k$.

$$\mathcal{O}(x) = \begin{cases} +1 \text{ if } x \in \Omega_k \\ -1 \text{ otherwise} \end{cases} \tag{4.1}$$

If the set is approximated for a high enough index $k$ (above the determinedness index) this could be assumed to be an approximation of the maximum control invariant set $\mathcal{C}_\infty$ as described in section 3.7. The problem that arises is then how one constructs such an oracle without knowing the set. One possibility of how such an oracle can be constructed is explored in the next section (4.1).

## 4.1 MPC as the oracle

One way to construct the oracle is to use the MPC itself. By setting the prediction horizon of the MPC to $N = k$ and the terminal constraint to $\mathbb{X}_f = \mathbb{X}$ the oracle in equation (4.2) could be constructed.

$$\mathcal{O}(x_0) = \begin{cases} +1 \text{ if MPC finds a feasible solution} \\ -1 \text{ otherwise} \end{cases}, \quad \text{for } x_0 \in \mathbb{X} \tag{4.2}$$

If the MPC finds a solution it means that the initial point lies in the set $\Omega_k$. So if the horizon is large enough (above the determinedness index), this oracle can be used to tell if a point lies in the maximum control invariant set $\mathcal{C}_\infty$ or not. In other words, the MPC can classify data points $x \in \mathbb{X}$ as being in $\mathcal{C}_\infty$. However, this only holds under the assumption that the MPC always finds the solution if it exists. This is not

necessarily the case and sometimes the optimization fails by for example reaching the maximum SQP iteration limit. Therefore, all points where this is the case were removed and only the points where the MPC either succeeded or failed before the iteration limit were used.

The next problem that arises is which points $x_0 \in \mathbb{X}$ should be chosen to create this data set. One trivial choice would be to grid the space to get an even distribution of points across the state space. However, one problem with this is that the MPC is quite computationally heavy and a single prediction takes around 0.1 seconds to calculate. And since the state space is 12-dimensional, just having 10 points per dimension would give $10^{12}$ points which would take approximately 3170 years to classify. This is of course not a feasible solution and another approach is needed. Another choice would be to randomly pick a certain amount of points in the state space and use this to estimate the set. The problem with this is that many points will also be needed here to accurately estimate the boundary.

A good choice would be to pick the most informative points, meaning the ones that are close to the boundary of the set, and use these. However, this is not a trivial choice since to pick points near the boundary, one needs to know the boundary. And if one knows the boundary, one would already know the set. However, if an Active Learning (AL) approach is used (presented in the next section), one could iteratively get a prediction of the boundary, use the current prediction to find the most informative points of that iteration, and use these points to improve the prediction of the boundary during the next iteration [25].

Fig. 4.1, represents how the set $\Omega_k$ can be approximated using the MPC for the same scenario as presented in Section 3.5. One can further see that the sets exhibit the same behaviour as illustrated in Fig. 3.4, where the determinedness index has approximately been reached (these sets were constructed using gridding which was only possible since the example is only two-dimensional).

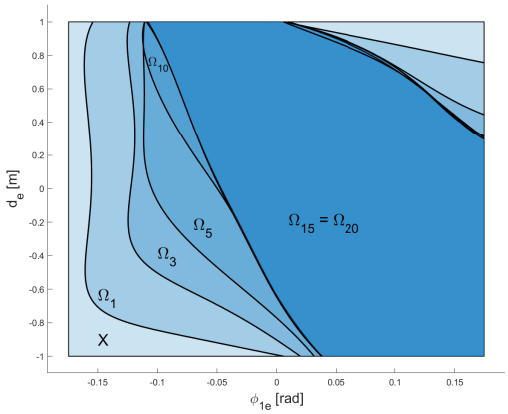



**Figure 4.1:** Visualisation of the convergence of the feasible area taken from the same test case as in section 3.5. One can see that the convergence is reached between $\Omega_{15}$ and $\Omega_{20}$. The data used for this figure is the same as for Fig. 3.3.

However, for a larger state space $\mathbb{X}$, the determinedness index must still be reached. This cannot be visualised as in the 2-dimensional case like the previous case which makes it harder to analyse where the convergence occurs. The horizon length is set to $N = 100$ to ensure that the index is attained. The value of $N$ could be slightly reduced, but this would need further investigation to determine the minimum value of the determinedness index.

## 4.2 Active Learning

AL is a special case of a supervised learning approach. Using classification in an iterative process, the most promising points are selected to increase the overall prediction performance during the next iteration. The benefit of AL, in this case, is that the labelled dataset can be drastically reduced since the AL algorithm chooses which data points $x \in \mathbb{X}$ that are most suitable for learning the decision boundary [26]. Based on the theory in [25] and [26], the AL algorithm used in this thesis was developed. An overview of the algorithm can be seen in Algorithm 1.

**Algorithm 1** Active learning

**Require:** Set of unlabeled samples, $\mathcal{S}$
**Require:** Initial set of labeled samples, $\mathcal{L}_0$
**Require:** Oracle, $\mathcal{O}$
**Require:** Final number of labelled samples, $N_L$
**Require:** Batch size, $B$
**Require:** Iteration counter, $n \leftarrow 0$
**while** $|\mathcal{L}_n| < N_L$ **do**
  $C_n \leftarrow$ classifier trained on $\mathcal{L}_n$
  $\mathcal{S}_r \leftarrow$ random subset of $\mathcal{S}$
  Find $B$ best samples in $\mathcal{S}_r$ using uncertainty from $C_n$
  $\mathcal{L}_{s,n} \leftarrow$ label B samples using oracle $\mathcal{O}$
  $\mathcal{L}_{n+1} \leftarrow$ append $[\mathcal{L}_n, \mathcal{L}_{s,n}]$
  $n \leftarrow n + 1$
**end while**
$C \leftarrow$ final classifier trained on $\mathcal{L}_{n+1}$

The algorithm starts with an arbitrary number of unlabeled data points $\mathcal{S}$. This is chosen to be all points $[x, p]$ belonging to the state- and parameter space. This set is named $\mathbb{X}_{ext}$ and is defined in equation (4.3).

$$\mathbb{X}_{ext} = \{[x, p] \mid x \in \mathbb{X}, p \in \mathbb{P}\} \tag{4.3}$$

The parameter space $\mathbb{P}$ is chosen to be all parameters $p = [c_r, \dot{v}_{ref}, \mu, v_{ref}]$ satisfying the following constraints. The curvature is set to only study larger road radii (excluding e.g. roundabouts). The velocity and acceleration limits are defined to follow with the vehicle's handling capabilities. Velocities close to zero are excluded, as they

do not pose any risk of unsafe behaviour. Finally, the friction coefficient $\mu$ is set to range between icy roads and dry asphalt, covering the range of road conditions.

$$c_r \in \begin{bmatrix} -\frac{1}{50} & \frac{1}{50} \end{bmatrix} \tag{4.4}$$

$$\dot{v}_{ref} \in \begin{bmatrix} -9.81 & 0 \end{bmatrix} \tag{4.5}$$

$$\mu \in \begin{bmatrix} 0.1 & 1 \end{bmatrix} \tag{4.6}$$

$$v_{ref} \in \begin{bmatrix} \frac{5}{3.6} & \frac{90}{3.6} \end{bmatrix} \tag{4.7}$$

An initial set of labelled data points $\mathcal{L}_0$ is then calculated according to Algorithm 2. The initial labelled dataset is labelled using the oracle $\mathcal{O}$ and defined to be balanced according to a balancing index $\zeta \in [\ 0 \quad 1\ ]$. The balancing index is implemented to ensure the initial dataset is fairly balanced since this can influence the convergence of the algorithm. The value of the balancing index $\zeta$ defines the minimum number of feasible points in the data set, e.g. if $\zeta = 0.4$, minimally 40 per cent of the data set will contain feasible points.

**Algorithm 2** Selecting suitable points for $\mathcal{L}_0$

**Require:** Size of $\mathcal{S}$ minimally the same as $N_L$
**Require:** Balancing index $\zeta \in [0, 1]$
$N_{feas} \leftarrow \text{round}(\zeta \cdot N_L)$, minimum size of feasible points in $\mathcal{L}_0$
**while** $n_+ + n_- < N_L$ **do**
    $x \leftarrow \mathbb{X}$, random sample from the state space
    $x_l \leftarrow \mathcal{O}(x)$
    $x_{\mathcal{L}} \leftarrow [x, x_l]$, concatenate state and label
    **if** $x_l = 1$ **then**
        $\mathcal{L}_0 \leftarrow [\mathcal{L}_0, x_{\mathcal{L}}]$
        $n_+ \leftarrow n_+ + 1$
    **end if**
    **if** $x_l = -1$ and $n_- \leq N_L - N_{feas}$ **then**
        $\mathcal{L}_0 \leftarrow [\mathcal{L}_0, x_{\mathcal{L}}]$
        $n_- \leftarrow n_- + 1$
    **end if**
**end while**

Using $\mathcal{L}_0$, a classifier $C_0$ is trained to create an initiatory prediction of the set $\Omega_k$. Based on these initial values, a first informative selection can be made of which points in the state space could add information to the current prediction. This is calculated based on the uncertainty of the trained classifier across the state space. The uncertainty model is calculated as a maximization problem, maximizing informativeness within the new batch of samples $B$. The uncertainty of a sample $x_n \in \mathbb{X}$ is computed based on the prediction of the labels $y_n$ by the trained classifier

$C_n$, resulting in the distribution $p(y_n \mid x_n, C_n)$. A discrete cross-entropy ascertains the total uncertainty of the sample calculated as follows.

$$\epsilon(x_n) = -\sum_{y\in\{-1,1\}} p(y \mid x_n, C_n) \log_2 \left(p(y \mid x_n, C_n)\right) \tag{4.8}$$

The maximization problem can then be formulated based on the calculated cross-entropy values for the selected points. The maximization problem extracts the points of highest informativeness accordingly.

$$\mathcal{L}_{s,n} = \underset{|s|\leq B}{\arg\max}\ \ \epsilon(s) \tag{4.9}$$

This is calculated for all points inside a set $\mathcal{S}_r$ which is constructed by a uniformly distributed random sampling of the set $\mathbb{X}_{ext}$, as described in Equation (4.3). Out of these points $B$ samples are chosen. The decision is made by taking the $B\eta$ most informative points according to equation (4.9). $B(1-\eta)$ points are then chosen randomly. Together, these $B$ samples are chosen as the next batch of points to be labelled. $\eta$ is a balancing index $\eta \in [\ 0 \quad 1\ ]$ that introduces some noise in the decision to avoid overfitting to the previous classifiers prediction. The batch of samples is then labelled and added to the labelled dataset $\mathcal{L}_n$. This procedure is repeated iteratively until the number of labelled samples reaches a predefined limit $N_L$.

### 4.2.1 Classifier

AL can be used with various classifiers for passive learning purposes (training on the entire labelled dataset at once). For this project, a Support Vector Machine (SVM) is chosen as the main classifier $C_n$. An SVM can perform non-linear classification using the so-called kernel trick [27]. The idea is that the non-separable (due to non-linearity) data points in the regular state space $\mathbb{R}^n$, are transformed using the mapping function $\phi(x)$ to a higher-dimensional feature space $\mathcal{F}$. A hyperplane can be trained to separate the data points linearly in this feature space. The kernel $K(x, x)$ can be defined as the inner product of the mapping function, i.e. $K(x, x) = \langle\phi(x), \phi(x)\rangle$. The SVM holds some beneficial properties which argue for the use of this classifier in AL. The SVM produces a linear decision boundary in the kernel space domain $\phi(x)$, simplifying the calculation of the informativeness of the data points. Furthermore, the calculated decision boundary can easily be extracted using the defined kernel function, which is beneficial later for the MPC [28]. Using the classifier together with the AL algorithm, the set can be approximated. Fig. 4.2 shows an example of how this algorithm works in a made-up 2-dimensional example for illustration purposes. There it can be seen how the algorithm manages to create a good classifier matching the true distribution, by fitting new data around the current predicted decision boundary of each iteration. More results regarding the performance of the AL algorithm will be introduced in Chapter 5.

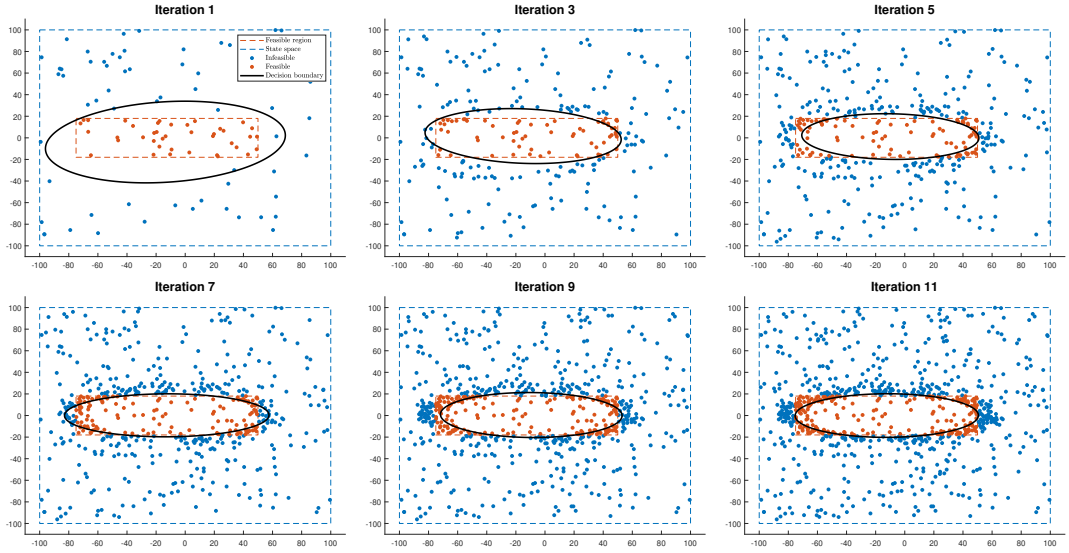



**Figure 4.2:** Illustration of the AL algorithm operating on a 2D dataset. Additional data points are added around the uncertainty of the state space during each iteration, typically near the current prediction boundary.

On the other hand, the MATLAB classifier has limitations in high-dimensional spaces according to the fitcsvm function description [29]. To the author's knowledge, there currently does not exist any other alternatives in MATLAB to train an SVM. Therefore, this will be further discussed in Chapter 6 to evaluate other directions.

## 4.3 SVM to constraint

After retaining the trained classifier $C_n$, the decision boundary of the final iteration of $C_n$ is to be extracted. The decision boundary will be used to formulate the predicted control invariant terminal constraint set $\mathbb{X}_f$. Since SVM uses internal support vectors to classify data points into two classes, utilizing the support vectors, the decision boundary can be calculated in the state space $\mathbb{R}^n$. This is computed by translating the decision boundary back from the feature space $\mathcal{F}$, using the sign of the decision function in equation (4.10) [30].

$$f(x) = \langle \mathbf{w}, \phi(x) \rangle + b = \sum_{n \in SV} \alpha_n y_n K(s_n, x) + b \tag{4.10}$$

The weight vector $\mathbf{w} \in \mathcal{F}$ of the support vectors can be defined using the following parameters. The Lagrange multiplier $\alpha_n$ for each support vector, the corresponding label of the support vector $y_n = \{-1, 1\}$ and the state space value $x_n$. Since the Lagrangian multipliers only are non-zero for support vectors, the notation for the state space coordinate $x_n$ can be changed to $s_n$ to clarify that these are support

vectors, i.e. $s_n \in SV$, where $SV$ is the set of support vectors.

$$\mathbf{w} = \sum_{n \in SV} \alpha_n y_n \phi(s_n) \tag{4.11}$$

The kernel can vary based on the desired outcome of the transformation, where some common versions are linear, gaussian or polynomial kernels. For this thesis, the polynomial kernel is chosen due to predictable behaviour in higher dimensions. The order of the polynomial is defined as $d$.

$$K(s, x) = \left(1 + s \cdot x^\top\right)^d \tag{4.12}$$

The sign of the decision boundary function $f(x)$ will determine the predicted class of a data point in $\mathbb{R}^n$, i.e. $f(x_n) \geq 0$ if $y_n = +1$ and $f(x_n) < 0$ if $y_n = -1$. Thus, the function can be used as a terminal set constraint for the MPC, indicated by $f(x) \geq 0$. The decision boundary can now be translated back to the regular state space $\mathbb{R}^n$, illustrated in Fig. 4.3 for a two-dimensional set of data points.

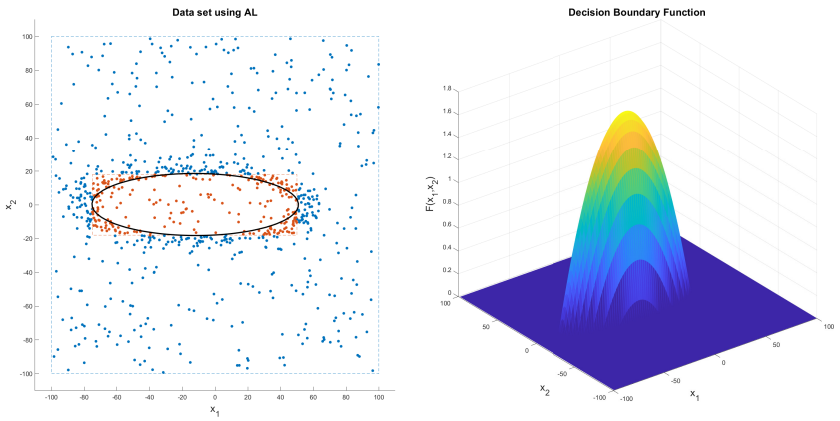



**Figure 4.3:** The extracted decision boundary using the AL algorithm. All data points where the value of $f(x) \geq 0$ are plotted on the surface. Values of $f(x) < 0$ are set equal to zero.

## 4.4 Hyperparameters

In the AL algorithm, multiple parameters can be changed to affect the performance of the algorithm. In this section, some of these will be presented and the choices made for this thesis will be motivated. The parameters will be explored using a 2-dimensional example (same as the previous section). This is made to understand the effects of the parameters easier to motivate what will be tested on the reduced 9-dimensional case. The state space was reduced to 9 dimensions in order to facilitate the lack of performance in higher dimensions of the fitcsvm function. The variables $\dot{\phi}_{1_e}$, $\dot{\theta}_e$ and $v_{y_e}$ are removed because they are assumed to be close to the defined

reference. Further investigation can be made to identify additional states that can be excluded to reduce the dimensionality.

### 4.4.1 Initial and final set size

The size of the initial labelled set $\mathcal{L}_0$ affects the prediction of the first classifier. As one can see in Fig. 4.4, the size of the initial set $\mathcal{L}_0$ will only influence the duration needed for the algorithm to converge but it does not alter the accuracy considerably. When the dimensionality of the problem increases, the initial set will also need to increase accordingly to give a reasonable impression of where the decision border is located, but a particularly large $\mathcal{L}_0$ is not necessary to increase the accuracy.

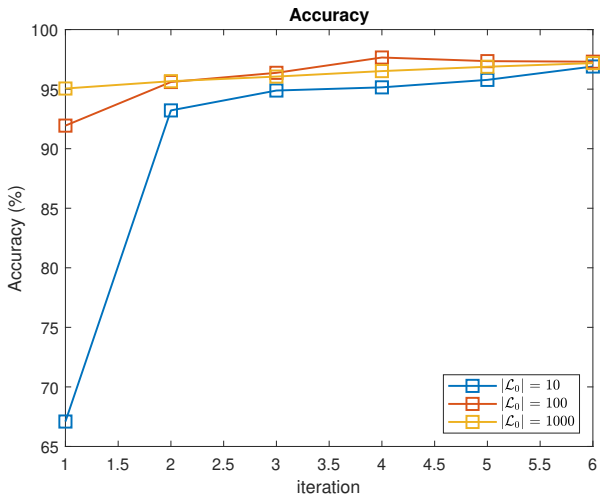



**Figure 4.4:** The accuracy of the model, based on the number of iterations of the AL algorithm.

The number of iterations $N_L$ needed for the accuracy to converge is dependent on the number of parameters to be trained and the size $\mathcal{L}_0$ as described earlier.

### 4.4.2 Configuration of the unlabelled batch

The batch size $|B|$ holds the same properties as the size of the initial labelled set $\mathcal{L}_0$. Increasing the batch size will speed up the rate of convergence; however, it is important to bear in mind that labelling the larger batch will also take longer. When introducing randomness to the batch, as is currently done through the balancing index $\eta$, a smaller batch size may cause the accuracy to decrease more. This can also be seen in Fig. 4.5.

The choice of samples for the batch $B$ impacts the ability of the algorithm to generalize to uncorrelated data (to the decision boundary). Adding a random factor $\eta$ enables the algorithm to adapt to new data while learning from the uncertainty of the precious predictor. Another approach would be to add a value to the entropy to create a divergence from the previous set of labelled data $\mathcal{L}_{n-1}$. This creates a weighting between informativeness and new data.

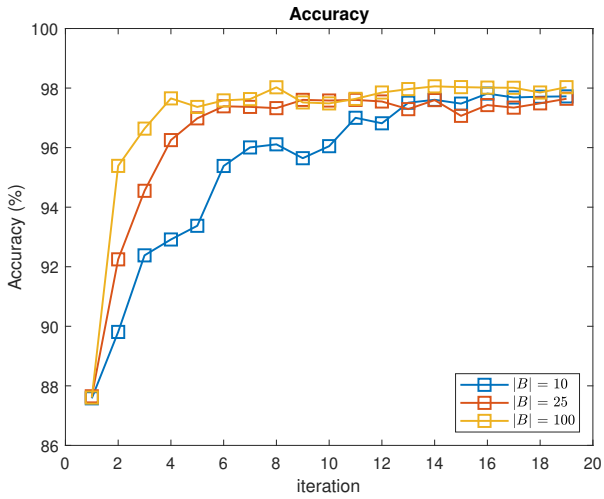
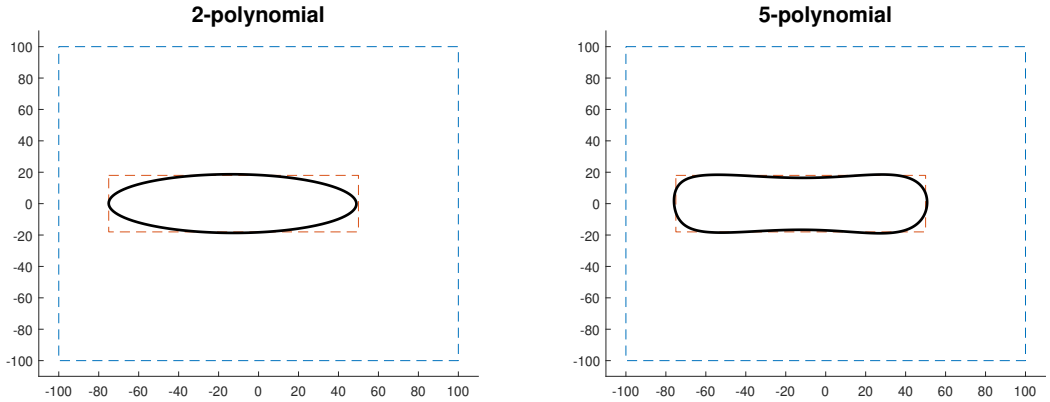



**Figure 4.5:** Convergance dependent on the size of the unlabelled batch $B$

### 4.4.3 SVM parameters

The kernel changes the flexibility of the decision boundary, changing the degree of freedom as can be seen in Fig. 4.6. The downside of using a higher-degree polynomial is that the number of parameters grows, making it more difficult to convert the decision boundary into a set constraint. Moreover, other kernels can also be implemented in the SVM, but these likewise require more parameters to get a good approximation of the decision boundary. From the standpoint of the MPC, it is preferable to minimize the number of required parameters.




**(a)** The decision boundary approximated by a second-degree polynomial.

**(b)** The decision boundary approximated by a fifth-degree polynomial.

**Figure 4.6:** Difference between polynomial degrees for the decision boundary of the SVM. A higher degree polynomial will enable more flexibility in the boundary as depicted.

To tune the ability of the classifier to determine how incorrect classifications are handled, a custom cost function for misclassification can be introduced. There are two types of misclassification, false positives (FP) and false negatives (FN). When

the classifier makes an accurate prediction, this is referred to as a true positive (TP) or a true negative (TN). In the case of the presented classifier, a feasible solution is referred to as a positive solution and an infeasible point is of the negative class.

The relative number of FP misclassifications is the precision score, which can be described as follows. When the precision score approaches 1, false positive classifications are nearly nonexistent.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.13}$$

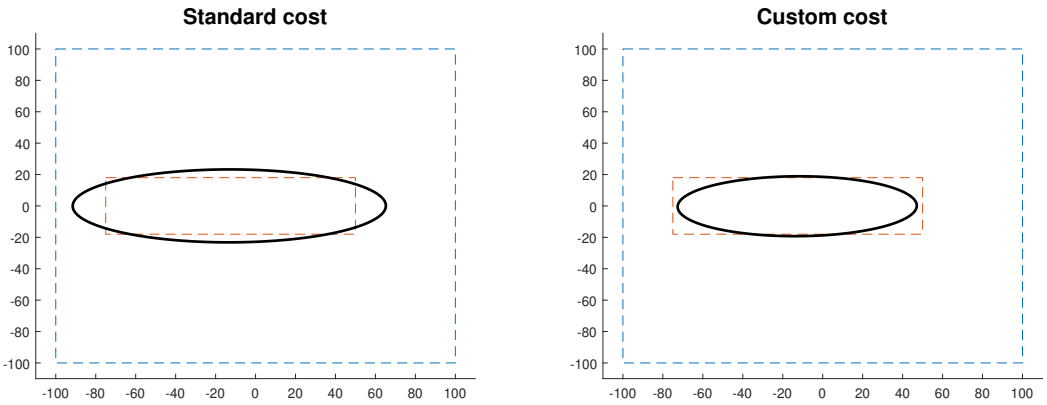
The relative number of FN misclassifications is the recall score, which can be described as follows. Accordingly, when the recall score approaches 1, the occurrence of false negative classifications is nearly none [31].

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.14}$$

The combination of these two scores is the accuracy score. This score evaluates the overall score of the classification. These scores will also be referred to in the Results chapter.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4.15}$$

If an inner approximation of the set is required, a high precision score reflects this. This will decrease the number of TP values captured by the decision boundary but will further reduce the misclassification of FP values, as can be seen in Fig. 4.7. The misclassification cost can be tuned accordingly and can be used to penalise the FN and FP classifications. Maximization of precision can be important in safety-critical systems, where FP values can lead to unsafe behaviour. This will also be the case in this thesis since FP values can lead to unsafe control signals from the MPC. Hence, the misclassification cost can be utilized to encourage the desired behaviour.




**(a)** Standard cost, where both misclassifications are weighted equally.

**(b)** Custom cost, where precision is prioritised to create an inner approximation.

**Figure 4.7:** The influence of the misclassification costs, where the cost can be tuned to achieve a certain type of misclassification prioritisation.

# 5

# Results

This chapter will present the performance of the control scheme with the integrated control invariant terminal set approximation. The outcomes of the system are depicted using a range of scenarios and environmental conditions. In Fig. 5.1, the system architecture is illustrated to give an overview of the presented solution. The parameters describing the vehicle model are used to determine the training of the AL algorithm and the OCP formulation of the MPC. Furthermore, the online control loop is closed for close-loop control of the vehicle.

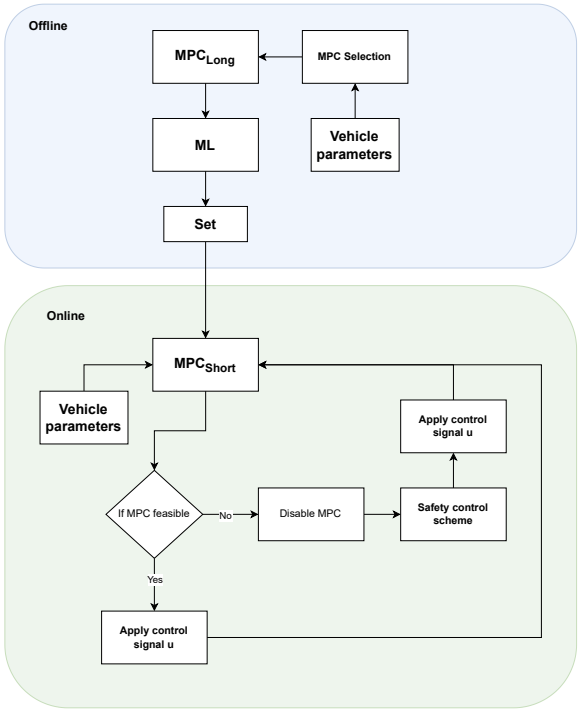



**Figure 5.1:** Overview of the system architecture with the offline ML approximation of the terminal set and the online calculation of control signals. The flowchart introduces the backup controller, which is used if the MPC results in an infeasible solution or reaches the iteration limits of the SQP. Parameters regarding the vehicle combination will be incorporated in the MPC used both online and offline.

## 5.1 Active Learning results

This section will present the results of the AL algorithm and will be split up into two. The first subsection will show some results for the same two-dimensional case as used in section 3.5 which is more interpretable since it can be plotted. The second subsection will then present the results for the total 9-dimensional case.

### 5.1.1 Two-dimensional test case

The test case here is the same as in section 3.5, meaning the heading angle error ($\phi_{1_e}$) and distance to road ($d_e$) are the only two dimensions in the state and parameter spaces that are not locked. The other states are defined as specific values presented in equation (5.1). These represent the truck going straight forward in $55\,\mathrm{km\,h^{-1}}$ and reaching a curve with radius 70 m at the initial state.

$$x_e = \begin{bmatrix} v_{x_e} \\ v_{y_e} \\ \dot{\phi}_{1_e} \\ \phi_{1_e} \\ \dot{\theta}_e \\ \theta_e \\ d_e \\ \delta_{11_e} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\dot{\phi}_{1_{ref}} \\ \phi_{1_e} \\ 0 \\ -\theta_{ref} \\ d_e \\ 0 \end{bmatrix}, \qquad p = \begin{bmatrix} c_r \\ \dot{v}_{ref} \\ \mu \\ v_{ref} \end{bmatrix} = \begin{bmatrix} 1/70 \\ -3 \\ 0.7 \\ 55/3.6 \end{bmatrix} \tag{5.1}$$

Fig. 5.2 shows the AL algorithm working for this test case. It can be seen that in about 5-6 iterations it converges to a solution approximating the set. As described in Chapter 4, the AL algorithm will add the most informative points of the state space to the new labelled data set. This behaviour can also be observed in Fig. 5.2. New data points are added around the current decision boundary since this is where the algorithm is the most uncertain, with the added noise to the batch resulting in a spread across the state space.

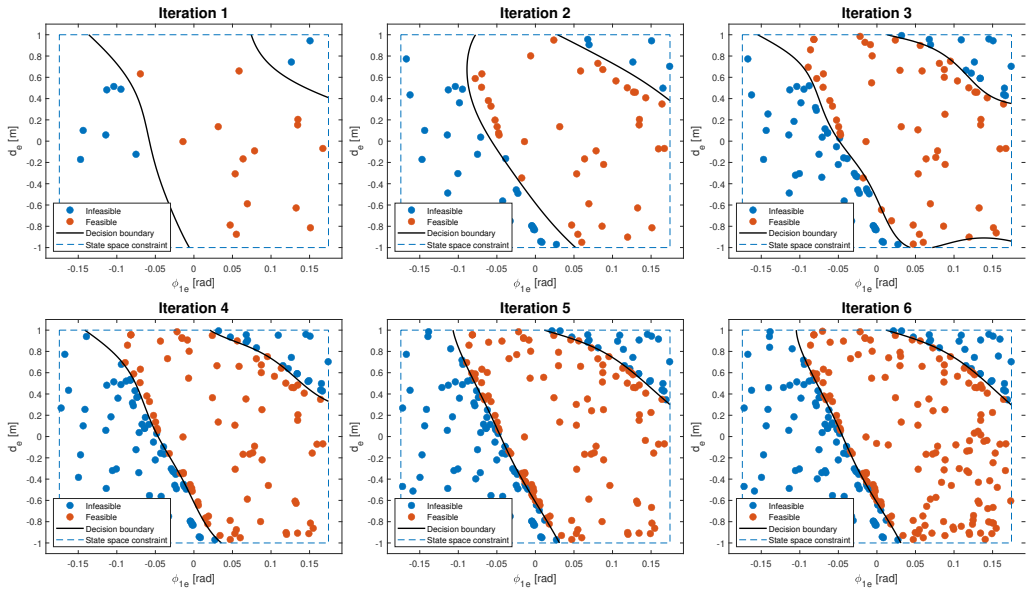



**Figure 5.2:** Six iterations of the AL algorithm during training. It can be seen that more data points are added during each iteration around the most informative areas of the state space.

In figure 5.3 the set after 6 iterations of the AL algorithm is plotted on top of a test set comprised of 101x101 gridded points. It can be seen that the approximated set matches the test data very well. The figure also shows a decision boundary for an SVM trained on equally many, but random, points. There it is seen that it is not as good, showcasing the AL algorithm's ability to find and train a classifier on more informative points is more effective.

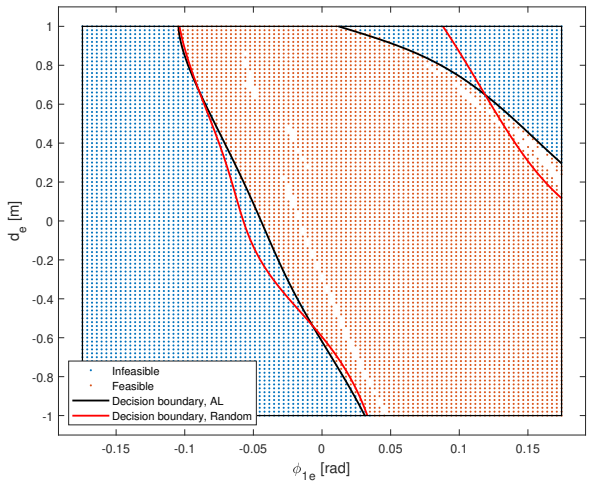



**Figure 5.3:** Decision boundary of AL vs Random sampling plotted on top of the test set. Fewer data points are needed than random sampling to get a better approximation of the set. The data points that are missing are due to the iteration limits of the solver.

Figure 5.4 shows the accuracy, precision and recall of the AL algorithm classifier, on the same test set as in figure 5.3, for all 21 iterations that were performed. Furthermore, it shows the results for when training an SVM classifier on the same number, but randomly sampled, points as each respective iteration of the AL algorithm used. There it is seen that AL can reach high accuracies in much fewer data points than random sampling. It also achieves an overall higher performance compared to random sampling.

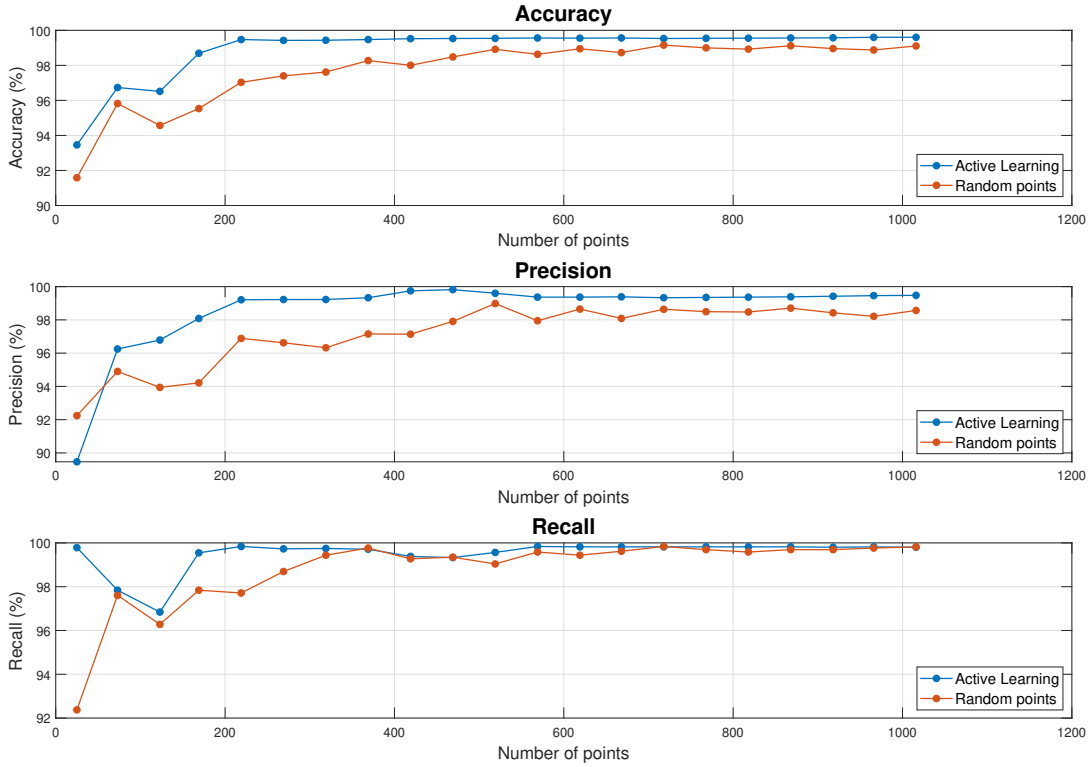



**Figure 5.4:** Performance metrics of AL and random sampling plotted against the number of data points. Furthermore, the AL method outperforms random sampling, achieving better results with fewer data points, as shown in Fig. 5.3 as well.

### 5.1.2 AL on all dimensions

When testing in all 9 dimensions the results can not be plotted in the same way as in the previous subsection.

Figure 5.5 shows the accuracy, precision and recall plots for when the AL method is used in the 9-dimensional case. It was tested for two different kernels (2- and 3-polynomial) and two different misclassification costs, the standard one where both FNs and FPs are penalised equally and one where FPs have a misclassification double the cost on FPs. The test set used to calculate these metrics consists of 15000 randomly sampled data points. It can be seen that the 3-polynomial kernel outperforms the 2-polynomial. It further shows that when using double cost, the precision increases but the accuracy and recall decrease, showcasing the tradeoff that

is normally the case. However, this effect is most noticeable for the 2-polynomial, while the 3-polynomial does not decrease as much in recall when using the higher cost. Worth noting is that during the labelling of points, around 25 per cent of the time the solver reached maximum iterations, resulting in those points being removed. The implications of this will further be discussed in Chapter 6.

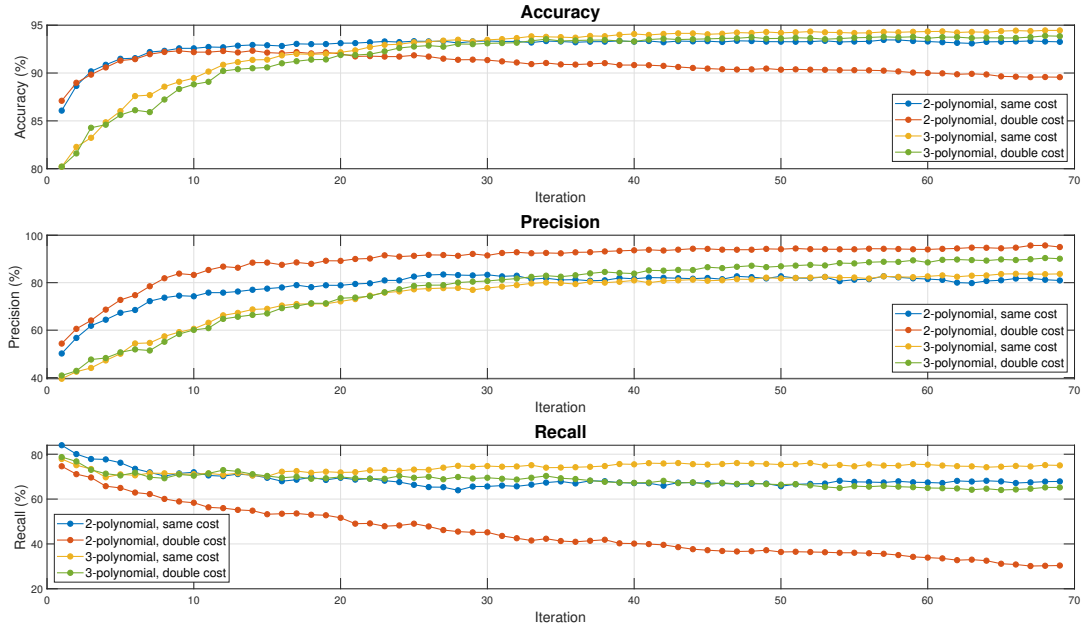



**Figure 5.5:** An illustration of the accuracy, precision and recall for the described kernels and misclassification costs. The 3-polynomial kernel shows better scores and the ability to generalise the data. Overall a decrease in accuracy, precision and recall is noticed compared to Fig. 5.4.

## 5.2 MPC controlling the VTM

In this section, 4 test cases will be presented where the MPC is tested with the VTM as a plant model. The tests will be in varying difficulty, meaning how dangerous of a scenario the truck is in. The MPC is here implemented as described in Fig. 5.1 where both the terminal set and backup of the controller are used. For the backup controller, the steering angle is decided by a path follower with a sample time of 1 ms. The brake force distribution is done proportional to the axle load as presented in Equation (5.2), where $F_{req}$ is the requested braking force, converted from the reference acceleration.

$$F_x = \begin{bmatrix} F_{x11} \\ F_{x12} \\ F_{x21} \end{bmatrix} = \begin{bmatrix} F_{req} \cdot F_{prop,11} \\ F_{req} \cdot F_{prop,12} \\ F_{req} \cdot F_{prop,21} \end{bmatrix} \quad \text{where} \quad F_{prop,ij} = \frac{F_{zij}}{F_{z11} + F_{z12} + F_{z21}} \tag{5.2}$$

The MPC is also, for every case, compared to when only the backup controller is

used for the same scenario. For these test cases, the scaling factor $\alpha$ for the target cost is set to 1, meaning that $\mathbf{P} = \mathbf{Q}$.

During the below-presented test cases, the approximated control invariant terminal constraint set was active in the MPC formulation, constraining the final state of the horizon $N$. The test cases were also simulated without the terminal constraint set activated, yet produced the same results. This will be discussed further in the Chapter 6.

All the following test cases are performed on a test road as in Fig. 5.6. It consists of a straight line of 20 meters, then a 10-meter-long transient followed by a constant radius $R$ curve.

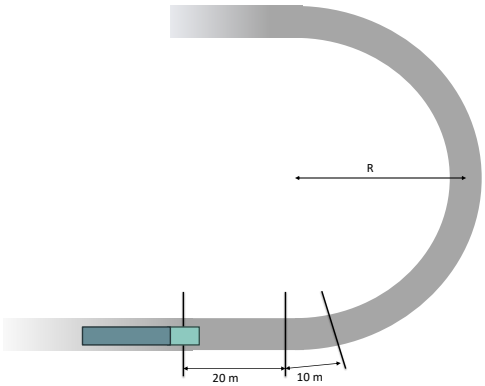
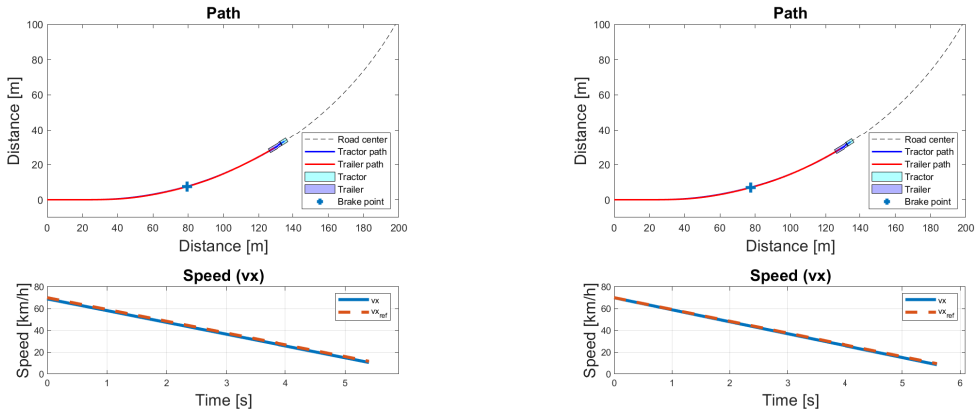



**Figure 5.6:** Illustration of the test road used during simulations of the MPC with the corresponding road parameters.

The tests are always initialised by having a path- and speed-follower that drives the truck into the curve before the controller starts. This is done because the VTM has some initial transients when initialised which could affect the results if the controller were to be started right away. The varying parameters between the test cases are the radius of the curve, initial velocity, friction and brake request.

### 5.2.1 Test case 1

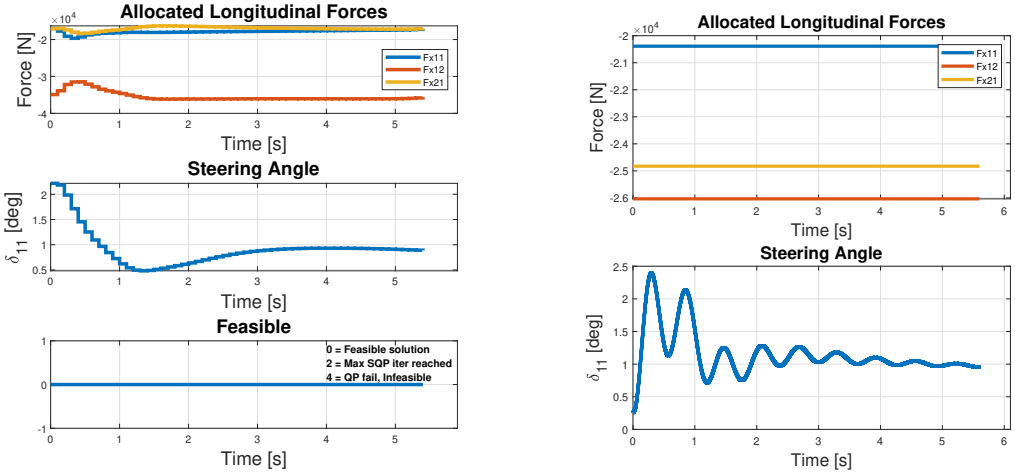
The first test case is done on a curve with a radius of 200 m, initial velocity of $70\,\mathrm{km\,h^{-1}}$, friction coefficient of 1 and a brake request of $-3\,\mathrm{m\,s^{-2}}$. This is a low curvature and high friction road and is a scenario that should not introduce any big instabilities. Figure 5.7 shows, for both controllers, the path taken by the vehicle together with the velocity and velocity reference. It can be seen that both controllers are very similar and both manage to stay centred on the road as well as following the braking reference.




**(a)** MPC **(b)** Proportional braking

**Figure 5.7:** Path and longitudinal velocity plots for both controllers, test case 1. The figures describe how the vehicle combination safely manages to stop giving the environmental parameters while following the given acceleration profile.

Fig. 5.8 shows the allocated longitudinal forces as well as the steering angle for the two controllers. Fig. 5.8a also shows if the MPC found a feasible solution or not for every given time step. The figure shows that the MPC is feasible for the entire simulation and that the braking allocation and steering differ from the proportional controller. The MPC solution brakes more with the driven axle ($F_{x12}$) than the proportional controller.

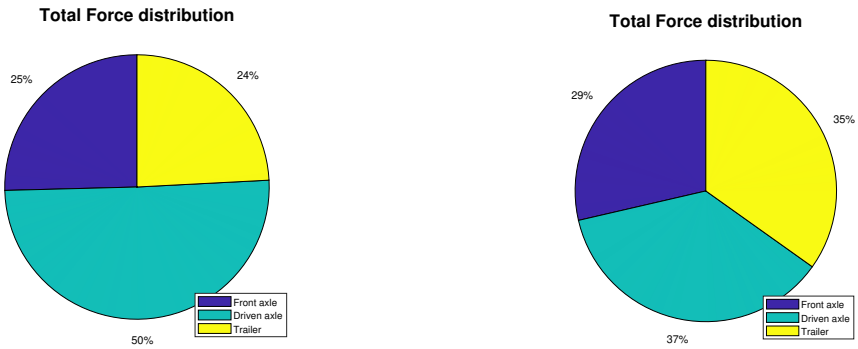



**(a)** MPC **(b)** Proportional braking

**Figure 5.8:** Allocated longitudinal forces, steering angle and feasibility for test case 1. The figure compares the allocated input of the MPC and the proportional controller.

Fig. 5.9 shows the total force distribution between the axles for the two controllers

throughout the simulation. It clearly shows that the MPC uses more of the driven axle than the proportional controller does.

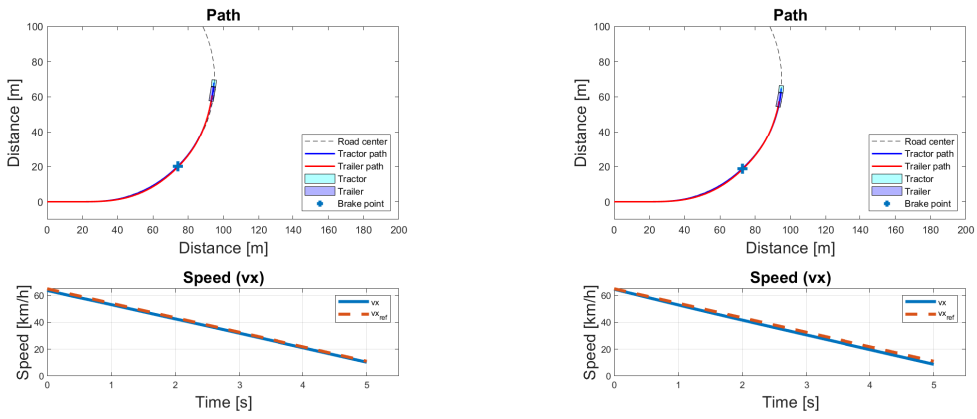



**(a)** MPC **(b)** Proportional braking

**Figure 5.9:** Total force distribution between the axles for test case 1. Driven axle usage is increased considerably more when using the MPC compared to the proportional controller.

### 5.2.2 Test case 2

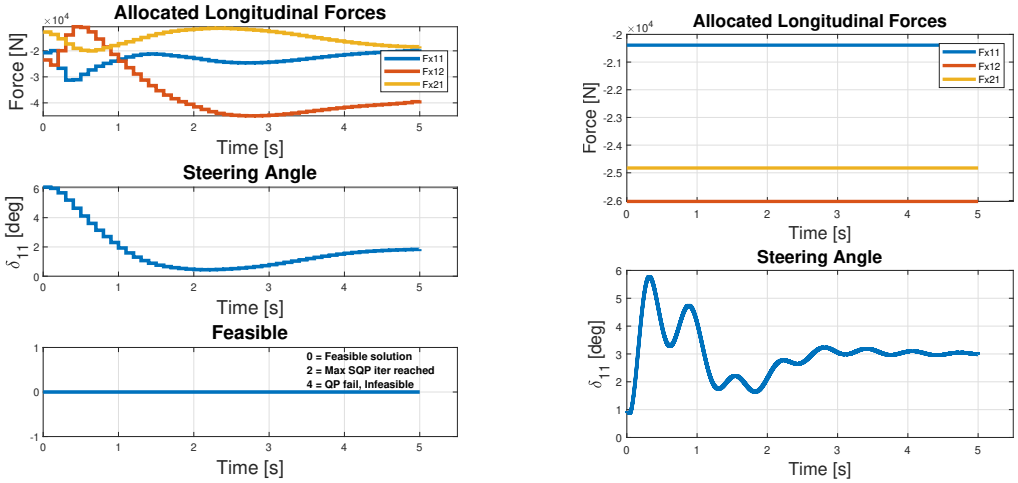
The second test case is done on a curve with a radius of 70 m, initial velocity of $65\,\mathrm{km\,h^{-1}}$, a friction coefficient of 1 and a brake request of $-3\,\mathrm{m\,s^{-2}}$. This is a higher curvature than the previous scenario and should be a bit more difficult to solve since the longitudinal and lateral forces will be higher and closer to the limits than in test case 1. Fig. 5.10 shows, for both controllers, the path taken by the vehicle together with the velocity and velocity reference. Similarly to test case 1, it can be seen that both controllers manage to stay centred on the road as well as follow the braking reference.




**(a)** MPC **(b)** Proportional braking

**Figure 5.10:** Path and longitudinal velocity plots for both controllers for test case 2. For this test case, the figures illustrate that the vehicle manages to maintain a safe vehicle motion while following the acceleration profile.

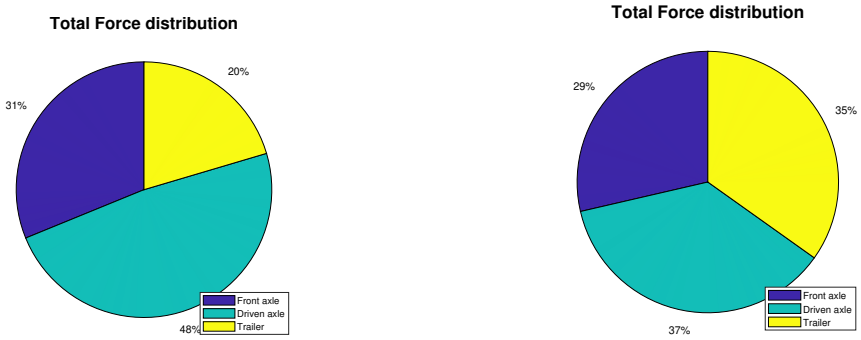
Fig. 5.11 shows that the MPC is feasible through the whole simulation in this test case as well. However, the force allocation graph shows that even though it prioritizes the driven axle, it cannot keep the allocation as much as in test case 1. It still uses the driven axle more in the MPC than the proportional controller.




**(a)** MPC **(b)** Proportional braking

**Figure 5.11:** Allocated longitudinal forces, steering angle and feasibility for test case 2. The comparison is made between the CA of the MPC and the proportional controller. One can see that a larger brake force is needed from the non-driven axles to maintain safe vehicle motion when using the MPC.

As seen in Fig. 5.12, the MPC utilizes the driven axle more than the proportional controller but can not keep the distribution as high as in test case 1.

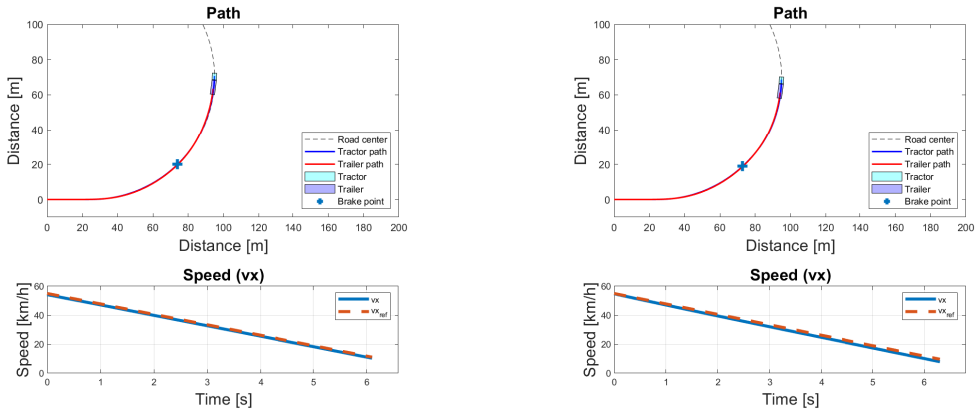



**(a)** MPC **(b)** Proportional braking

**Figure 5.12:** Total force distribution between the axles for test case 2. For this test case, the driven axle usage could be improved compared to the proportional controller with some compensation to ensure the stability of the vehicle.

### 5.2.3 Test case 3

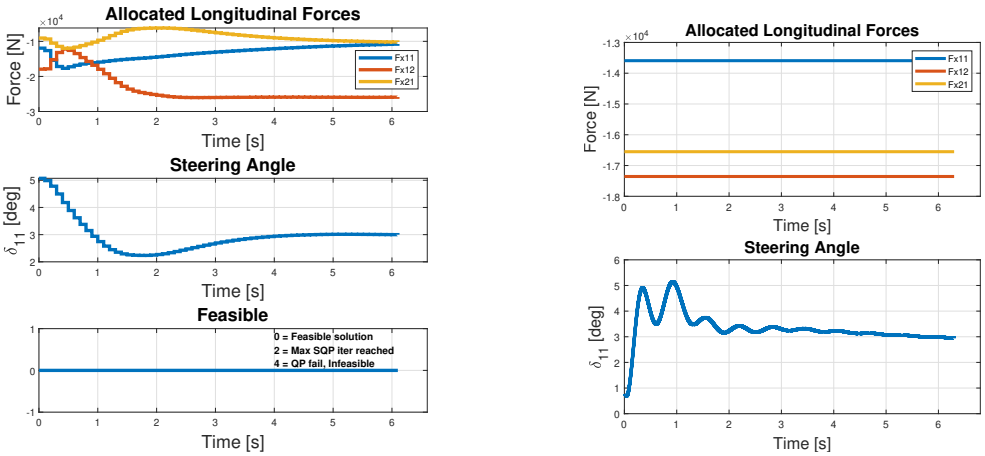
Test case 3 is performed on a curve with a radius of 70 m, initial velocity of 55 km h$^{-1}$, friction coefficient of 0.7 and a brake request of $-2$ m s$^{-2}$. This is an even riskier case than test case 2 since the friction is lower and the longitudinal and lateral forces will be even closer to the friction limits than in test case 2. Fig. 5.13 shows that just as in the previous test cases, both controllers manage to keep the vehicle centred on the road as well as following the braking reference.




**(a)** MPC **(b)** Proportional braking

**Figure 5.13:** Path and longitudinal velocity plots for both controllers for test case 3. The vehicle combination maintains safe motion while meeting the acceleration request during the procedure.

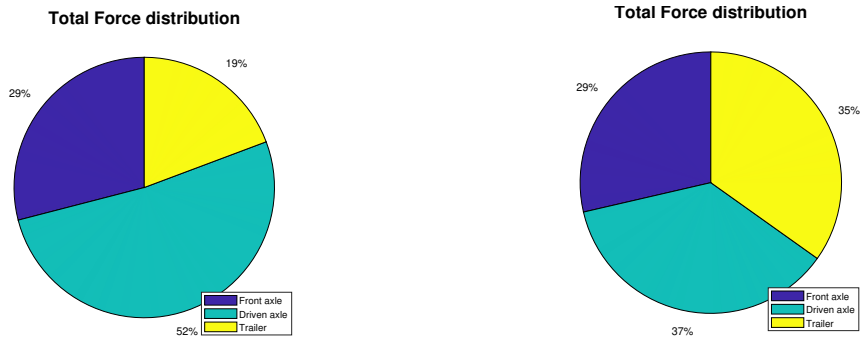
Fig. 5.14 shows that the MPC utilizes the driven axle more in this test case. This is because the MPC does not need to compensate as much as in test case 2 to maintain safe vehicle motion.




**(a)** MPC **(b)** Proportional braking

**Figure 5.14:** Allocated longitudinal forces, steering angle and feasibility for test case 3. The figures represent the CA for the MPC and the proportional controller.

In Fig. 5.15, it can be seen that also in this test case, the MPC utilizes more braking on the driven axle compared to the proportional controller.

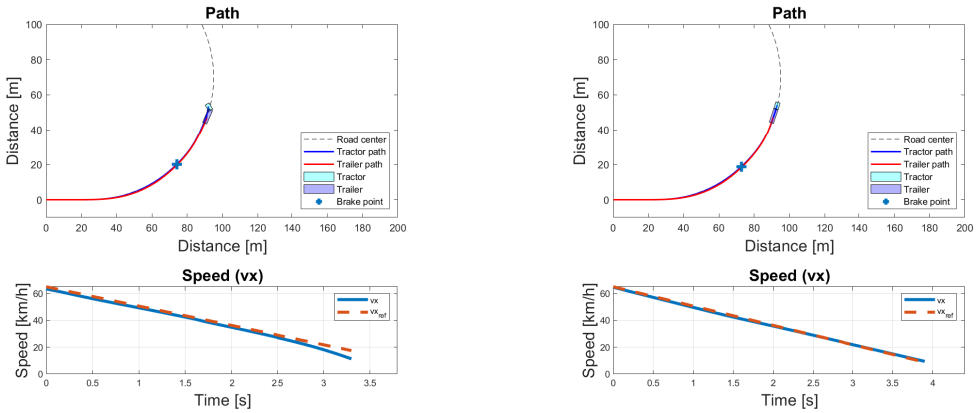



**(a)** MPC **(b)** Proportional braking

**Figure 5.15:** Total force distribution between the axles for test case 3. Utilization of the driven axle during the braking manoeuvre could be improved significantly compared to the proportional controller.

### 5.2.4 Test case 4

The fourth test case is on the same road and initial velocity as test case 2, a radius of 70 m, the initial velocity of $65\,\mathrm{km\,h^{-1}}$ and a friction coefficient of 1. However, the brake request is instead $-4\,\mathrm{m\,s^{-2}}$. This puts the vehicle in an even riskier case where the tyre forces are high. As can be seen in Fig. 5.16, the MPC fails at keeping the vehicle safe. The proportional controller however manages to keep it safe and shows that it is not an impossible scenario.

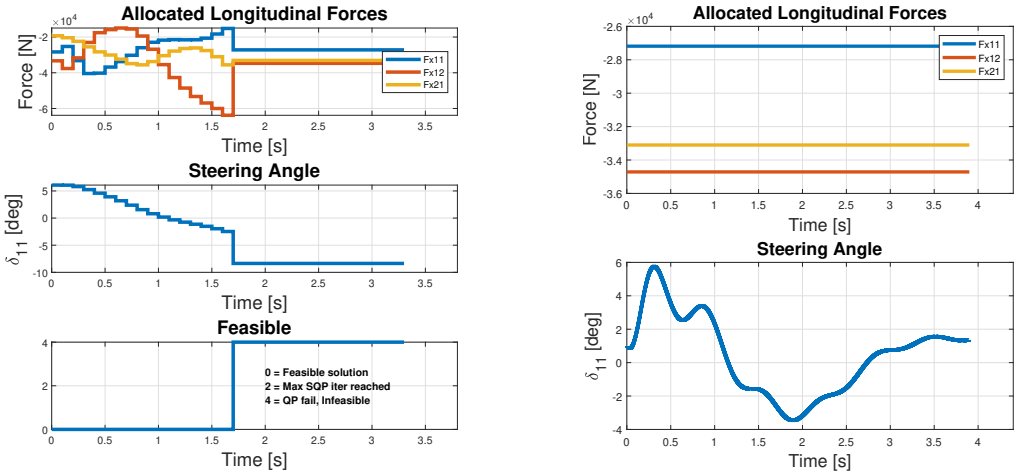



**(a)** MPC **(b)** Proportional braking

**Figure 5.16:** Path and longitudinal velocity plots for both controllers for test case 4. Here, the vehicle is not able to maintain a safe vehicle motion as can be seen from the path plot.

In Fig. 5.17, it can be seen that the MPC initially finds feasible solutions but that

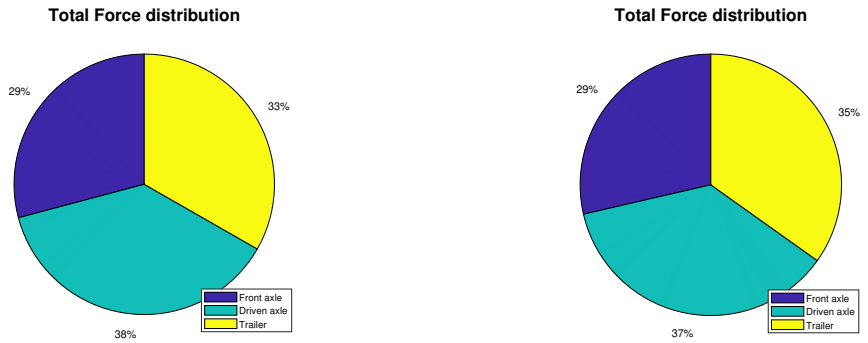
it later fails and that the proportional controller can not save the vehicle after that.




**(a)** MPC **(b)** Proportional braking

**Figure 5.17:** Allocated longitudinal forces, steering angle and feasibility for test case 4. The backup controller can be observed to kick in when the MPC becomes infeasible, which changes to force allocation to the same value as the proportional controller.

Fig. 5.18 shows that the MPC solution does not improve the driven axle usage since the backup controller kicks in, resulting in an overall similar CA as compared to the proportional controller.




**(a)** MPC **(b)** Proportional braking

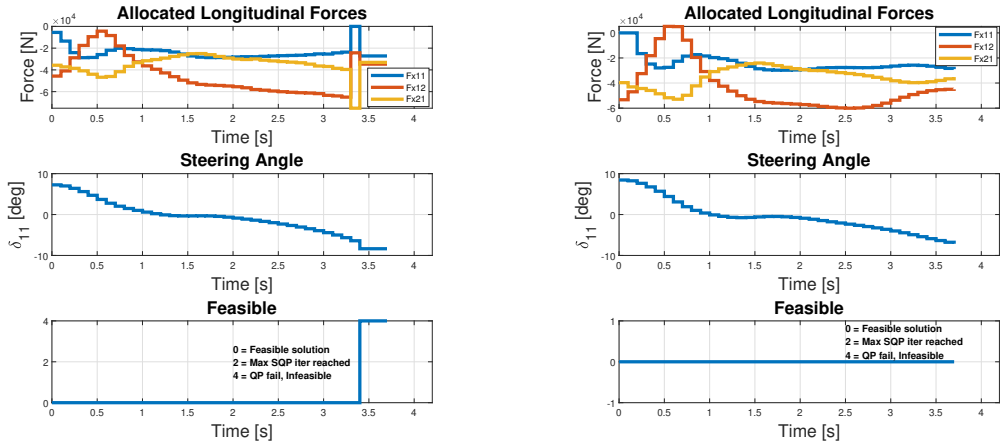
**Figure 5.18:** Total force distribution between the axles for test case 4. Only a small difference can be observed here due to the backup controller taking over.

## 5.3 MPC controlling the VTM with modified target cost

For the results in this section, the same four test cases as previously were performed, with the only difference being the tuning of the target cost scaling factor $\alpha$. It is

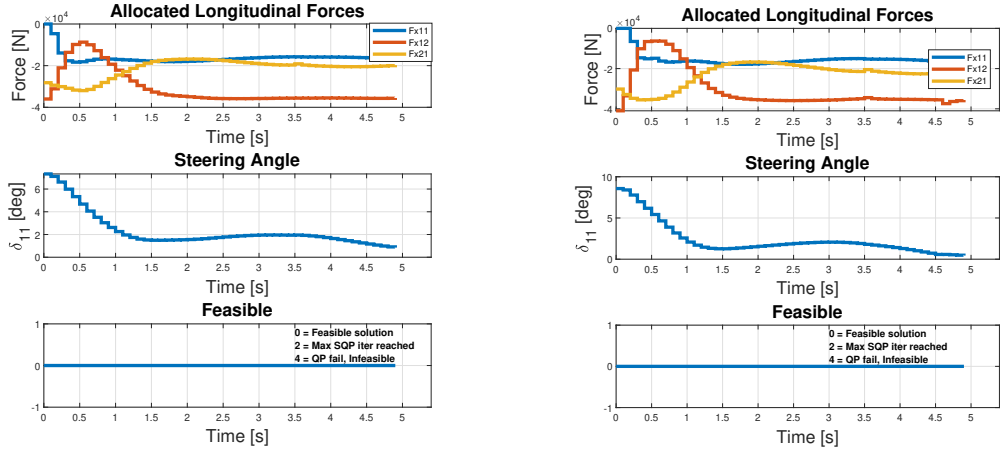
tested for both $\alpha = 5$ and $\alpha = 15$ to showcase the impact of the scaling factor. It means that the cost on the last state is either $\mathbf{P} = 5\mathbf{Q}$ or $\mathbf{P} = 15\mathbf{Q}$.

Fig. 5.19 shows the results from test case 4 and Fig. 5.20 from test case 2. It can be seen that a higher $\alpha$ contributes to this case becoming safe. If compared to the results in subsection 5.2.4, it can also be seen that selecting the scaling factor $\alpha$ to be 5 results in the MPC being feasible for a longer time, even though it eventually also jacknifes. Furthermore, selecting a higher $\alpha$ lowers the use of the driven axle which will be discussed further in Chapter 6.




(a) $\mathbf{P} = 5\mathbf{Q}$ (b) $\mathbf{P} = 15\mathbf{Q}$

**Figure 5.19:** Allocated longitudinal forces, steering angle and feasibility in test case 4 with different target cost $\mathbf{P}$. CA is performed differently for the values of the target cost which results in the target cost $\mathbf{P} = 5\mathbf{Q}$ being infeasible, while $\mathbf{P} = 15\mathbf{Q}$ produces a feasible solution.




(a) $\mathbf{P} = 5\mathbf{Q}$ (b) $\mathbf{P} = 15\mathbf{Q}$

**Figure 5.20:** Allocated longitudinal forces, steering angle and feasibility in test case 2 with different target cost $\mathbf{P}$. Independent of the value for $\alpha$, the MPC manages to find a feasible solution. When $\alpha = 15$, one can see that the MPC uses the trailer axle a bit more to improve stability.

In Fig. 5.20, it can be seen that just as in subsection 5.2.2, the MPC manages to solve the problem and control the VTM without ending up in an unsafe mode. It also shows that the vehicle starts by braking a bit harder with the trailer to stabilize the tractor but then converges to a similar solution as when $\alpha = 1$. When braking harder with the trailer than the tractor, the trailer stretches the vehicle combination, pulling on the tractor ahead of it. This can be used to avoid jack-knifing and can as mentioned be observed in Fig. 5.19 and Fig. 5.20.

The results for the remaining two test cases (1 and 3) are similar and can be seen in Appendix B. The path and speed plots for the two test cases presented in this section are also included in Appendix B.

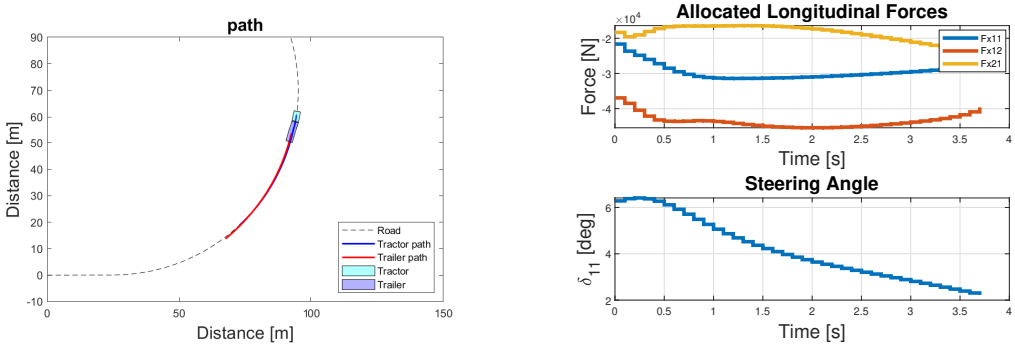
## 5.4 MPC controlling the single-track plant model

Another simulation was run to compare the investigated VTM simulations to a model similar to the model incorporated in the MPC and compare the test results. The MPC was operated on the single-track plant model presented in chapter 2 and the results can be seen in Tab. 5.1. The model was initialized both directly on the road (Single-track simulation) and in the same state as the VTM when the MPC was activated (Single-track with VTM initial state). This shows that the MPC finds a solution that is feasible according to the model integrated with the MPC, suggesting that the unsafe behaviour in test case 4 is due to a model mismatch rather than the formulated controller.

**Table 5.1:** Comparison between simulation results of controlling the single-track model and the VTM. The table presents whether the simulation resulted in a safe or unsafe vehicle mode for the 4 test cases and 3 simulation environments.

| Test Case: | Parameters | Singe-track simulation | Single-track with VTM initial state | VTM |
|---|---|---|---|---|
| **1:** | $R = 200$, $v_x = 70$, $ref_{acc} = -3$, $\mu = 1$ | Safe | Safe | Safe |
| **2** | $R = 70$, $v_x = 65$, $ref_{acc} = -3$, $\mu = 1$ | Safe | Safe | Safe |
| **3** | $R = 70$, $v_x = 55$, $ref_{acc} = -2$, $\mu = 0.7$ | Safe | Safe | Safe |
| **4** | $R = 70$, $v_x = 65$, $ref_{acc} = -4$, $\mu = 1$ | Safe | Safe | Unsafe |

In figure 5.21 the path and control signal can be seen when the MPC was used on the single-track model for test case 4 and with the initial state taken from the VTM simulation. These show how it manages to safely brake for the same scenario and how it prioritises the driven axle ($F_{x12}$) during the whole manoeuvre, something it doesn't do when in the VTM.

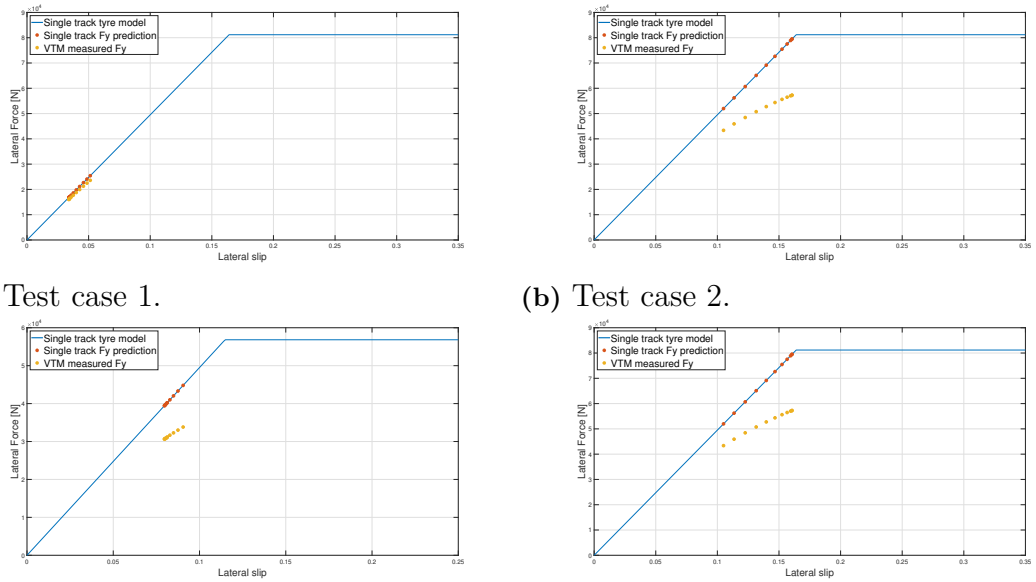



**(a)** Path **(b)** Forces and Steering

**Figure 5.21:** Path and inputs for MPC controlling the Single-track plant model for test case 4. Here one can see that safe vehicle motion is achieved using the MPC.

## 5.5 Tyre forces

In Fig. 5.22, the lateral tyre forces produced during each above-mentioned test case are presented. The figures show an increase in deviation between the predicted force production and the measured lateral force from the VTM. The divergence is observed to appear particularly when the tyre reaches the edge of the friction circle.




**(a)** Test case 1. **(b)** Test case 2.

**(c)** Test case 3. **(d)** Test case 4.

**Figure 5.22:** Lateral tyre forces during the test cases before initiating the MPC. For the test cases that can be considered "high-risk", one can observe that the deviation between the predicted lateral force and the measured lateral force from the VTM increases.

# 6

# Conclusions

This chapter analyzes the results of the thesis and discusses potential areas for improvement in future work on the subject. Overall, the solution provides a system that can perform control allocation within the safety boundary defined in the OCP and can retrieve an efficient solution during regenerative braking. However, in the provided solution, areas for improvement regarding safety are identified.

## 6.1 Terminal set approximation

The AL algorithm is effective in efficiently sampling the feature space and reducing the number of labelled samples needed to train a high-performing classifier. In low-dimensionality scenarios, the classifier shows high values in accuracy, precision, and recall.

As the number of features increases, the performance of the classifier decreases. In higher dimensions, the tradeoff between precision and accuracy or recall becomes more noticeable. As precision increases, the performance of the other two metrics declines quite quickly. As mentioned in Section 4.4.3 a high precision is favourable since it suggests that the approximated set is an inner approximation of the maximum control invariant set in this case. However, recall values that are too low suggest that the approximated set is much smaller than the maximum one, which in turn can lead to the MPC not finding solutions even when they exist because the optimization is constrained too much.

The reason for this behaviour could be due to a variety of factors. One reason could be that the Kernel of the SVM is not able to accurately approximate the shape of the actual set. This is very likely since, out of the complexity reason for the MPC, the chosen kernel was a 2-degree polynomial which is only able to create a hyper-ellipsoid in a multidimensional state space, limiting the shape in bounding the corners of the set as depicted in Fig. 4.6a. An inner approximation will furthermore result in a lower cost since sampling is performed around the prediction boundary in AL (where the predictor is the most uncertain). Since the area outside of the ellipse is larger than on the inside, generally more points will be sampled on the outside of the boundary. Thus, an inner approximation reduces the overall cost for the fitting

of the SVM.

Moreover, the MATLAB function used to train the classifier has limitations in high-dimensional spaces. The only applicable function for fitting a predictor during the thesis project using MATLAB was the fitcsvm function as described in Section 4.2. Therefore, an alternative solution could be sourced from a third-party provider to the MATLAB interface or by utilizing a different programming language, such as Python, to train the SVM to the labelled data set with the Python libraries for ML.

As presented in the results in Section 5.1, when sampling data in the 9-dimensional state space, around 25 per cent of the data points reach the maximum iteration limit of the SQP solver. This can influence the training of the AL algorithm since this can create gaps in the training data set calculated by the AL scheme. The MPC used as the oracle for the AL thus needs to be improved to reduce the number of points reaching the iteration limit. One improvement point can be implementing the same OCP in a different solver, such as Casadi using the IPOPT solver. One could investigate whether this oracle produces a similar number of maximum iteration limit breaches.

## 6.2 MPC

The MPC performs well for the CA of the defined vehicle model during most driving scenarios. Nevertheless, when the vehicle is simulated in high-risk cases where the produced tyre forces are near the friction limit and a large longitudinal force request is made to meet the defined acceleration profile, the system fails to find a feasible solution. This can be observed in test case 4 in section 5.2.4. This prompts the question of whether this is due to the controlled model or the defined MPC formulations with the terminal control invariant set. Firstly, when analysing the predicted terminal constraint set with regard to the results in 5.1, the precision score is high. This indicates that few infeasible points are included in the set as discussed, making it an inner approximation. Therefore, the set should not drive the system to a state where infeasibility will occur. This then suggests that the problem can be due to the MPC formulation.

As seen from simulating the same scenarios with the single-track plant model, safe vehicle motion could be achieved. This suggests that the problem may stem from the motion model not being accurate enough. The motion model used in the MPC has limitations to reduce computational complexity. To increase the performance of the MPC, changes to the vehicle model used in the MPC can be applied to increase performance at the cost of computational efficiency. A suggestion is to integrate a non-linear tyre model such as the Magic formula tyre model as mentioned in section 2.1.2 instead of the currently implemented friction circle model. As seen in section 2.4, the currently implemented tyre model produces too much lateral force in high-risk conditions. This in combination with high braking can probably be the reason for the unsafe mode exhibited in test case 4 in section 5.2.4.

Another interesting result is that switching to the backup controller after an infeasible presence in section 5.2.4 results in an unsafe mode, whilst proportional braking manages the same test case. A reason is that the switch from the MPC to the backup controller causes transients in the vehicle model. This can unsettle the system and lead to unsafe behaviour, especially in high-risk scenarios where the model can be sensitive to input shifts. Another alternative is that the system has reached a state where it is no longer possible to engage proportional braking to save the vehicle. The backup controller and the shifting condition thus need to be investigated more for increased robustness. It should also be noted that the current backup controller is just the proportional controller. What should be used instead is a more reactive controller that focuses on safety. This could for example be some kind of rule-based controller that activates when something unsafe is about to happen that tries to save the vehicle. For example, when a jack-knife is about to occur, a possible strategy would be to brake the trailer more than the tractor to stretch the combination, which would in that case be safer than proportional braking.

Adding the target cost $\mathbf{P}$ to the MPC formulation aids in increasing the stabilisation of the control scheme. By increasing the target cost, the system is driven back to the reference of the cost function at the end of the horizon, increasing the ability of the controller to stabilise and follow the setup deviation states. On the other hand, adding the target cost at the end of the horizon $N$ reduces the flexibility in CA, thus influencing the ability to enhance energy efficiency. In Section 5.3, the reduced driven-axle usage can be observed at the cost of improved stability, a decrease in the amount of regenerated power. Therefore, the target cost $\mathbf{P}$ is a parameter which can be tuned using $\alpha$, enabling further improvement by tuning. There are multiple ways of calculating the target cost $\mathbf{P}$, but this will not be further investigated in this thesis.

An addition can be made to the predicted terminal set constraint which can enable a more robust MPC formulation. Integrating data from the high-fidelity VTM in the training of the set predictor can incorporate both the differences between the models and the current control invariant terminal set to create a new set. This can prevent infeasibility beyond the horizon due to persistent feasibility issues and limit control signals to account for model mismatches.

## 6.3 Future Work

In all, the presented framework can be used flexibly. The control scheme that was implemented can be modified to suit different vehicle combinations, parameters, and objectives. Training multiple control invariant set approximations can create a library of sets which can be equipped by the controller accordingly, dependent on these factors. Below are presented some future areas of development which can further improve the proposed solution.

As future work to further improve the results presented in this thesis, some methods are proposed. One point of improvement is to implement a higher fidelity tyre model

in the MPC to get a better approximation of the lateral tyre forces in the vehicle model. This could improve the performance and behaviour of the vehicle model in the MPC which then further could improve the performance of the complete system. This would further enable the possibility of evaluating the approximated control invariant in-loop together with the MPC and the VTM.

The AL algorithm proposed in this thesis could also have applications elsewhere as a valid sampling technique for finding valuable data points to train a classifier or a set. In the introduction, the concepts of SOEs were mentioned to limit the allocated forces for a vehicle combination. Here, data is uniformly sampled for a variety of parameters which create a high-dimensional state space where data is to be classified as safe or unsafe. The concept of AL could also be implemented here to get a faster approximation of the SOE than using regular gridding techniques since several points are simulated which do not add new information to the approximation. This approach could be further investigated and readily implemented.

Another potential area for investigation is that by slightly adjusting the MPC, it can be repurposed for similar applications or issues. For example, it may not be preferable that the MPC controls the steering angle of the tractor. This is likely if the MPC were to be deployed in a non-autonomous vehicle combination. The objective would then be to only allocate the longitudinal forces since the manual driver would do the steering. This can be done in multiple ways, e.g. locking the steering (assuming constant steering or rate for the horizon), allowing small changes, or predicting the steering angle, etc.

Some additional ideas that could improve the overall developed system include implementing the higher degree polynomial SVM in AL to get a better approximation of the constraint set. The higher degree polynomial showed promising results concerning accuracy but will extend the computation time during the MPC optimization phase due to the increased amount of variables that these kernels contain. An analysis can be conducted to determine which variables could be removed to narrow down the approximated set, eliminating states that do not significantly influence the approximate boundary.

# Bibliography


[1] A. Hansson, E. Andersson, L. Laine, M. Sadeghi Kati, U. Erdinc, and M. Jonasson, "Safe operating envelope for limiting actuation of electric trailer in tractor-semitrailer combination," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3886–3893, 2022.

[2] K. Uhlén, P. Nyman, J. Eklöv, L. Laine, M. Sadeghi Kati, and J. Fredriksson, "Coordination of actuators for an a-double heavy vehicle combination using control allocation," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 641–648, 2014.

[3] U. Erdinc, M. Jonasson, M. Sadeghi Kati, B. Jacobson, J. Fredriksson, and L. Laine, "Modelling of articulated heavy vehicles for computation of accurate safe operating envelope for yaw stability," 11 2023.

[4] U. Erdinc, M. Jonasson, M. Sadeghi Kati, B. Jacobson, J. Fredriksson, and L. Laine, "Safe operating envelope based on a single-track model for yaw instability avoidance of articulated heavy vehicles," *Vehicle System Dynamics*, vol. 0, no. 0, pp. 1–24, 2023.

[5] P. Stano, U. Montanaro, D. Tavernini, M. Tufo, G. Fiengo, L. Novella, and A. Sorniotti, "Model predictive path tracking control for automated road vehicles: A review," *Annual Reviews in Control*, vol. 55, pp. 194–236, 2023.

[6] N. Chowdhri, L. Ferranti, F. S. Iribarren, and B. Shyrokau, "Integrated nonlinear model predictive control for automated driving," *Control Engineering Practice*, vol. 106, p. 104654, 2021.

[7] A. Norouzi, H. Heidarifar, H. Borhan, M. Shahbakhti, and C. R. Koch, "Integrating machine learning and model predictive control for automotive applications: A review and future directions," *Engineering Applications of Artificial Intelligence*, vol. 120, p. 105878, 2023.

[8] R. E. Allen, A. A. Clark, J. A. Starek, and M. Pavone, "A machine learning approach for real-time reachability analysis," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2202–2208, 2014.

[9] T. Ghandriz, B. Jacobson, P. Nilsson, L. Laine, and N. Fröjd, "Computationally efficient nonlinear one- and two-track models for multitrailer road vehicles," *IEEE Access*, vol. 8, pp. 203854–203875, 2020.

[10] Y. Xu, W. Tang, B. Chen, L. Qiu, and R. Yang, "A model predictive control with preview-follower theory algorithm for trajectory tracking control in autonomous vehicles," *Symmetry*, vol. 13, no. 3, 2021.

[11] J. L. Meriam and L. G. Kraige, *Engineering Mechanics: Dynamics*, ch. 7/6. Wiley & Sons, Incorporated, 2012.

[12] H. Pacejka and I. J. M. Besselink, *Tire and Vehicle Dynamics.* Elsevier Science & Technology, 2012.

[13] B. J. H. Jacobson, *Vehicle Dynamics Compendium*, ch. 2. Chalmers University of Technology, 2020.

[14] M. Sadeghi Kati, J. Fredriksson, B. Jacobson, and L. Laine, "A feedback-feedforward steering control strategy for improving lateral dynamics stability of an a-double vehicle at high speeds," *Vehicle System Dynamics*, vol. 60, no. 11, pp. 3955–3976, 2022.

[15] L. Gao, C. Beal, J. Mitrovich, and S. Brennan, "Vehicle model predictive trajectory tracking control with curvature and friction preview," *IFAC-PapersOnLine*, vol. 55, no. 24, pp. 221–226, 2022. 10th IFAC Symposium on Advances in Automotive Control AAC 2022.

[16] Trafikverket, *VGU, Supplement 1.* Trafikverkets publikationer ; 2022:167, Trafikverket, 2022.

[17] S. Janardhanan, L. Laine, M. Jonasson, B. Jacobson, and E. Gelso, "Motion control and power coordination of electric propulsion and braking distributed on multiple axles on heavy vehicles," in *2022 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1–8, 2022.

[18] J. Nan, X. Ye, and W. Cao, "Nonlinear model predictive control with terminal cost for autonomous vehicles trajectory follow," *Applied Sciences*, vol. 12, no. 22, 2022.

[19] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[20] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados: a modular open-source framework for fast embedded optimal control," 2020.

[21] J. B. Rawlings, D. Q. Mayne, M. Diehl, *et al.*, *Model predictive control: theory, computation, and design*, vol. 2. Nob Hill Publishing Madison, WI, 2017.

[22] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[23] E. Kerrigan and J. Maciejowski, "Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control," in *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*, vol. 5, pp. 4951–4956 vol.5, 2000.

[24] Z. Zhao, A. Girard, and S. Olaru, "Nonlinear model predictive control based on k-step control invariant sets," *European Journal of Control*, p. 101040, 2024.

[25] A. Chakrabarty, A. Raghunathan, S. Di Cairano, and C. Danielson, "Data-driven estimation of backward reachable and invariant sets for unmodeled systems via active learning," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 372–377, 2018.

[26] J. Kremer, K. Steenstrup Pedersen, and C. Igel, "Active learning with support vector machines," *WIREs Data Mining and Knowledge Discovery*, vol. 4, no. 4, pp. 313–326, 2014.

[27] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[28] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, 1992.

[29] MathWorks, "fitcsvm," 2024. Last accessed June 7th 2024.

[30] C. Campbell, N. Cristianini, A. Smola, *et al.*, "Query learning with large margin classifiers," in *ICML*, vol. 20, p. 0, 2000.

[31] D. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation," *J. Mach. Learn. Technol*, vol. 2, pp. 2229–3981, 01 2011.

# A

# Vehicle Paramaters

The presented vehicle parameters table A.1 is defined based on a Volvo FH ReVeRe 6x4 tractor and a semitrailer.

**Table A.1:** Vehicle parameters of the defined vehicle combination.

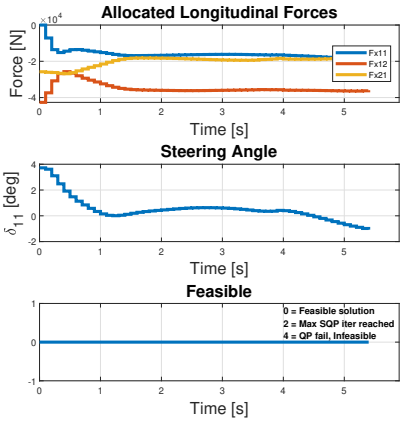
| Unit: | Truck | | | Trailer | | |
|---|---|---|---|---|---|---|
| Distance from CoG | ($a_1$) 1.5344 | ($b_1$) 2.5506 | ($c_1$) 2.2406 | ($a_2$) 5.1185 | ($b_2$) 3.2315 | |
| Wheel radii | 0.43 | 0.43 | 0.43 | 0.47 | 0.47 | 0.47 |

Further vehicle parameters are not included due to the confidentiality of the models by Volvo Group Trucks Technology.

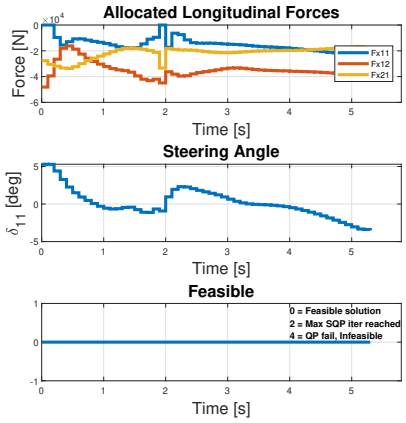
# B

# Results with modified target cost

## B.1 Test case 1

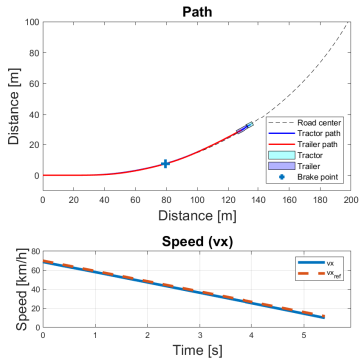



(a) $\mathbf{P} = 5\mathbf{Q}$

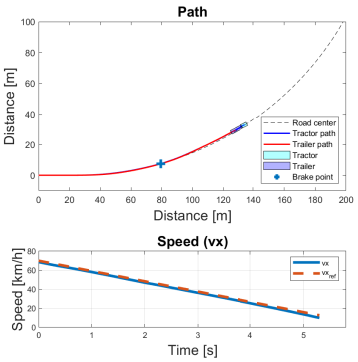



(b) $\mathbf{P} = 15\mathbf{Q}$

**Figure B.1:** Allocated longitudinal forces, steering angle and feasibility in test case 1 with different target costs $\mathbf{P}$



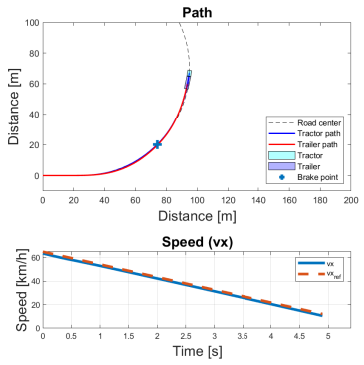

(a) $\mathbf{P} = 5\mathbf{Q}$



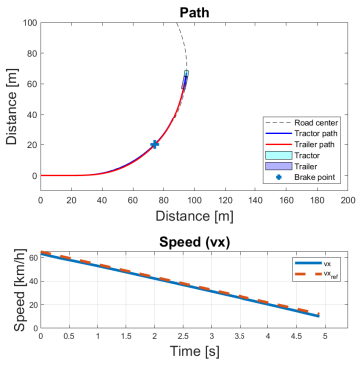

(b) $\mathbf{P} = 15\mathbf{Q}$

**Figure B.2:** Path and acceleration profile tracking for test case 1 with different target costs $\mathbf{P}$

## B.2 Test case 2



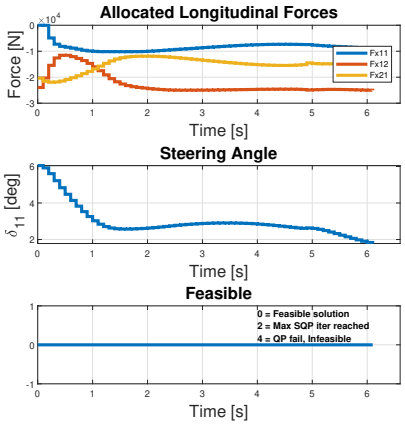

(a) $\mathbf{P} = 5\mathbf{Q}$



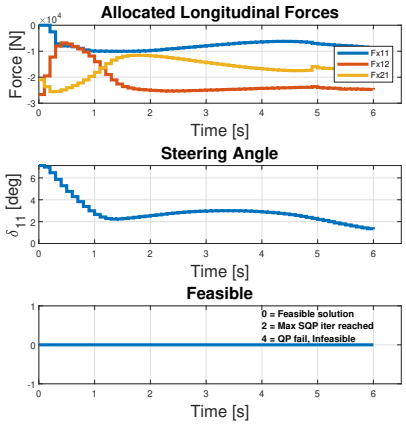

(b) $\mathbf{P} = 15\mathbf{Q}$

**Figure B.3:** Path and acceleration profile tracking for test case 2 with different target costs $\mathbf{P}$

## B.3 Test case 3

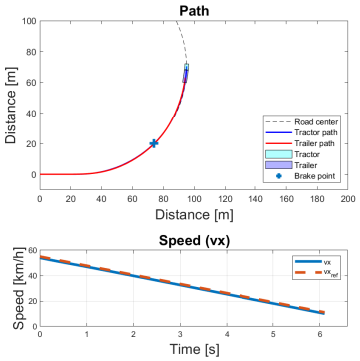



(a) $\mathbf{P} = 5\mathbf{Q}$

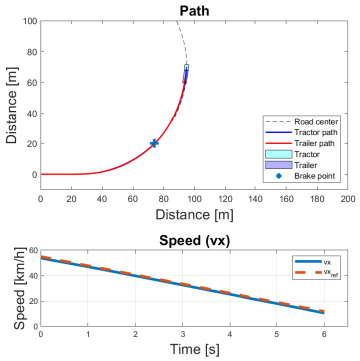



(b) $\mathbf{P} = 15\mathbf{Q}$

**Figure B.4:** Allocated longitudinal forces, steering angle and feasibility in test case 3 with different target costs $\mathbf{P}$



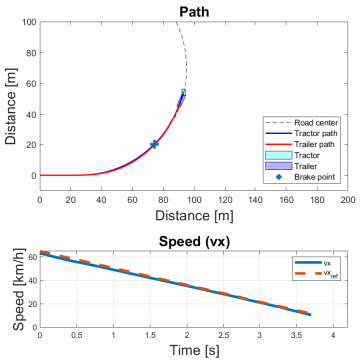

(a) $\mathbf{P} = 5\mathbf{Q}$



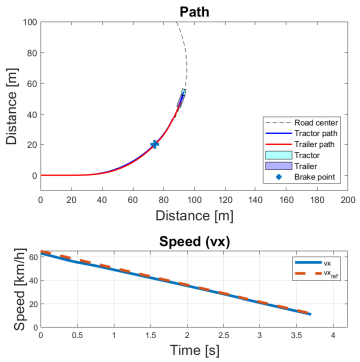

(b) $\mathbf{P} = 15\mathbf{Q}$

**Figure B.5:** Path and acceleration profile tracking for test case 3 with different target costs $\mathbf{P}$

## B.4 Test case 4




(a) $\mathbf{P} = 5\mathbf{Q}$




(b) $\mathbf{P} = 15\mathbf{Q}$

**Figure B.6:** Path and acceleration profile tracking for test case 4 with different target costs $\mathbf{P}$

**DEPARTMENT OF ELECTRICAL ENGINEERING**
**CHALMERS UNIVERSITY OF TECHNOLOGY**
Gothenburg, Sweden
www.chalmers.se


CHALMERS
UNIVERSITY OF TECHNOLOGY