MODEL PREDICTIVE CONTROL USING NEURAL NETWORKS

a Study on Platooning without Intervehicular Communications

KLAS LINDSTEN AND GUSTAV LING
Master of Science Thesis in Electrical Engineering

Model Predictive Control using Neural Networks - a Study on Platooning without Intervehicular Communications

Klas Lindsten and Gustav Ling
LiTH-ISY-EX--17/5079--SE

Supervisors:
Oskar Ljungqvist
ISY, Linköpings universitet
Christoffer Norén
Scania CV AB
Christian Larsson
Scania CV AB

Examiner: Johan Löfberg
ISY, Linköpings universitet

Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Copyright © 2017 Klas Lindsten and Gustav Ling
In the shadow of Evald
Abstract

As the greenhouse effect is an imminent concern, motivation for the development of energy efficient systems has grown fast. Today heavy-duty vehicles (HDVs) account for a growing part of the emissions from the vehicular transport sector. One way to reduce those emissions is by driving at short intervehicular distances in so called platoons, mainly on highways. In such formations, the aerodynamic drag is decreased which allows for more fuel efficient driving, meanwhile the roads are used more efficiently. This thesis deals with the question of how those platoons can be controlled without using communications between the involved HDVs.

In this thesis, artificial neural networks are designed and trained to predict the velocity profile for an HDV driving over a section of road where data on the topography are available. This information is used in a model predictive controller to control the HDV driving behind the truck for which the aforementioned prediction is made. By having accurate information about the upcoming behaviour of the preceding HDV, the controller can plan the velocity profile for the controlled HDV in a way which minimizes fuel consumption. To ensure fuel optimal performance, a state describing the mass of consumed fuel is derived and minimized in the controller. A system modelling gear shift dynamics is proposed to capture essential dynamics such as torque loss during shifting. The designed controller is able to predict and change between the three highest gears making it able to handle almost all highway platooning scenarios.

The prediction system shows great potential and is able to predict the velocity profile for different HDVs with an average error as low as 0.04 km/h. The controller is implemented in a simulation environment and results show that compared to a platoon without these predictions of the preceding HDV, the fuel consumption for the controlled HDV can be reduced by up to 6 %.
Acknowledgments

We would like to thank our supervisors at Scania, Christian Larsson and Christoffer Norén, and at Linköping University, Oskar Ljungqvist, for their guidance and support along the course of the project, and for the many meetings, ideas and laughs we’ve had together. We would also like to express our utmost gratitude to Johan Löfberg for his guidance and support in general, and for his outstanding work developing YALMIP, without which our work would not have been possible. We would also like to thank the natural born Bayesian and PhD-to-be Fredrik Ljungberg, the almost competition level eaters Ermin Kodzaga, Joakim Mörhed and Filip Östman, and the past Ginetta champion Nicanen, for valuable input, interesting discussions and for brightening dark days during the project. Finally we would like to express our gratitude to Filip & Fredrik, Peter Persson and Lars Monsen for making entertaining TV productions which we have enjoyed during the evenings in Södertälje.

Linköping, June 2017
Klas Lindsten and Gustav Ling
Contents

Abbreviations xi

1 Introduction 1
  1.1 Background .................................................. 1
  1.1.1 The ACC-LACC Platoon .................................. 2
  1.2 Problem Formulation and Purpose ............................. 3
  1.3 System Description ........................................ 3
  1.4 Delimitations ............................................... 4
  1.5 Thesis Outline ............................................ 4
  1.6 Related Work ............................................. 5

2 Modelling of Longitudinal Dynamics & Fuel Consumption 7
  2.1 Nonlinear HDV Model ...................................... 7
  2.1.1 Powertrain ............................................ 8
  2.1.2 External Forces .................................... 11
  2.1.3 Resulting Nonlinear HDV Model .......................... 12
  2.2 Nonlinear Two Vehicle Platoon Model .......................... 13
  2.2.1 Model Derivation ................................... 14
  2.3 Inclusion of a Mass of Consumed Fuel State .................. 15
  2.3.1 The Fuel Map .................................... 15
  2.4 Discretization and Linearization .............................. 18
  2.4.1 Resulting Discrete Linear Parameter Varying Model ...... 19

3 Gear Shift Modelling 21
  3.1 Modelling the Gear Shift Process ............................ 21
  3.2 Discretization and Linearization ............................. 22
  3.2.1 Resulting Discrete Linear Gearshift Model ................. 23

4 Piecewise Affine Approximation of System Dynamics 25
  4.1 PWA Vehicle Dynamics .................................... 25
  4.1.1 The Neutral Model ................................ 26
  4.1.2 Resulting Discrete Piecewise Affine System Approximation 27
  4.1.3 Example of Gearshift Sequence .......................... 27
### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC</td>
<td>Adaptive Cruise-Controller</td>
</tr>
<tr>
<td>CC</td>
<td>Cruise Controller</td>
</tr>
<tr>
<td>HDV</td>
<td>Heavy-Duty Vehicle</td>
</tr>
<tr>
<td>LACC</td>
<td>Look-Ahead Cruise Controller</td>
</tr>
<tr>
<td>LPV</td>
<td>Linear Parameter Varying (model)</td>
</tr>
<tr>
<td>LQ</td>
<td>Linear Quadratic (controller)</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multiple Input, Multiple Output (system)</td>
</tr>
<tr>
<td>MIP</td>
<td>Mixed Integer Program</td>
</tr>
<tr>
<td>MPC</td>
<td>Model Predictive Controller</td>
</tr>
<tr>
<td>NARX</td>
<td>Nonlinear Auto-Regressive eXogenous input</td>
</tr>
<tr>
<td>PID</td>
<td>Proportional, Integral, Differential (controller)</td>
</tr>
<tr>
<td>PWA</td>
<td>Piecewise Affine</td>
</tr>
<tr>
<td>V2V</td>
<td>Vehicle-to-Vehicle (communication)</td>
</tr>
</tbody>
</table>


1 Introduction

1.1 Background

In 2015, HDVs accounted for 30% of the total vehicular CO\textsubscript{2} emissions in the EU while representing only 5% of the vehicle fleet [14]. Scania CV AB stated in 2010 that one third of the costs related to a truck is derived from fuel [1]. These facts together with other benefits that come from energy efficient trucks have motivated the development of new systems and technologies. One such system is look-ahead cruise control (LACC), which is a control strategy utilizing information (primarily road slope) about the upcoming stretch of road to employ an economic manner of operation, using gravity in a beneficial way. Another one is driving HDVs in convoys, called platooning.

Platooning is the concept of driving two or more HDVs at short intervehicular distances. By doing this, a reduction in aerodynamic drag offers the possibility of reduced fuel consumption. With drag accounting for a significant portion of the resistance in HDV operation, the development of systems to enable safe operation in such formations have been the target of major research in the past decade.

Another benefit with platooning is the fact that the roads will be used more efficiently, preventing traffic congestion. This means that the traffic throughput will increase, which reduces pollutions and the time people spend in their vehicles. Platooning is also one major step towards autonomous vehicles.

A key concept in platooning is what is known as string stability. There have been several different definitions on what constitutes string stability but they all relate to whether disturbances affecting the leading HDV is amplified or attenuated as they propagate backwards in the platoon. At the time of writing this thesis there is no established communication standard that enables platooning with vehicle-to-vehicle communication (V2V), which is required to obtain string stable constant spacing platoons with more than two HDVs [22]. However, there
is an interest to investigate what performance can be achieved without V2V and a solution that can be implemented in the trucks of today. Many HDVs are today equipped with what is known as an adaptive cruise controller (ACC), employing a control strategy which, with the help of data from various sensors, aims to balance the (sometimes conflicting) goals of maintaining a constant velocity as well as a constant distance (or time headway gap) to a preceding vehicle.

Therefore, the benchmark in this thesis, will be in the form of an ACC-LACC platoon.

1.1.1 The ACC-LACC Platoon

Since string stability for longer platoons can not be guaranteed without V2V, concepts with two vehicle platoons have been developed. The ACC-LACC platoon consists of one HDV controlled with an LACC followed by an HDV controlled using ACC.

LACC is a feature that can be bought in today’s HDVs. The system uses information about upcoming road grade and accelerates the HDV before uphills and releases the throttle before downhills preventing the truck from using brake action to avoid violating the speed limit. A visualization of the LACC behaviour can be seen in Figure 1.1. The follower HDV uses an ACC, also a technology available in most modern HDVs and cars.

![Figure 1.1: An example of LACC behaviour, represented by the solid red line, compared to a conventional cruise controller, represented by the dashed line.](image)

To sum up, the ACC-LACC platoon is basically a lead truck that drives with a smart strategy to save fuel and a follower truck that tries to follow with a small set distance to the lead truck in order to reduce its aerodynamic drag. This strategy works well on roads with small inclinations and declinations if the trucks are somewhat equal in weight and performance. However, to motivate the need
for investigation of more advanced control systems we can look at the following example.

Consider an ACC-LACC platoon where the lead truck is significantly lighter compared to the follower truck. When the platoon reaches an uphill slope, the follower truck will likely decelerate more than the lead truck which leads to an increase in distance between the trucks and thereby a reduction in the platoon effect on the aerodynamic drag. When the same platoon reaches a downhill slope, the heavier follower truck will accelerate faster compared to the leading truck and will therefore need to brake to avoid a collision. These cases may be avoided if the follower truck has better knowledge of the expected behaviour of the lead truck and uses that information to control its velocity in advance.

### 1.2 Problem Formulation and Purpose

The main question this thesis work aims to answer is; what can be done to improve platooning performance in the case of no vehicle-to-vehicle communication? More precisely, can a controller be developed that uses models of both the controlled and preceding vehicle to make a given trip more fuel efficient than a conventional ACC-LACC platoon.

Throughout this thesis two different scenarios will be examined. The first, hereinafter referred to as the haulage scenario, depicts the case where a platoon is formed with two vehicles from the same haulage contractor. They drive a common road segment and plan the trip and driving configuration with respect to these vehicles in advance. This means that vehicle parameters such as mass and performance characteristics such as maximum engine power and active operational mode, that is ACC/LACC, is known beforehand. This information can thus be taken into consideration when ordering the vehicles and selecting models for the prediction of the preceding vehicle velocity. With this information, is it possible to utilize offline-developed speed predictor models of the leading vehicle in the controller of the follower vehicle to reduce fuel consumption?

The other scenario, catch-up and follow, consists of one HDV catching up with and deciding to follow an unknown HDV, forming a two vehicle platoon. In this case, is it possible to classify the vehicle in front and from a series of offline-developed velocity predictor models select one appropriate for use in the follower vehicle controller to achieve reduced fuel consumption?

### 1.3 System Description

This section describes the modules designed and implemented in this thesis. It should clarify the purpose of the different modules and their interconnections, respectively. An overview of the entire system is depicted in Figure 1.2 where blue modules are considered and developed in this thesis work. The data from sensors is considered filtered and ready for usage as is the road grade information. The designed controller sends reference signals to relevant control systems in the
vehicle, which means that it works as a higher level controller in a cascade control structure.

![Figure 1.2: An overview of the different modules involved in this thesis project. The ones in blue are designed in this work, the yellow ones are data from the vehicle and the purple ones are internal vehicle control systems.]

### 1.4 Delimitations

The most important delimitation in this thesis is the fact that the vehicles are assumed to have no means of communication between each other apart from radar measurements and camera data. The platoons are limited to configurations consisting of only two HDVs. The platoons are assumed to operate in absence of disturbances in the form of surrounding traffic.

### 1.5 Thesis Outline

In Chapter 2 the basic physical nonlinear vehicle and platoon models are derived. Also a mass of consumed fuel state is derived, which is to be used in the controller. The derived models are then linearized and discretized to yield a Linear Parameter Varying model for use in the controller.

Chapter 3 covers the gear shifting and models of its dynamics are developed.

In Chapter 4 different models for each gear are introduced to form a more accurate piecewise affine (PWA) model of the complete system.

In Chapter 5, the background, principles and methodology of the preceding vehicle velocity predictors are covered. A system for the correction of predictions based on measurements is proposed, along with a system to classify the preceding vehicle in order to select an appropriate velocity predictor.

Chapter 6 covers the derivation of the controller and describes its modifications along with implementation details.

Results from each subsystem as well as results from the complete system are gathered in Chapter 7, followed by discussion and conclusions in Chapter 8.
1.6 Related Work

The area of platooning is by no means an unexplored subject of research. A large number of theses and articles have been written, where different problems that arise in platooning have been considered.

Modelling of the Preceding HDV

When no V2V is utilized, a significant part of the problem is to predict the future behaviour of the preceding vehicle. Methods to estimate the mass and maximum power of the preceding vehicle have been developed, these however require following the vehicle to be modelled for some distance with non-flat sections of road to yield satisfactory performance. An example of such work can be found in [6]. Attempts have been made to develop grey-box as well as black-box models for prediction of velocity profiles of HDVs controlled by either conventional or look-ahead cruise controllers. Whereas the black-box models developed in this thesis are trained offline using data from simulations, these other works have focused on online estimation of model parameters [18].

Spacing Policies and Aerodynamic Drag

The two spacing policies mainly used and studied in platooning problems are constant spacing and constant time headway gap. Constant time headway gap yields a more energy efficient way of operation than does constant spacing according to [2]. There have also been some work on nonlinear spacing policies [2].

Most related works model the HDVs with the same dynamic model, a nonlinear model describing the motion of the truck under the influence of propelling and resisting forces. The key idea in platooning is the fact that the aerodynamic drag is significantly reduced for vehicles operating in a platooning fashion. The function describing the relationship between intervehicular distance, vehicle order in platoon and reduction in aerodynamic drag is based on wind tunnel measurements and similar data have been found in several independent experiments [2] [22].

Gear Shift Modelling

The inclusion of gear shifting in longitudinal vehicle models significantly add to their complexity, but is nonetheless desirable to describe properties such as the powertrain inertia and gear ratios being different among gears, and the absence of torque output to the powertrain during a shift. A common approach to the modeling of gear shift dynamics is to use a dynamic programming formulation, setting some period of time $\tau_{\text{shift}}$ with zero torque during a shift, such as in [12] and [11]. A mixed integer approach has been considered in [17], which does not however, model the period of time with torque loss during a gear shift.
Control Methods

Many different types of controllers have been proposed and studied for platooning with v2v, these include linear quadratic controllers and model predictive controllers, such as in [19] and [16]. There are a number of works on model predictive control platooning with the inclusion of road topographical data in the problem. Several types of platooning configurations, both with and without v2v, are considered in [7]. Model predictive controllers with different structures (centralized and decentralized) and different objective functions (linear and quadratic), are examined. The novelties of the model predictive control approach taken in this thesis consist of the development and weaving together of an augmented system description with gear shift dynamics as well as systems for the preceding vehicle speed profile prediction. Also, an objective function which allows for the balancing of the tendency to prefer a look-ahead strategy to remaining within a platooning distance of the preceding vehicle is proposed. Additional components is the preceding vehicle velocity predictor models as well as the accompanying correction and classification system.
In this chapter a nonlinear model of the longitudinal HDV dynamics is derived together with a nonlinear two vehicle platoon model. A model describing the mass of consumed fuel is also derived which explicitly describes how much fuel the vehicle will consume given certain parameters. The models are to be used for both control design and simpler simulations.

2.1 Nonlinear HDV Model

Today’s trucks are highly complex systems involving many actuators and subsystems with different dynamical properties. As always when modelling dynamical systems, a trade-off has to be made between model accuracy and model complexity. This thesis focus on longitudinal fuel efficient control of one HDV in a platoon and therefore fast dynamics are neglected. The main purpose of this section is to derive a model taking into account the subsystems relevant for the study of longitudinal HDV dynamics with respect to fuel consumption. The work in [2], [7] and [16] takes a similar approach as is done in this thesis for the modelling of the longitudinal vehicle dynamics.

The controller that is developed in this thesis work controls the vehicle in the longitudinal direction and thus no lateral motion is considered. The main forces contributing to the longitudinal dynamics of a HDV are depicted in Figure 2.1 and compiled in Table 2.1. The propelling force $F_p$ is produced by combustion of fuel in the engine and transferred via the powertrain to the wheels. This force is mainly used to control positive acceleration of the vehicle whereas the brake force $F_b$ is used to decelerate the vehicle.
Figure 2.1: The longitudinal forces acting on an HDV driving on a road with inclination $\alpha$.

Table 2.1: Explanation of longitudinal forces acting on the HDV depicted in Figure 2.1.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_p$</td>
<td>Propelling force</td>
</tr>
<tr>
<td>$F_b$</td>
<td>Brake force</td>
</tr>
<tr>
<td>$F_a$</td>
<td>Force due to aerodynamic drag</td>
</tr>
<tr>
<td>$F_g$</td>
<td>Force due to gravitation</td>
</tr>
<tr>
<td>$F_r$</td>
<td>Force due to rolling resistance</td>
</tr>
</tbody>
</table>

2.1.1 Powertrain

The powertrain of a vehicle is usually defined as the chain consisting of engine, clutch, gearbox, propeller shaft, final drive, drive shaft and wheels. A graphical representation of a powertrain can be seen in Figure 2.2 which also depicts the different effort and flow variables used for the modelling of one. When modelling the powertrain of the HDV, the propeller and the drive shafts are assumed to be stiff and are therefore left out of the figure.

Engine

Torque is produced in the engine by combustion of a mixture of fuel and air. It is then transferred through the connecting rods to the crank shaft and flywheel which connects with the clutch. In a real engine this is a very complex process involving fast dynamics and torque losses including pumping, friction and temperature variations. For the purpose of this thesis however, it is sufficient to model the engine as a rotating mass with known inertia. Euler’s laws of rigid body motion yield

$$J_e \frac{d\omega_e}{dt} = T_e(\omega_e, \delta) - T_c$$  \hspace{1cm} (2.1)

in which $T_e(\omega_e, \delta)$ represents the net torque produced by fuel combustion. $T_c$ is the external load from the clutch, the constant $J_e$ represents the engine and fly-
Figure 2.2: Simple model of an HDV powertrain with effort and flow variables for each subsystem except for propeller and drive shafts which are assumed to be stiff.

wheel mass moment of inertia and $\omega_e$ denotes the angular velocity of the engine flywheel. The amount of fuel injected to the cylinders measured in grams per second is represented by $\delta$.

**Clutch**

The clutch considered here is of classic frictional type which is often the case in trucks due to its high efficiency. It consists of two frictional disks which can be locked together and thereby transfer engine torque to the gearbox. Since the efficiency is high and the connection is assumed stiff, the clutch is simply modelled as

$$T_c = T_e \quad (2.2)$$

$$\omega_c = \omega_e \quad (2.3)$$

where $\omega_c$ denotes the rotational speed of the gearbox side clutch disk.

**Gearbox**

In the gearbox, torque is converted according to the gear ratio of the selected gear, which is modelled with a ratio $i_G$. The index $G$ is here used to denote the selected gear, which can be any element in the set of possible gears $G$. Power losses occur in the gearbox and are mainly due to friction. This is simply modelled as a constant efficiency $\eta_G$. Furthermore, the gears are assumed to have no inertia.
leading to the following static relationships

\[ T_G = i_G \eta_G T_e \]  
\[ i_G \omega_G = \omega_c \]  
\[ G \in G = \{ 0, 12, 13, 14 \} \]

where \( T_G \) and \( \omega_G \), respectively, denote the gearbox output side torque and rotational speed. The set of possible gears is here reduced to contain only the neutral gear and three out of the fourteen forward speed gears actually available in the real gearbox, the reason for which is made clear in Chapter 3 and 4.

**Final Drive**

The final drive is where the torque from the gearbox is transferred to the wheels. This is modeled in the same way as the gearbox with a conversion ratio \( i_{fd} \) and an efficiency \( \eta_{fd} \) resulting in

\[ T_{fd} = i_{fd} \eta_{fd} T_G \]  
\[ i_{fd} \omega_{fd} = \omega_G \]

where \( T_{fd} \) and \( \omega_{fd} \), respectively, denote the final drive output side torque and rotational speed.

**Wheels**

The final part in the chain is the wheels. They are modeled here as a rotational mass with radius \( r_w \), rotational speed \( \omega_w \) and mass moment of inertia \( J_w \) which, assuming no slip, yields

\[ J_w \frac{d\omega_w}{dt} = T_w - T_e - r_w F_p \]  
\[ v = \omega_w r_w = \frac{r_w}{i_G i_{fd}} \omega_c \]

where the driveshaft is assumed stiff, that is \( T_w = T_{fd} \). The propelling force is denoted by \( F_p \) and \( v \) denotes the vehicle speed.

**Resulting Powertrain Model**

Usage of equations (2.1) - (2.10) result in a model from input engine torque \( T_e \) to propelling force \( F_p \) as per

\[ F_p - F_b = \frac{i_{fd} \eta_{fd} i_G \eta_G}{r_w} T_e(\omega_c, \delta) - \frac{J_e i_{fd} \eta_{fd} i_G \eta_G^2}{r_w^2} \frac{dv}{dt} \]

where the coefficient of \( T_e \) is just a scaling dependent on the current gear and the coefficient of \( \frac{dv}{dt} \) is related to the system inertia. The variable \( F_b \) is used to denote the braking force.
2.1.2 External Forces

In this section the external forces depicted in Figure 2.1 are modelled and then put together with the powertrain model to arrive at the complete longitudinal HDV model.

Aerodynamic Drag

The main purpose of platooning is the potential reduction in fuel consumption due to the reduced aerodynamic drag. It is therefore important to have a model of the aerodynamic drag $F_a$ which takes this reduction into account. In Figure 2.3, adopted from [5], the reduction in aerodynamic drag is shown for one HDV driving behind another HDV at different intervehicular distances. It should also be mentioned that the very first vehicle in the platoon also experiences a reduction in aerodynamic drag, but of significantly smaller magnitude. The relationship is clearly nonlinear in distance and a nonlinear approximation

$$\tilde{\phi}_d(d) = C_1 \left(1 - \frac{C_2}{C_3 + d}\right),$$

(2.12)

is assumed to model this relationship.

![Data and Fitted Curve](image)

**Figure 2.3:** Measured air drag reduction data from [5] together with a nonlinear approximation.

The distance $d$ is defined as

$$d \triangleq s_p - l_p - s,$$

(2.13)
where \( l_p \) denotes the length of the preceding vehicle while \( s_p \) and \( s \) denote the (one-dimensional) positions of the preceding and following vehicles’ front surfaces along an arbitrary road, respectively. Figure 2.4 gives an illustration of these quantities. The design parameters \( C_1, C_2 \) and \( C_3 \) in (2.12) are found by means of regressing the measured data in Figure 2.3, which also depicts the resulting approximating function. The function approximates the measured data well in the region from 0 up to 60 meters, and then levels out leading to an air drag reduction coefficient of around 0.85 for all distances greater than 150 meters. This is however not a problem since this nonlinear approximation will be linearized at a distance of roughly 20 meters for use in the linear state space description. For simulation the measurement data points will be interpolated and for large distances the reduction coefficient will tend to 1. The model used to describe the resistive force due to aerodynamic drag while taking the platooning reduction into account is

\[
F_a = \frac{1}{2} c_D A_a \rho_a \tilde{\phi}_d(d)v^2
\]  

(2.14)

where \( c_D \) is a shape specific drag constant which is scaled with the frontal area of the truck \( A_a \). Further is \( \rho_a \) the density of air and \( \tilde{\phi}_d(d) \) is the drag reduction approximation (2.12). Lastly, \( v \) is the longitudinal velocity of the HDV through the air which in the case of no wind equals the velocity of the truck.

**Gravitation**

HDVs typically have very low specific power (power-to-weight ratio), which makes them sensitive to the longitudinal component of the gravitational force. According to [2], a typical 40 ton HDV will only be able to keep a set speed of 80 km/h on road inclinations up to 2.9 %. In comparison, Swedish highways can have sections as steep as 6-7 %. The force due to gravity is modelled as

\[
F_g = mg \sin \alpha
\]  

(2.15)

where \( m \) is the mass of the HDV, \( g \) is the gravitational constant and \( \alpha \) is the road inclination.

**Rolling Resistance**

The resistive force due to rolling resistance consists mainly of friction between the tire and the road and hysteresis in the tire. This is simply modelled as

\[
F_r = c_r mg \cos \alpha
\]  

(2.16)

where \( c_r \) is the rolling resistance coefficient.

**2.1.3 Resulting Nonlinear HDV Model**

Applying Newton’s second law of motion on the HDV depicted in Figure 2.1 yields the following model

\[
m \frac{dv}{dt} = F_p(T_r, G) - F_b - F_a(d, v) - F_g(\alpha(s)) - F_r(\alpha(s))
\]  

(2.17)
where the propelling force from the powertrain $F_p$ is described by (2.11) and the external forces are modelled in (2.14), (2.15) and (2.16). These relationships, along with (2.17), yield

$$m \frac{dv}{dt} = \frac{i_{fd} \eta_f \eta_d i_G \eta_G}{r_w} T_e(\omega_e, \delta) - \frac{J_e i_{fd} \eta_f i_G \eta_G^2 + J_w \frac{dv}{dt}}{r_w^2} - F_b - \frac{1}{2} c_D A_d \rho_d \tilde{\phi}_d(d)v^2 - mg \sin \alpha(s) - c_r mg \cos \alpha(s)$$

which can be rewritten using the following definitions

$$k_e(G) \triangleq \frac{i_{fd} \eta_f \eta_d i_G \eta_G}{r_w}$$

$$k_d(d) \triangleq \frac{1}{2} c_D A_d \rho_d \tilde{\phi}_d(d)$$

$$k_g \triangleq mg$$

$$k_r \triangleq c_r mg$$

as

$$\left( m + \frac{J_e i_{fd} \eta_f i_G \eta_G^2 + J_w}{r_w^2} \right) \frac{dv}{dt} = k_e(G) T_e(\omega_e, \delta) - F_b - k_d(d)v^2 - k_g \sin \alpha(s) - k_r \cos \alpha(s)$$

where $m_t$ is the total accelerated mass. For numerical values on the parameters in (2.18), refer to Table A.1 in Appendix A. In the resulting differential equation the slope of the road $\alpha$ is dependent on the position $s$ along the road.

### 2.2 Nonlinear Two Vehicle Platoon Model

This section outlines the derivation of the two vehicle platoon model serving as a basis for the controller design.

**Figure 2.4:** Conceptual visualization of a two vehicle platoon. Note that the distance between the HDVs is here represented as a time gap $\tau_{hw}$, defined by (2.27).
2.2.1 Model Derivation

Consider a two-vehicle platoon as shown in Figure 2.4, where the controlled truck is the follower. Since there is no vehicle-to-vehicle communication nor any ability to control the preceding vehicle, its velocity $v_p$ is consequently considered an exogenous input, as is the slope of the road $\alpha$. We define these exogenous inputs as

$$w \triangleq \begin{bmatrix} \alpha \\ v_p \end{bmatrix} \quad (2.24)$$

and the control signals as

$$u \triangleq \begin{bmatrix} T_e \\ F_b \end{bmatrix} \quad (2.25)$$

where again, $T_e$ is the engine torque and $F_b$ is the braking force. The platoon states are

$$x \triangleq \begin{bmatrix} v \\ \tau_{hw} \end{bmatrix} \quad (2.26)$$

where $v$ is the velocity of the controlled follower vehicle and $\tau_{hw}$ is the time headway gap to the preceding vehicle according to

$$\tau_{hw} \triangleq \frac{d}{v} = \frac{s_p - l_p - s}{v} \quad (2.27)$$

Since the length of the preceding vehicle $l_p$ is constant over time, the time derivative of (2.27) is

$$\frac{d\tau_{hw}}{dt} = \frac{d}{dt} \left( \frac{s_p - l_p - s}{v} \right) = \frac{(v_p - v)v - (s_p - l_p - s)\frac{dv}{dt}}{v^2} \quad (2.28)$$

With the definition of $\tau_{hw}$ according to (2.27), $s_p - l_p - s$ can be expressed as

$$s_p - l_p - s = v\tau_{hw} \quad (2.29)$$

Inserting (2.29) and (2.23) into (2.28) yields

$$\frac{d\tau_{hw}}{dt} = \frac{(v_p - v) - \tau_{hw} \frac{1}{m_{t(G)}} \left(k_e T_e - F_b - k_d (v\tau_{hw}) v^2 - k_g \sin \alpha(s) - k_r \cos \alpha(s) \right)}{v} \quad (2.30)$$

Since the disturbance $w$ will be dependent on position rather than time, the platoon state time derivatives are rewritten so as to yield a model with positional dependency

$$\frac{dx}{dt} = \frac{dx}{ds} \frac{ds}{dt} = v \frac{dx}{ds} \quad (2.31)$$
2.3 Inclusion of a Mass of Consumed Fuel State

With the following definitions,

\[
    f_{ld}(x, u, w, G) \equiv \begin{bmatrix} f_1(x, u, w, G) \\ f_2(x, u, w, G) \end{bmatrix} \equiv \begin{bmatrix} \frac{dv}{ds} \\ \frac{d^2v}{ds^2} \end{bmatrix},
\]  

(2.32)

the longitudinal model

\[
    f_{ld}(x, u, w, G) = \begin{bmatrix} 1 \\ \frac{1}{vm_{1}(G)} \left( k_e T_e - F_b - k_d(v \tau_{hw})v^2 - k_g \sin \alpha(s) - k_r \cos \alpha(s) \right) - \tau_{hw} \frac{1}{vm_{1}(G)} \left( k_e T_e - F_b - k_d(v \tau_{hw})v^2 - k_g \sin \alpha(s) - k_r \cos \alpha(s) \right) \right] \\
    \frac{1}{v^2} \end{bmatrix},
\]

(2.33)

is attained.

2.3 Inclusion of a Mass of Consumed Fuel State

When the model is later extended with different gear states, it becomes hard to explicitly tune for the minimization of fuel consumption. For example, a standard MPC formulation penalizing the control signals will motivate the lowest gear to be active since that mode requires less engine torque. This behaviour is obviously suboptimal and a more intuitive way of describing fuel consumption is desired. In this section, the model is extended with a mass of consumed fuel state which explicitly describes how much fuel the vehicle has consumed. In the controller, a minimization of this state at the end of a horizon should be an efficient way to tune for energy optimal performance.

2.3.1 The Fuel Map

Measured data of fuel flow for different engine speed and torque operating points is often represented in fuel maps. A generic fuel map is represented in Figure 2.5. The engine torque is already a control signal in the model (2.33), and the engine speed is a gear dependent scaling of the vehicle velocity which is an existing state. To make a linear approximation of a fuel map like the one depicted in Figure 2.5, one plane for each gear is fitted to the surface which, by transforming to positional dependency, results in the following nonlinear model for the evolution of the state describing mass of consumed fuel

\[
    \frac{dm_f}{ds} = C_1 \frac{i_G i_{fd}}{r_w} + \frac{C_2 T_e}{v} + \frac{C_3}{v},
\]

(2.34)

The mass of consumed fuel \( m_f \) is added as an additional state in the model. Equation (2.34) results in three planes, one for each gear, as depicted in Figure 2.6. From the figure it becomes clear that for most operating points, gear 14 (blue) yields the lowest fuel consumption. However, it should be noted that the highest gear is not feasible at all times of operation, this will later be modelled when the controller is derived. In Figure 2.7, one of the planes is plotted against the fuel map for validation and as can be seen, the plane approximates the map well in the region of interest.
Figure 2.5: A graphical representation of a generic fuel map.

Figure 2.6: The planes approximating the fuel maps, one for each gear.
2.3 Inclusion of a Mass of Consumed Fuel State

Figure 2.7: One of the fitted planes plotted together with the generic fuel map in the region of interest.
2.4 Discretization and Linearization

The modelling done in this chapter has resulted in nonlinear continuous models for longitudinal vehicle dynamics (2.33) and fuel consumption (2.34). For convenience the models are repeated here. The longitudinal vehicle dynamics as well as the mass of consumed fuel are modelled by

\[
\begin{bmatrix}
\frac{dv}{ds} \\
\frac{d\tau_{hw}}{ds} \\
\frac{df}{ds}
\end{bmatrix}
= \begin{bmatrix}
\frac{1}{vm_i(G)} (k_e T_c - F_b - k_d (v \tau_{hw}) v^2 - k_g \sin \alpha(s) - k_r \cos \alpha(s)) \\
(v_p - v) - \frac{1}{m_i(G)} (k_e T_c - F_b - k_d (v \tau_{hw}) v^2 - k_g \sin \alpha(s) - k_r \cos \alpha(s)) \\
C_1 \frac{(v^2 - F_b)}{\tau_w} + C_2 \frac{v}{v} T_e + C_3
\end{bmatrix}
\]

(2.35)

To be of any use designing and implementing controllers in a discrete linear MPC framework, the model (2.35) requires discretization and linearization. Discretizing the model using the Euler Forward method yields

\[
x_{k+1} = x_k + hf(x_k, u_k, w_k, G)
\]

(2.36)

where \( k \) is the discrete step, \( h \) the positional step size given by \( h = v_0 T_{\text{MPC}} \) where \( T_{\text{MPC}} \) denotes the MPC sampling time, and \( x_{k+1} = x((k+1)h) \). This discretization (2.36) of the longitudinal dynamics and fuel consumption is a nonlinear state-space model. We define

\[
\begin{aligned}
\bar{x}_{k+1} &\triangleq x_{k+1} - x_k - x_0 \\
\bar{u}_{k} &\triangleq u_{k} - u_0(G) \\
\bar{w}_{k} &\triangleq w_{k} - w_0
\end{aligned}
\]

Equation (2.37) is linearized using a first order Taylor expansion about a stationary linearization point \((x_0, u_0(G), w_0)\), for numerical values refer to Table A.2 in Appendix A. The stationary control signal \( u_0(G) \) is found by substituting \( x \) and \( w \) in (2.33) for suitable values of \( x_0 \) and \( w_0 \), fixing \( F_b \) to zero, inserting parameters corresponding to a specific gear \( G \), and then solving for \( T_e \). The first order Taylor expansion yields

\[
\begin{aligned}
\bar{x}_{k+1} &\approx \frac{\partial f_d(x_k, u_k, w_k, G)}{\partial x_k} \bigg|_{x_0, u_0(G), w_0} \bar{x}_k + \frac{\partial f_d(x_k, u_k, w_k, G)}{\partial u_k} \bigg|_{x_0, u_0(G), w_0} \bar{u}_k + \\
&\quad + \frac{\partial f_d(x_k, u_k, w_k, G)}{\partial w_k} \bigg|_{x_0, u_0(G), w_0} \bar{w}_k
\end{aligned}
\]

(2.38)

where

\[
\begin{aligned}
\bar{x}_k &\triangleq x_k - x_0 \\
\bar{u}_k &\triangleq u_k - u_0(G) \\
\bar{w}_k &\triangleq w_k - w_0
\end{aligned}
\]

With the following definitions of the state space matrices
2.4 Discretization and Linearization

\[ \mathcal{F}(x_0, u_0(G), w_0, G) = \left. \frac{\partial f_d(x_k, u_k, w_k, G)}{\partial x_k} \right|_{x_0, u_0(G), w_0} \]

(2.39)

\[ \mathcal{G}(x_0, u_0(G), w_0, G) = \left. \frac{\partial f_d(x_k, u_k, w_k, G)}{\partial u_k} \right|_{x_0, u_0(G), w_0} \]

(2.40)

\[ \mathcal{J}(x_0, u_0(G), w_0, G) = \left. \frac{\partial f_d(x_k, u_k, w_k, G)}{\partial w_k} \right|_{x_0, u_0(G), w_0} \]

(2.41)

### 2.4.1 Resulting Discrete Linear Parameter Varying Model

Summing up equations (2.38) - (2.41), we get the following model for the longitudinal dynamics and fuel consumption which is, for fixed \( x_0 \) and \( w_0 \), linear parameter varying (LPV) with respect to the gear \( G \)

\[
\begin{align*}
\dot{x}_{k+1} &= \mathcal{F}(x_0, u_0(G), w_0, G)\tilde{x}_k + \mathcal{G}(x_0, u_0(G), w_0, G)\tilde{u}_k + \\
&\quad + \mathcal{J}(x_0, u_0(G), w_0, G)\tilde{w}_k \\
y_k &= x_k \\
x &\in \mathcal{X} \\
u &\in \mathcal{U} \\
w &\in \mathcal{W}
\end{align*}
\]

(2.42)

where \( \mathcal{X}, \mathcal{U} \) and \( \mathcal{W} \), respectively, denote the constraints on states, control signals and disturbances. The constraints on the states and control signals are dependent on the engaged gear, \( G \). The state constraints consist of the time headway limitations and velocity limitations. The time headway is constrained with a minimum and maximum value. The vehicle velocity is globally (meaning for all gears) constrained to a minimum and maximum value corresponding to the road speed limits. Constraints on velocity specific to each value of \( G \) are the engine speed limitations, which consist of minimum and maximum engine speeds converted into corresponding vehicle velocities for that particular gear.

The control signal constraints consist of minimum and maximum engine torque and braking force. The maximum engine torque is a function of engine speed and since the conversion ratio between engine speed and vehicle velocity is different among gears, the constraints modelling this are different for each gear. The constraints on the disturbances consist of minimum and maximum values of the road slope and preceding vehicle velocity.

Summing up, the sets \( \mathcal{X}, \mathcal{U} \) and \( \mathcal{W} \) are, for a given gear, all convex polytopic subsets of \( \mathbb{R}^3, \mathbb{R}^2 \) and \( \mathbb{R}^2 \), respectively. A set is said to be convex if a line connecting any two points in the set is also completely contained within the set. A set is said to be polytopic if it can be defined using linear inequalities [13].
In this chapter, models are derived to account for the transitional phase when shifting between gears. Earlier work incorporating gear shift dynamics in the model have mainly been formulated in a dynamic programming framework [12]. An alternative approach to the shift dynamics problem is presented. By introducing pairs of states and control signals governed by first order system dynamics, the gearshift process can be modelled.

**Motivation of Gear Shift Modelling**

By now it has been made clear that the relative great mass of HDVs exposes them to lots of resistance when accelerating or driving uphill. To cope with this the HDVs are equipped with strong but low speed engines. Their upper engine speed limit is typically in the range of 2000 rpm, meanwhile a common passenger car may go up to 5000 rpm. This low engine speed limit introduces the need for more gears in the gearbox. The truck considered throughout this thesis has a 14-speed gearbox. However, since this thesis work considers platooning, which is mainly efficient at higher velocities, there is no need to model all fourteen gears in this control framework. In this work the three highest gears are considered, thus covering most platooning situations.

### 3.1 Modelling the Gear Shift Process

What is here referred to as the gear shift process is the transition times between gears. When shifting between gears, there is a period of time with open clutch during which no propulsive torque is transmitted from the engine to the powertrain. This state of zero torque output will be referred to as the *neutral* state and will be modelled using what we have chosen to call the *neutral model*, explained
later on. For vehicles with a low power-to-weight ratio in a steep uphill, such as HDVs, this brief period without propulsion can have a significant impact on its velocity. By modelling the gear shift process, these effects can be taken into account in the controller.

To model the gear shift process of the three highest gears, we define

\[
\begin{align*}
x_g & \triangleq \begin{bmatrix} g_{12} \\ g_{13} \\ g_{14} \end{bmatrix} \\
u_g & \triangleq \begin{bmatrix} b_{12} \\ b_{13} \\ b_{14} \end{bmatrix}
\end{align*}
\]

where

\[
x_g \in \mathcal{X}_g = \{ x \in \mathbb{R}^3 : 0 \leq x_i \leq 1 \}
\]

\[
u_g \in \mathcal{U}_g = \begin{cases} 
\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, & 
\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, & 
\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{cases}
\]

The gear shift control signals \( b_i \) in \( u_g \) are binary, while the gear states \( g_i \) in \( x_g \) are continuous but designed to stay between 0 and 1. Note that only one gear shift control signal can be 1 at a time, since a decision to engage two gears simultaneously would make no sense. The gear shift is modeled as

\[
\dot{x}_g = \frac{1}{\tau_g} \left( -x_g + u_g \right)
\]

where \( \tau_g \) is the gear shift time constant. Model (3.5) is time dependent, to fit with (2.35) it needs to be made dependent on position. Again, utilizing the chain rule

\[
\frac{dx_g}{dt} = \frac{dx_g}{ds} \frac{ds}{dt} = v \frac{dx_g}{ds}
\]

results in the following gearshift model with positional dependency

\[
\frac{dx_g}{ds} = \frac{1}{v \tau_g} \left( -x_g + u_g \right) \triangleq g(x, x_g, u_g)
\]

The idea is then to threshold the gear state vector components by some value \( g_{th} \) to describe which gear is engaged. The threshold \( g_{th} \) and the time constant \( \tau_g \) are adjusted to mimic the real system with respect to the time (or rather distance) spent between gears during a shift.

### 3.2 Discretization and Linearization

Discretizing model (3.7) using Euler Forward yields

\[
x_{g,k+1} = x_{g,k} + h g(x_k, x_{g,k}, u_{g,k})
\]
where $k$ is the discrete step, $h$ the positional step size given by $h = v_0 T_{\text{MPC}}^s$, and $x_{k+1}^g = x((k + 1)h)$. Using the following approximation

$$v_k \approx v_0, \quad k = 1, \ldots, N$$  \hfill (3.9)

the discretized model (3.8) for the gear dynamics can be simplified to

$$x_{g,k+1}^g = x_{g,k}^g + \frac{T_{\text{MPC}}^s}{\tau_g} \left( -x_{g,k}^g + u_{g,k}^g \right)$$  \hfill (3.10)

which is a discrete linear state-space model, independent of the other states in the system.

### 3.2.1 Resulting Discrete Linear Gearshift Model

Summing up (3.3), (3.4) and (3.10), we get the discrete linear model

$$x_{g,k+1}^g = x_{g,k}^g + \frac{T_{\text{MPC}}^s}{\tau_g} \left( -x_{g,k}^g + u_{g,k}^g \right)$$  \hfill (3.11)

where $X_g$ is a convex polytopic subset of $\mathbb{R}^3$, while $U_g$ is a subset of $\mathbb{B}^3$ (the set of three dimensional binary vectors), thus nonconvex.
Piecewise Affine Approximation of System Dynamics

As can be noticed by looking at many of the equations in Section 2.4, the selected gear $G$ influences the dynamical properties of the models. This leads to the resulting model (2.42) being (for fixed $x_0$ and $w_0$) LPV in the gear $G$. Since only one gear can be engaged at a time, a specific affine model can be assigned to each gear resulting in a piecewise affine (PWA) system approximation as exemplified in Figure 4.1.

4.1 PWA Vehicle Dynamics

With the thresholding of the gear states $x_g = [g_{12} g_{13} g_{14}]^T$ described in Section 3.1, we achieve a partitioning of the state space of the gear states. We then let each partition correspond to a specific gear being engaged, as in

$$G = \begin{cases} 
12 & \text{if } g_{12} \geq g_{th} \\
13 & \text{if } g_{13} \geq g_{th} \\
14 & \text{if } g_{14} \geq g_{th} \\
0 & \text{else} 
\end{cases} \quad (4.1)$$

$g_{th} \equiv 0.8$

The partitioning (4.1) effectively results in a switching system consisting of 4 different models, each a gear specific version of (2.42). Together they approximate the nonlinear gear dependent system analogously with that of the example in Figure 4.1. Note that the neutral model is active when all gear states are below their threshold values and the torque output is then zero. To sum up, the fundamental idea is that each gear has its own dynamical model and during shifting a neutral model is active which captures the time of zero torque output to the powertrain. The linearization points for the models coincide except for the engine
torque which is affected by the selected gear resulting in a unique linearization torque for each model as described in Section 2.4.1.

\[
f(x) \approx \begin{cases} 
    \text{Model1}(x) & \text{if } x \in A \\
    \text{Model2}(x) & \text{if } x \in B \\
    \text{Model3}(x) & \text{if } x \in C 
\end{cases}
\]

![PWA System Approximation](image)

**Figure 4.1:** An example of how several affine models can approximate a broader region of a nonlinear function compared to a single linear model.

### 4.1.1 The Neutral Model

The model accounting for the case when \( G = 0 \), is used to model the dynamics during a gear shift. The neutral model is achieved simply by setting \( i_G = 0 \) which affects (2.19) as

\[
k_c(0) = 0
\]

this effectively leads to a model where torque cannot be used to control the states, and the accelerated mass in (2.23) becomes somewhat smaller. As an effect of this, the system loses controllability and

\[
\text{rank } S_{\text{neutral}} < \text{rank } S_{\text{gear}}
\]
where $S$ denotes the controllability matrix of the respective models. This is, however, not a problem but rather the purpose of the neutral model. This works analogously to a situation where the neutral gear is engaged in a car, during that time the velocity cannot be controlled using the throttle.

### 4.1.2 Resulting Discrete Piecewise Affine System Approximation

Combining (2.42), (3.11) and (4.1) result in the following discrete piecewise affine model

$$
\begin{align*}
\bar{x}_{k+1} &= \mathcal{F}(x_0, u_0(G), w_0, G)\bar{x}_k + \mathcal{G}(x_0, u_0(G), w_0, G)\bar{u}_k + \mathcal{J}(x_0, u_0(G), w_0, G)\bar{w}_k \\
y_k &= x_k \\
x &\in \mathcal{X} \\
u &\in \mathcal{U} \\
w &\in \mathcal{W} \\
\bar{x}_{g,k+1} &= x_{g,k} + \frac{T_{g}^{\text{MPC}}}{T_g} \left( -x_{g,k} + u_{g,k} \right) \\
x_g &\in \mathcal{X}_g \\
u_g &\in \mathcal{U}_g \\
G &= \begin{cases} 
12 & \text{if } g_{12} \geq g_{\text{th}} \\
13 & \text{if } g_{13} \geq g_{\text{th}} \\
14 & \text{if } g_{14} \geq g_{\text{th}} \\
0 & \text{else}
\end{cases}
\end{align*}
$$

### 4.1.3 Example of Gearshift Sequence

To demonstrate how the gear shifts affect the system dynamics, an exemplified shift procedure has been made and can be seen in Figure 4.2. For complete understanding, look at the figure meanwhile referring to (4.1) as the subplots show each discrete gear state.

At step 0 of the simulation, gear 2 is engaged and the binary control signal for gear 1 is active. This means that during the next step gear 2 will start to disengage and gear 1 start to engage. During this period of time, between step 1 and 2, the neutral model is active and the output torque on the wheels is thus zero. In step 3, gear 1 has reached above the threshold and becomes active for two steps, and so on.

Note that it takes two steps for a gear state to become active, and one step to become passive. This means that the neutral model always will be active during one step between gear shifts given that the binary control signals always sum to 1.
Figure 4.2: A simulated sampled gear change procedure. Red line represents the binary control signal and the blue line the continuous gear state. Red area indicates active gear and blue area indicates active neutral.
When modelling based on physical relationships between the quantities of interest is deemed infeasible, black-box modelling provides an alternative framework. The goal is to infer a model by using measurements of the inputs and outputs of the system or process in question [20]. The measurement data is split into three sets, one training set, one validation set and one test set. By fitting generic models to training data and then propagating the validation input data through the models and computing a performance measure with respect to the validation output data, a suitable model can be inferred. This model is then evaluated on the test data set.

5.1 Problem

The problem of interest is to, given past and future road grade data as well as past velocities predict the velocity profile of a preceding vehicle up to a given horizon (in distance). The only thing assumed to be known about the preceding vehicle is that it operates using some sort of cruise controller. Thus, an important property of the velocity profile prediction system is robustness with respect to cruise controller types, operating points (set speeds) and power-to-weight ratios of the subject vehicle. For an illustration of the problem of one step ahead prediction of the preceding vehicle velocity, see Figure 5.2.

5.2 Supervised Learning

Supervised learning is a branch of machine learning concerned with inferring models from a set of known pairs of inputs and outputs. Suppose the system or
process we are modelling have input vectors on the form

\[ X = [x_1, \ldots, x_n]^T \]  

(5.1)

and that these \( X \) belong to the following input space constituting all possible inputs

\[ X \in \mathbb{I} \subseteq \mathbb{R}^n_x \]  

(5.2)

Additionally, the system have output vectors on the following form

\[ Y = [y_1, \ldots, y_n]^T \]  

(5.3)

belonging to the output space \( \mathbb{O} \), which is the image of the input space under the system, let's denote it \( \mathcal{S} \), as in

\[ \mathcal{S} : \mathbb{I} \rightarrow \mathbb{O} \]  

(5.4)

Thus, the following is true for the outputs and output space

\[ Y \in \mathbb{O} \subseteq \mathbb{R}^n_y \]  

(5.5)

The data which is used for training, validating and testing a model then consists of pairs on the following form

\[ \mathbb{D} = \{(X_1, Y_1), \ldots, (X_N, Y_N)\} \]  

(5.6)

where \( N \) denotes the number of examples. One example thereby contains information about the mapping we wish to infer, let's denote it \( f \), under which the image of any given \( X \) in the input space should be the same as under the system \( \mathcal{S} \). The data \( \mathbb{D} \) is usually split into three disjoint sets, one called training data, which is directly used for optimization of model parameters, the second called validation data, which is used to prevent overfitting during training, and finally test data, which serves as an independent set of data with which to verify that the inferred function \( f \) describes the mapping in a satisfactory manner [9]. Thus, we have

\[ \mathbb{D}_{train} \subset \mathbb{D} \]  

(5.7)

\[ \mathbb{D}_{val} \subset \mathbb{D} \]  

(5.8)

\[ \mathbb{D}_{test} \subset \mathbb{D} \]  

(5.9)

\[ \mathbb{D}_{train} \cap \mathbb{D}_{val} = \emptyset \]  

(5.10)

\[ \mathbb{D}_{train} \cap \mathbb{D}_{test} = \emptyset \]  

(5.11)

\[ \mathbb{D}_{val} \cap \mathbb{D}_{test} = \emptyset \]  

(5.12)

With the data collected, a model on some form (which can be of many different types) is assumed and trained. The fitting of a model to data is associated with the minimization of an objective function (sometimes called performance measure),
which is usually chosen to be the mean squared error between the model outputs and the training and validation data, respectively. To prevent overfitting, the training of a model is stopped when the error with respect to the validation set starts to increase with further training, as displayed in figure 5.1. There are also additional methods to prevent overfitting from occurring, such as regularization, where model parameters are added (with a weight) to the objective function [9]. There are also probabilistic approaches to prevent overfitting, such as Bayesian regularization [15].

![Figure 5.1: Illustration of cross-validation between training and validation data. The training error always decrease with additional training iterations, whereas the validation error does not. A typical case is shown in the figure, where the validation error initially decreases up to a point of best fit, where further training will result in lower performance, known as overfit. It is thus desirable to end the training process in the vicinity of this point. Stopping too early results in underfit.](image)

### 5.3 Artificial Neural Networks

There exist many different types of black-box models with structures suitable for modelling various dynamic behaviours. The type under consideration here is called Artificial Neural Networks. They are a type of black-box model inspired loosely on their namesake biological counterparts. These networks are made up of three types of layers, an *input layer*, *hidden layers*, and an *output layer*.

The input layer consists of an arbitrary input vector $X$ from the input space $\mathcal{I}$, and the output of the network consists of a vector $Y \in \mathbb{R}^m$ (hopefully) falling within the output space $\mathcal{O}$ of the system $S$ while also being the image of $X$ under
The hidden layers consist of one or more neurons. A neuron is essentially a function which takes as input a weighted sum of a subset of the previous layer’s output, add to it a bias, and propagate it through an activation function, which is often chosen to be a monotonically increasing and differentiable function with a range of \((-1, 1)\) or \((0, 1)\). The connections between the neurons between layers are sometimes called axons. In a fully connected feed-forward neural network, the output of each neuron in layer \(k\) is connected (with individual weights for each separate connection) to each neuron in layer \(k+1\). Let \(y_{i}^{k+1}\) denote the output of neuron \(i\) in layer \(k + 1\), \(h_{i}^{k+1}\) its input, \(b_{i}^{k+1}\) a bias and \(w_{j,i}^{k+1}\) the weight with which the output \(y_{j}^{k}\) of neuron \(j\) in layer \(k\) enters the input of neuron \(i\) in layer \(k + 1\). Finally, \(n_{k}\) denotes the number of neurons in layer \(k\). A neuron can thus be expressed mathematically as

\[
y_{i}^{k+1} = f(h_{i}^{k+1})
\]

\[
h_{i}^{k+1} = b_{i}^{k+1} + \sum_{j=1}^{n_{k}} w_{j,i}^{k+1} y_{j}^{k}
\]

where \(f(x)\) is the activation function, commonly chosen to be the standard logistic sigmoid function

\[
f(x) = \frac{1}{1 + e^{-x}}
\]

because of its property of having a range of \((0,1)\), being monotonically increasing and continuously differentiable (which allows for gradient-based methods to be used in the training process) over its entire domain of all the real numbers. Propagating an input through the entire network reveals the structure to be a composition of functions and they are usually trained using algorithms falling within the backpropagation class of methods. The interested reader is referred to [9] for more information on neural networks, backpropagation and other related topics.

### 5.4 Model Structure

A common choice of black-box models in time series prediction problems are the Nonlinear Auto-Regressive eXogenous input (NARX) class of models. The output of a general NARX model can be expressed as [20]

\[
\hat{y}(k) = g(y(k - 1), \ldots, y(k - m), u(k), \ldots, u(k - n))
\]

where \(g(\cdot)\) is a nonlinear function, \(\hat{y}(k)\) is a prediction of an upcoming value \(y(k)\), and \(u(k)\) denotes an input. Note that the predictions are dependent on past values of both the output \(y\) and the input \(u\). Also note that for use in an MPC, the predictions should be computed not only for one future step, but for an entire
horizon of length $N$. This means that while the NARX model will initially be based on measurements or estimates of the output $y$, all of the following predictions over $N$ must be, in part or completely, based on feedback of past predictions of the output $\hat{y}$. The function is, in a sense, repeated over the entire horizon.

Let $H^p_\alpha$ denote the horizon for past road slope values in number of discrete positional steps and $H^f_\alpha$ the future equivalent. Also, let $H^p_v$ denote the horizon in number of discrete positional steps for which past outputs (velocities) are used to predict the upcoming one. For an illustration of this see Figure 5.2. We thus seek a function

$$\hat{v}_{k+1} = f\left(\alpha_{k-H^p_\alpha+1}, \ldots, \alpha_k, \ldots, \alpha_{k+H^f_\alpha}, \hat{v}_{k-H^p_\alpha+1}, \ldots, \hat{v}_k\right)$$  \hfill (5.17)

**Figure 5.2:** Illustration of the one step ahead velocity prediction problem and its inputs and output. The inputs are the past and future road slope values and past velocities, and the output is an estimate of the velocity in the next discrete step, $\hat{v}_{k+1}$.

First, we split the velocity predictions into two parts according to

$$\hat{v}_{k+1} = \hat{v}^{cc}_{k+1} + \hat{v}^\delta_{k+1}$$  \hfill (5.18)

where $\hat{v}^{cc}_{k+1}$ denotes a bias representing an assumed or estimated cruise controller set speed of the preceding vehicle at discrete positional step $k + 1$. The second term in (5.18) $\hat{v}^\delta_{k+1}$ represents the estimated deviation from the cruise controller set speed of the preceding vehicle at discrete positional step $k + 1$. The cruise controller set speed is assumed to remain constant for a horizon, thus

$$\hat{v}^{cc}_{k+1} = f_{cc}(\hat{v}^{cc}_k) = \hat{v}^{cc}_k$$  \hfill (5.19)

and can be initiated with a guess from the road speed limit, or be estimated with methods covered later on. The deviations from set speed are modelled by

$$\hat{v}^\delta_{k+1} = f_{\delta}(\alpha_{k-H^p_\alpha+1}, \ldots, \alpha_k, \ldots, \alpha_{k+H^f_\alpha}, \hat{v}^\delta_{k-H^p_\alpha+1}, \ldots, \hat{v}^\delta_k)$$  \hfill (5.20)

where the function $f_{\delta}(\cdot)$ is a NARX neural network. The neural network is thus a model for the velocity response in terms of how it increases or decreases given some input data on the road topography and past estimates of velocity deviations.

### 5.4.1 ACC Set Speed Deviation Predictors

Several architectures of varying number of layers and neurons have been evaluated. In the case of ACC prediction, adding more than one hidden layer seems
to have an almost negligible effect on performance. The final structure used for all of the ACC networks consists of one hidden layer with four neurons, with 1 through 30 past road slope values and 1 past set speed deviation used as input.

The neural networks modelling the ACC set speed deviations are thus functions on the following form

\[
\hat{v}_{k+1}^\delta = f_\delta(\alpha_k-H_p^a+1, \ldots, \alpha_k, \ldots, \alpha_k+H_f^p, \hat{v}_{k-H_p^p+1}^\delta, \ldots, \hat{v}_k^\delta) \tag{5.21}
\]

where

\[
H_p^a = 30 \tag{5.22}
\]
\[
H_f^a = 0 \tag{5.23}
\]
\[
H_p^v = 1 \tag{5.24}
\]

The training, validation and test data are obtained from simulations running ACC trucks with a set speed of 80 km/h for a total distance of roughly 2000 km. The networks are trained using Bayesian regularization backpropagation, on which more information can be found in [15].

### 5.4.2 LACC Set Speed Deviation Predictors

Finding a structure suitable for LACC prediction is somewhat more complex due to the lower correlation between road slope and velocity [18]. Empirical results indicate that using more than two hidden layers will not increase performance. The final structure used for all of the LACC networks consists of one hidden layer with ten neurons, with 1 through 30 past road slope values, every fifth future road slope value up to a horizon of 300, and 1 past set speed deviation used as input. The future road slope inputs are thinned to reduce the model complexity and reduce training times.

The neural networks modelling the LACC set speed deviations are thus functions on the following form

\[
\hat{v}_{k+1}^\delta = f_\delta(\alpha_k-H_p^a+1, \ldots, \alpha_k, \alpha_k+1, \ldots, \alpha_k+H_f^a, \ldots, \alpha_k+H_f^a-5, \alpha_k+H_f^a, \ldots, \hat{v}_{k-H_p^p+1}^\delta, \ldots, \hat{v}_k^\delta) \tag{5.25}
\]

where

\[
H_p^a = 30 \tag{5.26}
\]
\[
H_f^a = 300 \tag{5.27}
\]
\[
H_p^v = 1 \tag{5.28}
\]

The training, validation and test data are obtained from simulations running LACC trucks with a set speed of 80 km/h for a total distance of roughly 2000 km. These networks are, like the ACC networks, trained using Bayesian regularization backpropagation.
5.5 Correction and Classification

A question of interest is whether a limited amount of predictors, together with a system for predictor output correction and classification, can yield satisfactory performance for several different HDVs. The parameters of these HDVs might not be the same as those for which the predictors were originally trained. A rudimentary system is proposed to handle these tasks. The basic idea consists of three steps. The first is to make predictions up to some future horizon, and then store speed measurements of the preceding vehicle until that future horizon is reached. The second step is to maximize the fit of each predictor for that window with respect to the measurements of the preceding vehicle, this is what we call the correction step. The third and last step is to compute the fit of each corrected predictor and then select the one with highest fit, which is then to be used for making predictions in the MPC for the duration of the next window. Before the first cycle of steps is completed, the MPC will assume constant velocity of the preceding vehicle.

5.5.1 Correction of Predictions

Let \( H_c \) denote the horizon of the window on which the correction is based, and \( n_p \) the number of models for the prediction of preceding vehicle set speed deviations. The following predictions of the preceding vehicle velocity response over the horizon are initially made

\[
\hat{v}^{\delta}_{1} = \begin{bmatrix} \hat{v}^{\delta,1}_{1} \\
\vdots \\
\hat{v}^{\delta}_{n_p} 
\end{bmatrix} 
\]

(5.29)

When the horizon is reached, the following measurements of the preceding vehicle have also been collected

\[
v^{\text{meas}} = \begin{bmatrix} v^{\text{meas}}_{1} \\
\vdots \\
v^{\text{meas}}_{H_c} \end{bmatrix} 
\]

(5.31)

By means of affine transformations, we get

\[
z_1 = \begin{bmatrix} z_{1} \\
\vdots \\
z_{H_c} \end{bmatrix}^T = A_1 x_1 = \begin{bmatrix} 1 & \hat{v}^{\delta}_{1} \\
\vdots & k_1 \end{bmatrix} \begin{bmatrix} v^{cc}_{1} \\
k_1 \end{bmatrix} 
\]

(5.32)

\[
\vdots 
\]

(5.33)

\[
z_{n_p} = \begin{bmatrix} z_{n_p}^{H_c} \end{bmatrix}^T = A_{n_p} x_{n_p} = \begin{bmatrix} 1 & \hat{v}^{\delta}_{n_p} \end{bmatrix} \begin{bmatrix} v^{cc}_{n_p} \\\nk_{n_p} \end{bmatrix} 
\]

where we want to, for each of these transformed predictions, find an estimate of the preceding vehicle set speed, \( v^{cc} \), and a gain \( k \) such that the fit of the transformed predictions \( z \) is maximized with respect to the measurements \( v^{\text{meas}} \).
First and foremost, the term fit usually refers to the complement of the normalized root mean square error (a number in the interval $[0, 1]$) that allows for comparison of performance on different datasets.

Let $\hat{y} = [\hat{y}_1 \ldots \hat{y}_n]^T$ denote predictions of some signal over some horizon $n$ and $y_{\text{meas}} = [y_{\text{meas}}^1 \ldots y_{\text{meas}}^n]^T$ measurements of that signal over the same horizon. The fit of $\hat{y}$ with respect to $y_{\text{meas}}$ is then commonly defined as

$$\text{Fit}(\hat{y}, y_{\text{meas}}) \doteq 1 - \frac{\sqrt{\frac{1}{n} \sum_{k=1}^n (\hat{y}_k - y_{\text{meas}}^k)^2}}{\sqrt{\frac{1}{n} \sum_{k=1}^n ((\frac{1}{n} \sum_{k=1}^n y_{\text{meas}}^k) - y_{\text{meas}}^k)^2}} \quad (5.34)$$

The fit between the corrected output of predictor $i$ and the measurement data is then

$$\text{Fit}(z_i, v_{\text{meas}}) \doteq 1 - \frac{\sqrt{\frac{1}{n} \sum_{k=1}^n (z_i^k - v_{\text{meas}}^k)^2}}{\sqrt{\frac{1}{n} \sum_{k=1}^n ((\frac{1}{n} \sum_{k=1}^n v_{\text{meas}}^k) - v_{\text{meas}}^k)^2}} \quad (5.35)$$

The problem of maximizing this fit with respect to the set speed $v^{cc}$ and correction gain $k$ can be reduced to the following minimization problem

$$\min_{x_i} (z_i - v_{\text{meas}})^T (z_i - v_{\text{meas}}) = (A_ix_i - v_{\text{meas}})^T (A_ix_i - v_{\text{meas}}) \quad (5.36)$$

which is a standard linear least squares problem [10]. Differentiating the objective function (5.36) with respect to the sought parameters $x_i$ yields

$$\nabla_{x_i} (A_ix_i - v_{\text{meas}})^T (A_ix_i - v_{\text{meas}}) = 2A_i^T A_ix_i - 2A_i^T v_{\text{meas}} \quad (5.37)$$

Solving (5.37) equal to zero for the correction parameters $x_i$ yields the solution to the minimization problem in (5.36), namely

$$x_i = (A_i^T A_i)^{-1} A_i v_{\text{meas}} \quad (5.38)$$

The parameters $x_i = [v_i^{cc} \ k_i]^T$ are then to be used to correct the output of model $i$ for the duration of the next window.

### 5.5.2 Classification of Preceding Vehicle

When the corrected predictor outputs have been computed for the latest window, the next step is to check the variance of the velocity measurements. If the variance is smaller than some threshold value, which is a design parameter, the predictor selected in the previous window will be used in the upcoming window as well. If no previous classification has been done, the classification system recommends a constant velocity assumption to the controller. The velocity variance filter is used to avoid misclassification based on measurements in windows where the velocity is relatively constant (a low signal-to-noise ratio). If the variance of
the velocity measurements exceeds the threshold value, the next step is to compute the fit of all predictors and then select one according to

\[ \arg \max_i \text{Fit}(z_i, v_{\text{meas}}) \]  

(5.39)

The selected predictor is then to be used in the controller for the duration of the next window.

**Classification & Correction Algorithm**

The principal function of the correction and classification system is depicted in Figure 5.3. Since it takes one correction horizon \( H_c \) to classify the preceding vehicle, the algorithm outputs constant velocity during the first window.

*Figure 5.3: Visualization of the classification and correction algorithm.*
This chapter starts with a brief background in the area of model-based control in general and model predictive control in particular. The aim is to provide the reader with an understanding and a motivation of the chosen controller. Later on, the controller developed in this thesis work is derived and implementation details are considered.

6.1 Control Theory

The classical control problem is to make a system behave in a desired way, even if uncertainties and disturbances are present and are acting on the system. This problem can be dealt with and solved in many different ways depending on the properties of the system and its surroundings. A very common controller that often yields satisfactory results is the PID-controller. In its simplest form it does not require a system model and can be easily implemented in many applications. However, when the system grows to include multiple input and output (MIMO) signals it can be hard to tune the PID-controller to achieve satisfactory performance. A popular approach in these cases is then to use a model-based controller.

6.1.1 The Linear Quadratic Controller

A common way to control MIMO-systems is to use the Linear Quadratic (LQ) control framework. For the following system,

$$\bar{x}_{k+1} = F\bar{x}_k + G\bar{u}_k$$  \hspace{1cm} (6.1)

the control law is chosen by minimizing a performance measure (often called cost function or objective function) quadratic in the state variables and control
signals, with respect to the control signal

$$\text{minimize} \quad \sum_{k=0}^{\infty} \|\tilde{x}(k)\|_{2,Q}^2 + \|\tilde{u}(k)\|_{2,R}^2$$

(6.2)

where $\|x\|_{2,Q}^2 = x^T Q x$ denotes the weighted and squared L2 norm, $Q$ and $R$ are positive semidefinite and positive definite weight matrices, respectively, and

$$\tilde{x}_k \triangleq x_k - x_0$$

where $x_0$ denotes a stationary linearization point, and likewise for the other barred variables. The weight matrices are design parameters for the controller which are chosen to achieve satisfactory behaviour of the closed-loop system. The optimal feedback control law to (6.2) is a linear state feedback law as per

$$\tilde{u}(k) = -L\tilde{x}(k)$$

(6.3)

where $L = (R + G^T P G)^{-1} G^T P F$ and $P$ is found by solving the discrete time algebraic Riccati equation which can be done numerically by utilizing appropriate software if certain conditions on the system are fulfilled [8]. The resulting LQ-controller often performs very well and is simple and numerically sound in its implementation since the only operation made in each call to the controller is the matrix multiplication (6.3). However, in practice there are often critical bounds on the states which the LQ-controller can not guarantee to satisfy. For example, the engine torque of an HDV will quite often saturate during an uphill road segment. Also, the truck will quickly pick up speed driving downhill and will likely need to brake to avoid violating the speed limitation. Such constraints can be implicitly dealt with by tuning the weight matrices $Q$ and $R$. This can work quite well, but the performance of the closed-loop system may worsen due to this tuning.

### 6.1.2 The Model Predictive Controller

In this thesis work, a model predictive controller (MPC) has been developed as an alternative or extension to the LQ-controller, in which constraints can be dealt with more explicitly. The fundamental idea is to solve the optimization problem (6.2) for $u$ with additional constraints, such as

$$|u| \leq u_{max}$$

(6.4)

$$|x| \leq x_{max}$$

(6.5)

The infinite horizon problem (6.2) cannot be solved in general since that would lead to an optimization problem with infinite decision variables. Instead (6.2) is
truncated and approximated as

$$\begin{align*}
\text{minimize} & \quad \sum_{j=0}^{N-1} \|\tilde{x}(k+j)\|_{2,Q}^2 + \|\tilde{u}(k+j)\|_{2,R}^2 \\
\text{s.t.} & \quad \bar{x}_{k+j+1} = F\bar{x}_{k+j} + G\bar{u}_{k+j} \\
& \quad |x_{k+j}| \leq x_{\text{max}} \\
& \quad |u_{k+j}| \leq u_{\text{max}} \\
& \quad \bar{x}_k \text{ given}
\end{align*}$$

(6.6)

where the objective function is now minimized over the finite prediction horizon $N$ which is an important tuning parameter in the MPC. The solution $\tilde{u}^*$ to (6.6) is the optimal control signal trajectory which needs to be recomputed every sampling instance when a new state measurement or estimate $\bar{x}(k)$ has been obtained. The MPC usage sequence is compiled in Algorithm 1. As described by step 2 in the algorithm, the optimization problem (6.6) needs to be solved in each call to the controller. That can sometimes be a relatively heavy procedure, computationally speaking, and has therefore led to MPCs mainly being used to control applications with slow dynamics, such as in the process industry, which allows for long sampling times. However, during the past years modern computers have made the usage of MPC in faster applications possible and its field of usage is growing quickly.

**Algorithm 1: MPC [4]**

1. Obtain current states $\bar{x}(k)$ by measurements or an observer
2. Calculate control signal trajectory $u$ by solving (6.6)
3. Apply the first element of $u$
4. Time update, $k := k + 1$
5. Repeat from 1

**Quadratic or Linear Objective Function**

In its current format, the objective function in (6.6) is quadratic with respect to the states and the control signals which is in accordance with the standard literature on MPC [8]. A quadratic cost function subject to linear inequality and equality constraints lead to a Quadratic Program (QP) to which there are numerous efficient solvers available. However, instead of using the weighted and squared L2 norm one can use the weighted L1 norm ($\|x\|_{1,w} = \sum_i |w_i x_i|$ where $w$ is a vector of weights) in the objective function, then the optimization problem can be rewritten as a Linear Program (LP). This will alter the solution a bit leading to different behaviour of the controller.
As an example, consider minimizing the objective functions depicted in Figure 6.1 subject to $X, Y \geq 0$. In the case of using the L1 norm the cost of moving a certain distance along the objective function is the same everywhere. If one instead employs a squared L2 norm objective function, the cost of moving a certain distance is very small near the origin compared to further away. This leads to a more smooth controller when the L2 norm (QP) is used while in the L1 case (LP) the controller tends to stay at points with active constraints more often.

![QP & LP Comparison](image)

**Figure 6.1:** Comparison between a quadratic and linear objective function

**Economic Model Predictive Control**

The term Economic Model Predictive Control (EMPC) is usually applied to control systems utilizing a controller on the form of Algorithm 1, but where the objective function in the minimization problem of (6.6) is instead chosen such that it reflects some economic aspect of operating the system [3]. In this case, such a function could describe the amount of fuel required to reach a certain state via some state and control trajectory.

**6.2 Controller Design**

We have in this thesis chosen to implement a controller based on the MPC framework. The main reason why MPC was chosen over LQ is that there are many explicit constraints in the HDV control problem. As an example, the HDV is by law limited to an upper speed limit which it likely will reach quite fast during downhill slopes due to its large mass. And due to the same mass the engine torque of the truck will saturate often during uphill slopes. Another important aspect
is that the optimization problem in the MPC framework can be designed to deal with integers and binary numbers, which are important in this work.

### 6.2.1 Setting Up the Optimization Problem

As described in Algorithm 1 the control signal is obtained by solving an optimization problem as per (6.6) during each sampling instance. The controller is therefore designed by formulating a proper objective function which is minimized subject to constraints on the states and the control signals. Since this thesis is about saving energy, the obvious choice would be to find an objective function that reflects the fuel consumption (2.34).

#### The Prediction Model

For prediction, the discrete and linear model (2.42) derived in Chapter 2 is used together with the gear shift model and PWA approximation from Chapter 3 and 4. The states in the model were previously defined as

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3 \\
  g_{12} \\
  g_{13} \\
  g_{14}
\end{bmatrix} =
\begin{bmatrix}
  \text{Velocity} \\
  \text{Time Headway} \\
  \text{Mass of Consumed Fuel} \\
  \text{State for Gear } 12 \in [0, 1] \\
  \text{State for Gear } 13 \in [0, 1] \\
  \text{State for Gear } 14 \in [0, 1]
\end{bmatrix}
\]

And the control signals as

\[
\begin{bmatrix}
  u_1 \\
  u_2 \\
  b_{12} \\
  b_{13} \\
  b_{14}
\end{bmatrix} =
\begin{bmatrix}
  \text{Engine Torque} \\
  \text{Brake Force} \\
  \text{Binary Activation Signal for Gear } 12 \\
  \text{Binary Activation Signal for Gear } 13 \\
  \text{Binary Activation Signal for Gear } 14
\end{bmatrix}
\]

#### Objective Function

The commonly used objective function in (6.6) often performs well but can be hard to tune for larger systems since the number of design parameters becomes many. Instead we make use of the mass of consumed fuel state derived in Chapter 2. A first step is to minimize the mass of consumed fuel at the end of the prediction horizon as

\[
\min_u x_3(k + N) \quad (6.7)
\]

Though using only (6.7) in the objective function is a good idea since the tuning becomes very easy, there is one possible addition. Recall the nonlinear air drag reduction function presented in Section 2.1.2. The function is linearized at \( \tau_{hw,0} \) (=1 s, \( \approx 22 \text{ meters @ 80 km/h} \)) which leads to an underestimation of the platooning effect. Therefore, additional penalties on deviations from the set velocity and
desired time headway are introduced as

\[
\text{minimize } \sum_{j=0}^{N-1} |Q_{11} \bar{x}_1(k + j)| + |Q_{22} \bar{x}_2(k + j)| + x_3(k + N) \quad (6.8)
\]

This has an emphasizing effect on the benefits of platooning to the controller, but adds more tuning complexity.

**Constraints**

Both the states and the control signals will be constrained, which as mentioned earlier is one of the main motivations for MPC. To stay with LP and QP problems the constraints need to be linear on the form

\[ Ax \leq b \]

which suits most of the constraints in this problem well. However, one constraint which makes the problem more realistic is the fact that maximum engine torque is dependent on engine speed. This dependency is in reality quadratic but is here approximated by linear constraints, as depicted in Figure 6.2. Engine speed is not available as an own state in (2.42) but during periods with locked clutch it is just a scaling of the velocity as

\[
\omega_e^* = \frac{i_G i_f d}{r_w} v \quad (6.9)
\]

and the linear constraints, can be written as

\[
\begin{align*}
    u_1 &\leq k_{\text{left}} x_1 + m_{\text{left}} \quad (6.10) \\
    u_1 &\leq k_{\text{right}} x_1 + m_{\text{right}} \quad (6.11) \\
    u_1 &\leq u_{1,\text{max}} \quad (6.12) \\
    u_1 &\geq u_{1,\text{min}} \quad (6.13)
\end{align*}
\]

Together these constraints make up a good approximation of the real system limitation.
Slack Variables

Since the controller is not continuously evaluated but rather sampled at a certain frequency there will, due to model errors, be times when states override the constraints. This will lead to an infeasible problem for the solver and a solution cannot be obtained. To get around this problem, slack variables $\sigma \geq 0$, are introduced in the problem on the form

$$\begin{align*}
\text{minimize} \quad & f(u) + f(\sigma) \\
\text{s.t.} \quad & g(u) \leq g_{\text{max}} + \sigma
\end{align*} \quad (6.14)$$

Now the hard constraints can be overridden at a cost of $f(\sigma)$ which is a tuning function that can be either linear or quadratic. This gets rid of the feasibility problems but on the other hand introduces more degree of complexity when it comes to tuning.
6.2.2 Resulting Controller

The optimization problem solved in the controller is (recall that \( x_g = [g_{12} \ g_{13} \ g_{14}]^T \) and \( u_g = [b_{12} \ b_{13} \ b_{14}]^T \))

\[
\text{minimize } \quad \left( \sum_{j=0}^{N-1} |Q_{11}\bar{x}_1(k+j)+|Q_{22}\bar{x}_2(k+j)|+R_\sigma(k+j) \right) + x_3(k+N) \\
\text{s.t. } \quad x_{k+j+1} = \begin{cases} 
F_{12}(x_{k+j} - x_0) + G_{12}(u_{k+j} - u_{0,12}) + J_{12}(w_{k+j} - w_0) & \text{if } g_{12} \geq g_{th} \\
F_{13}(x_{k+j} - x_0) + G_{13}(u_{k+j} - u_{0,13}) + J_{13}(w_{k+j} - w_0) & \text{if } g_{13} \geq g_{th} \\
F_{14}(x_{k+j} - x_0) + G_{14}(u_{k+j} - u_{0,14}) + J_{14}(w_{k+j} - w_0) & \text{if } g_{14} \geq g_{th} \\
F_0(x_{k+j} - x_0) + G_0(u_{k+j} - u_{0,0}) + J_0(w_{k+j} - w_0) & \text{else} 
\end{cases} \\
u_{\min} \leq u_{k+j} \leq u_{\max} \\
u_{1,k+j} \leq k_{\text{left}} x_{1,k+j} + m_{\text{left}} \\
u_{1,k+j} \leq k_{\text{right}} x_{1,k+j} + m_{\text{right}} \\
x_{\min} - \sigma_{k+j} \leq x_{k+j} \leq x_{\max} + \sigma_{k+j} \\
\sigma_{k+j} \geq 0 \\
x_{g,k+j+1} = x_{g,k+j} + \frac{T_s^{\text{MPC}}}{\tau_g} \left( -x_{g,k+j} + u_{g,k+j} \right) \\
(g_{12,k+j}, g_{13,k+j}, g_{14,k+j}) \in [0, 1] \\
b_{12,k+j} + b_{13,k+j} + b_{14,k+j} = 1 \\
b_{12,k+j}, b_{13,k+j}, b_{14,k+j} \in [0, 1]
\] (6.15)

Numerical values of the constraint limits are presented in Table A.4 in Appendix A. The motion model (2.42) is modeled in (6.15) as an equality constraint leading to an implicit prediction form. This can sometimes be numerically advantageous over the explicit prediction form and the problem gets a well defined structure, something that can be used by the solver [21]. Since (2.42) is linearized around the reference points no external reference signal is needed in (6.15). A slightly modified version of (6.15) which penalizes the square of the state deviations is also implemented leading to a mixed integer quadratic program according to

\[
\text{minimize } \quad \left( \sum_{j=0}^{N-1} (Q_{11}\bar{x}_1(k+j))^2 + (Q_{22}\bar{x}_2(k+j))^2 + R_\sigma(k+j) \right) + x_3(k+N) \\
\] (6.16)

Note that the slack \( \sigma \) remains linear together with the mass of fuel state, since it makes no further sense to minimize the square of consumed fuel and experiments have shown that the slack tends to mimic the no slack objective function better when linear.
6.2 Controller Design

6.2.3 Solving the Optimization Problem

When the optimization problem has been formulated it obviously needs to be solved, preferably as fast as possible. In (6.15) there is a binary constraint on $u_g$ and the PWA models is implemented using binary logic as well. This results in a mixed integer program (MIP) or more specifically a mixed integer linear program, when written as (6.15), or a mixed integer quadratic program when (6.16) is used in (6.15). In a MIP the convexity is lost and most solvers deal with the problem by utilizing branch and bound techniques. This can for some problems be very efficient, and for others very time demanding.

Move-blocking

To reduce the computational complexity of the MIP and hopefully make it faster to solve, move-blocking is introduced. The move-blocking technique allows for control signals to be blocked during some parts of the horizon over which the optimization occurs. This can make things easier for the solver, since the number of valid solutions may decrease.

In the implemented controller (6.15) we utilize move-blocking for the binary gear control signal $u_g$. This means in practice that the truck is prevented from shifting gear during some portions of the prediction horizon. This also makes sense since in a real truck it is seldom energy optimal to shift up and down with a high frequency. A drawback with move-blocking is that it introduces another tuning parameter $M$, the number of prediction steps the control signal is blocked.

In (6.15) we let $u_g$ be free at predictions steps $j = 0, M, 2M, \ldots, N − 1$ which constrains it to

$$u_{g,k+j} = \begin{cases} 
\text{free} & \text{if } (j \mod M) = 0 \\
 u_{g,k+j-1} & \text{else} 
\end{cases}$$

$$j = 0, \ldots, N − 1$$

(6.17)

Controller Memory

Even though move-blocking is introduced to prevent the controller from considering too many gear shifts over the prediction horizon, there is no memory in the controller. The controller works in a fashion where it is called at a certain frequency and upon a call it solves the optimization problem and sends out control signals, all according to Algorithm 1. The first control signal is applied and then the controller is called again. However, when called again it has no memory of the previous solution and as described by (6.17) $u_g$ is free at the first prediction step. This may in cases where the cost for two gears is similar lead to a behaviour where the controller shifts up in one call and down in the next call and so on.

To prevent this an extra term is added to the objective function as

$$J = J + R_{\text{shift}} \|u_{g,k|k} - u_{g,k-1|k-1}\|_1$$

(6.18)

where $J$ denotes the objective function in (6.15) and $R_{\text{shift}}$ is the cost of shifting at the first prediction step which becomes a tuning parameter. The notation $u_{g,k|k}$
here denotes the first element of the gear control signal trajectory obtained in the controller call at discrete step $k$, while $u_{g,k-1|k-1}$ denotes the first element of the gear control signal trajectory obtained in the controller call at the previous discrete step, $k - 1$.

**Tuning**

The resulting controller consists of a couple of tuning parameters which have, as explained in previous sections, been added to achieve satisfactory performance. Reasonable values of the parameters have been found by means of simulation and are concluded in Table A.3 in the appendix. Since the controller internally works with sampling distance as explained in Section 2.2.1 the effective length of the prediction horizon becomes

$$\text{Prediction Distance} = NT_s v$$

which at a speed of 80 km/h becomes approximately 2.2 km using parameters from Table A.3, Appendix A.

**YALMIP**

The practical implementation of (6.15) in this project is done using YALMIP [21], a toolbox for MATLAB that allows for rapid controller prototyping and automatic solver interfacing with neat syntactical features overall. This means that (6.15) can be implemented in MATLAB more or less as it is and there is no need to reformulate the problem or manually handle communication with solvers. To see how that can be done the interested reader is refereed to [4] for a textbook approach or [7] for a more practical example. In the implementation a state of the art MIP solver from GUROBI OPTIMIZATION is used.

**Computational Complexity**

To see how different combinations of the prediction horizon $N$ and the moveblocking parameter $M$ affects the controller evaluation time, refer to Figure 6.3. The figure presents evaluation times measured on a PC running WINDOWS 7 with an Intel®i5 processor at 3.30 GHz and 8 GB of RAM. Note that by definition $M \leq N$, thus not all combinations of $M$ and $N$ can be evaluated.
6.2 Controller Design

![Controller evaluation times for different N and M](image)

**Figure 6.3:** Controller evaluation times for different combinations of $N$ and $M$. 

[Image of graph showing evaluation times for different $N$ and $M$ values]
In this chapter, results of both the individual modules and the whole system are presented.

### 7.1 Set Speed Deviation Predictor Performance

The set speed deviation predictors developed in Chapter 5 are evaluated by predicting the set speed deviation profile of a truck which is controlled either using ACC or LACC and then comparing it to the true profile. The test data used for evaluation is obtained from simulations on two different road sections. One is the road between Södertälje and Norrköping, a varying stretch with average slopes. The other is the road between Koblenz and Trier, a very hilly section on which most of the trucks struggle with maintaining their set speed. The predictions are always made over the complete horizon at step 0, and the only measurement available for the predictors is the set speed deviation at step 0. Note that each predictor in Tables 7.1 - 7.4 is evaluated against test data obtained from simulating an identical HDV, but driving on new road sections not used in training.

#### 7.1.1 ACC Predictor

The first evaluations of the ACC predictors are made on the section between Södertälje and Norrköping. The resulting profile for a 40 t HDV is presented in Figure 7.1. Performance measures for all modelled ACC HDVs are presented in Table 7.1.

The second set of evaluations are made on the section between Koblenz and Trier. The resulting profile for a 50 t HDV is presented in Figure 7.2. Performance measures for all modelled ACC HDVs are presented in Table 7.2.
Table 7.1: Performance measures of the ACC predictors predicting deviations from the cruise controller set speed between Södertälje and Norrköping.

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Avg. error [km/h]</th>
<th>RMSE [km/h]</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 t ACC</td>
<td>5.2e-04</td>
<td>0.0523</td>
</tr>
<tr>
<td>20 t ACC</td>
<td>0.0025</td>
<td>0.1176</td>
</tr>
<tr>
<td>30 t ACC</td>
<td>-0.0176</td>
<td>0.2000</td>
</tr>
<tr>
<td>40 t ACC</td>
<td>-0.0999</td>
<td>0.3935</td>
</tr>
<tr>
<td>50 t ACC</td>
<td>0.0407</td>
<td>0.5596</td>
</tr>
<tr>
<td>60 t ACC</td>
<td>-0.1428</td>
<td>0.6152</td>
</tr>
</tbody>
</table>

Table 7.2: Performance measures of the ACC predictors predicting deviations from the cruise controller set speed between Koblenz and Trier.

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Avg. error [km/h]</th>
<th>RMSE [km/h]</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 t ACC</td>
<td>0.0107</td>
<td>0.0688</td>
</tr>
<tr>
<td>20 t ACC</td>
<td>0.0137</td>
<td>0.1370</td>
</tr>
<tr>
<td>30 t ACC</td>
<td>-0.0846</td>
<td>0.6642</td>
</tr>
<tr>
<td>40 t ACC</td>
<td>-0.1925</td>
<td>0.8640</td>
</tr>
<tr>
<td>50 t ACC</td>
<td>0.1310</td>
<td>1.1024</td>
</tr>
<tr>
<td>60 t ACC</td>
<td>-0.0732</td>
<td>0.9701</td>
</tr>
</tbody>
</table>
7.1 Set Speed Deviation Predictor Performance

**Figure 7.1:** Evaluation of the 40 t ACC set speed deviation predictor on the road section between Södertälje and Norrköping.

Prediction of deviations from set speed
Road Profile: Södertälje - Norrköping
HDV: 40 t ACC

Road Profile

Error

![Graph showing prediction of deviations from set speed, road profile, and error between Södertälje and Norrköping.](image-url)
Figure 7.2: Evaluation of the 50 t ACC set speed deviation predictor on the road section between Koblenz and Trier.
7.1.2 LACC Predictor

The LACC predictors are evaluated on the same sections of road as the ACC predictors. Figure 7.3 presents the profile of a 40 t truck on the road between Södertälje and Norrköping. Figure 7.4 presents the resulting profile of a 50 t truck on the road between Koblenz and Trier. Performance measures for all LACC models are compiled in Table 7.3 and Table 7.4.

Table 7.3: Performance measures of the LACC models predicting deviations from the cruise controller set speed between Södertälje and Norrköping.

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Avg. error [km/h]</th>
<th>RMSE [km/h]</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 t LACC</td>
<td>-0.0255</td>
<td>0.5367</td>
</tr>
<tr>
<td>40 t LACC</td>
<td>-0.0927</td>
<td>1.0275</td>
</tr>
<tr>
<td>50 t LACC</td>
<td>0.0618</td>
<td>1.1657</td>
</tr>
</tbody>
</table>

Table 7.4: Performance measures of the LACC models predicting deviations from the cruise controller set speed between Koblenz and Trier.

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Avg. error [km/h]</th>
<th>RMSE [km/h]</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 t LACC</td>
<td>-0.0541</td>
<td>0.7842</td>
</tr>
<tr>
<td>40 t LACC</td>
<td>-0.0299</td>
<td>1.3993</td>
</tr>
<tr>
<td>50 t LACC</td>
<td>-0.3098</td>
<td>1.4521</td>
</tr>
</tbody>
</table>
Figure 7.3: Evaluation of the 40 t LACC set speed deviation predictor on the road section between Södertälje and Norrköping.
Figure 7.4: Evaluation of the 50 t LACC set speed deviation predictor on the road section between Koblenz and Trier.
7.2 Corrector Performance

The performance of the corrector system is evaluated in Figure 7.5. The top plot of Figure 7.5 shows the correction window which displays what has happened during the latest correction window (of length $H_c$). The measured profile was the actual profile of the preceding truck. The raw prediction is the output from the predictor at step 0, using a correct initial set speed guess of 90 km/h, but no gain $k$ correcting the set speed deviation part, $v^δ$. Estimates of $v^{cc}$ and $k$ are made at step 500, and used to form the corrected predictions, demonstrating their ability to improve the raw prediction.

The bottom plot shows the future horizon, thus measurements of the preceding truck are not available. However, the true profile is plotted for the sake of this evaluation. It then shows the raw output from the predictor and the corrected output formed using the parameters obtained from the correction window. It should be noted that the set speed deviation predictors are trained on data from trucks operating at a set speed of 80 km/h. The subject truck on which the corrector system is tested operates at 90 km/h, demonstrating the ability of the correction gain $k$ to adjust for the dynamic response being different at different set speeds.
Figure 7.5: The top plot shows a comparison between the measured velocity, the raw prediction and the corrected prediction. The bottom plot shows the future outcome with the true profile (only for comparison, not available for the controller), the raw prediction and the corrected prediction.

7.3 Classifier Performance

The classifier system is evaluated simply by letting it classify unknown vehicles driving a known road profile. Figure 7.6 presents histograms for different setups and shall be viewed at together with Table 7.5 decoding the identifiers. In the titles of each subplot in Figure 7.6, the true vehicle is presented together with its control strategy and set speed. The correction (and classification) horizon $H_c$ denotes the number of samples over which the classification occurs. It therefore follows naturally that the number of classifications made is inversely proportional to $H_c$. Note that the 40 t ACC truck with a set speed of 90 km/h in the lower left plot of Figure 7.6 is classified as a 50 t ACC truck. This is due to the predictor models having been trained on data where the trucks in question have operated using a set speed of 80 km/h. The dynamic response in set speed deviations of a 50 t ACC truck operating at a set speed of 80 km/h is simply more similar to a 40 t ACC at 90 km/h than a 40 t ACC at 80 km/h. Even though the classification system can provide some clues as to the mass of the preceding vehicle, it should
not be considered a system to be used for mass estimation. The classifier system should be treated as a system with the purpose of selecting the predictor model most appropriate for the prediction of the preceding vehicle velocity profile.

Table 7.5: Explanation of identifiers used in the classification system.

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Classified as</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No classification made</td>
</tr>
<tr>
<td>1</td>
<td>10 t, ACC</td>
</tr>
<tr>
<td>2</td>
<td>20 t, ACC</td>
</tr>
<tr>
<td>3</td>
<td>30 t, ACC</td>
</tr>
<tr>
<td>4</td>
<td>40 t, ACC</td>
</tr>
<tr>
<td>5</td>
<td>50 t, ACC</td>
</tr>
<tr>
<td>6</td>
<td>60 t, ACC</td>
</tr>
<tr>
<td>7</td>
<td>20 t, LACC</td>
</tr>
<tr>
<td>8</td>
<td>40 t, LACC</td>
</tr>
<tr>
<td>9</td>
<td>50 t, LACC</td>
</tr>
</tbody>
</table>

Figure 7.6: Histograms for classifier evaluation. The title of each subplot presents which is the true vehicle, its control strategy and set speed. It also specifies the classification horizon $H_c$ used.
7.4 Prediction, Correction and Classification on Real Data

A question of great interest is how well prediction models trained on simulated data perform when evaluated on data from real world subject trucks. Figure 7.7 presents results achieved with a predictor evaluated on real world truck velocity data.

**Figure 7.7:** Evaluation against real truck velocity data, shown in blue. The top plot shows a comparison between measured velocity and raw prediction initialized with an incorrect set speed guess of 90 km/h. It also shows the profile obtained from correcting the raw predictions with new estimates of \( v^{cc} \) and \( k \). The bottom plot shows the future outcome with the true profile (only for comparison, not available for the controller), the raw prediction and the corrected prediction.
7.5 Controller Performance

The performance of the controller is evaluated by simulations together with the nonlinear models derived in Chapter 2 implemented in SIMULINK. To make evaluation easier artificial road profiles and preceding vehicle velocity profiles are used.

7.5.1 QP and LP Evaluation

To evaluate the differences between the LP and QP objective functions derived in Chapter 6, simulations over an identical road segment have been made. Results from these simulations are gathered in Table 7.6 where $Q_{22}$ is the penalty for time headway deviation and $E_b$ is the brake energy. Other parameters are set according to Table A.3, Appendix A. The vehicle mass is 40 t both for the controlled vehicle and the preceding, which is controlled using ACC.

The best LP and QP from Table 7.6 are simulated behind an LACC, from which results are compiled in Table 7.7. As can be noticed in both tables, there are very small differences between the QP and the LP when the weight $Q_{22}$ is small. At larger $Q_{22}$ however, the two solutions start to differentiate. In Figure 7.8 a comparison between the solutions is displayed for $Q_{22} = 10$.

To prove the need for adding a state penalty as was done to (6.15), a simulation without state penalty is done. The objective function then only consists of (6.7), repeated here for convenience.

$$\min_u x_3(k + N)$$

This objective function only emphasizes the minimization of the total amount of fuel consumed at the end of an MPC horizon. Results from the simulation are presented in Table 7.8.

---

Table 7.6: Results from simulation with LP and QP controllers over the same road segment. The preceding vehicle is controlled using ACC.

<table>
<thead>
<tr>
<th>LP/QP</th>
<th>$Q_{22}$</th>
<th>$\tau_{hw,avg}$ [s]</th>
<th>$\tau_{hw,max}$ [s]</th>
<th>Tot. Fuel Cons. [L]</th>
<th>$E_b$ [J]</th>
</tr>
</thead>
<tbody>
<tr>
<td>LP</td>
<td>1</td>
<td>4.00</td>
<td>8.64</td>
<td>6.06</td>
<td>3.46e+7</td>
</tr>
<tr>
<td>LP</td>
<td>10</td>
<td>4.00</td>
<td>8.61</td>
<td>6.06</td>
<td>3.51e+7</td>
</tr>
<tr>
<td>LP</td>
<td>100</td>
<td>1.03</td>
<td>1.33</td>
<td>6.98</td>
<td>9.77e+7</td>
</tr>
<tr>
<td>QP</td>
<td>1</td>
<td>4.00</td>
<td>8.60</td>
<td>6.06</td>
<td>3.48e+7</td>
</tr>
<tr>
<td>QP</td>
<td>10</td>
<td>1.66</td>
<td>3.32</td>
<td>6.45</td>
<td>6.00e+7</td>
</tr>
</tbody>
</table>
Table 7.7: Results from simulation with LP and QP controllers over the same road segment. The preceding vehicle is controlled using LACC.

<table>
<thead>
<tr>
<th>LP/QP</th>
<th>$Q_{22}$</th>
<th>$\tau_{hw,avg}$ [s]</th>
<th>$\tau_{hw,max}$ [s]</th>
<th>Tot. Fuel Cons. [L]</th>
<th>$E_b$ [J]</th>
</tr>
</thead>
<tbody>
<tr>
<td>LP</td>
<td>1</td>
<td>4.05</td>
<td>8.38</td>
<td>5.95</td>
<td>3.15e+7</td>
</tr>
<tr>
<td>QP</td>
<td>1</td>
<td>3.98</td>
<td>8.31</td>
<td>5.95</td>
<td>3.31e+7</td>
</tr>
</tbody>
</table>

Table 7.8: Result from simulation with objective function as (6.7). The preceding vehicle is controlled using LACC, thus these results should be compared to the ones in Table 7.7.

<table>
<thead>
<tr>
<th>LP/QP</th>
<th>$Q_{22}$</th>
<th>$\tau_{hw,avg}$ [s]</th>
<th>$\tau_{hw,max}$ [s]</th>
<th>Tot. Fuel Cons. [L]</th>
<th>$E_b$ [J]</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>-</td>
<td>4.55</td>
<td>9.22</td>
<td>6.06</td>
<td>3.94e+7</td>
</tr>
</tbody>
</table>

Figure 7.8: Comparison between simulations made with LP and QP. In both cases $Q_{22}$ was set to 10.
7.5.2 Shifting Performance

The shifting performance is evaluated using the LP controller and an artificial road profile with the three hills;

1. Inclination \( +\) 0.06 radians (\( \approx 6\% \))
   Slope length 140 meters

2. Inclination \( +\) 0.04 radians (\( \approx 4\% \))
   Slope length 200 meters

3. Inclination \( +\) 0.08 radians (\( \approx 8\% \))
   Slope length 100 meters

and controller parameters according to Table A.3. Relevant simulation data is presented in Figure 7.9. The mass of the controlled vehicle is 40 t and the preceding vehicle is set to drive with a fixed velocity of 80 km/h. The black vertical lines crossing all of the figures in 7.9 mark points where the controller decides to engage a lower gear. All these gear shifts occur before the uphill slopes, indicating a desired predictive behaviour which avoids the drawbacks that come from shifting in the middle of a steep uphill.

Looking at the time headway figure of 7.9, a couple of interesting things can be noticed. The controller increases the time headway during the flat segments on top of the hills. When the controlled vehicle reaches the point where the downhills start, it is almost 10 seconds behind the preceding vehicle. This allows the vehicle to take advantage of almost all potential energy stored from the uphill climb during the downhill. There is however, some brake action during the downhill slopes due to limitations in maximum time headway and minimum/maximum velocity. The controller is not allowed to increase the distance and decrease the velocity to the point where no braking would have been required during the downhill.

In the bottom plot of Figure 7.9, which presents the engaged gear, one can at around 60 seconds notice the ambivalent behaviour mentioned in Section 6.2.3. The controller is having a hard time deciding whether to stay at gear 14 a bit longer or to shift down and stay at gear 13 or 12. This behaviour is however, not present during the other two downshifts which indicates that the extension (6.18) to the objective function works somewhat satisfactory, but may benefit from further tuning.
Figure 7.9: Simulation to demonstrate gear shift performance. Decision to engage a lower gear is marked with a black line crossing all figures.
7.6 System Performance

This section will present results from and evaluate the complete system.

7.6.1 Simulation Environment

The simulation environment used for system evaluation is developed by Scania CV AB and includes accurate nonlinear models of all relevant vehicle components and systems. The controller implemented in Chapter 6, hereinafter referred to as the MPC, is connected to this environment and its output signals are sent to relevant lower level controllers. The MPC in the simulations uses the LP objective function, mostly due to numerical stability problems with the QP.

Evaluation Road Section

The system performance will foremost be evaluated using simulations on the road between Södertälje and Norrköping. This 119 km stretch of road consists of both flat portions as well as some intermediate slopes, and a maximum inclination around 3.6 %.

Evaluation Truck

The controlled truck used for all simulations with the MPC has a mass of 40 t. This since the models are developed for an engine which suits that mass well.

Benchmarks

As a general benchmark for the results, several simulations of non-platooning trucks utilizing the most efficient controllers available today have been run, refer to table 7.9 for performance measures on these. Simulations where the controlled vehicle gets the true, not predicted, velocity profile of the preceding vehicle are also made as benchmarks for both predictor models and the system as a whole. Those will be referred to as MPC with oracle. Each simulation will include a comparison in fuel consumption between the simulated truck and a standard 40 t truck which is controlled using a conventional cruise controller, $\Delta_{\text{ACC}40}$.

<table>
<thead>
<tr>
<th>Contr.</th>
<th>Mass [kg]</th>
<th>Vel. [km/h]</th>
<th>Cons. [L/10km]</th>
<th>$E_h/s$ [J/10km]</th>
<th>$\Delta_{\text{ACC}40}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>40 000</td>
<td>79.39 ± 3.21</td>
<td>2.757</td>
<td>2.2049e+06</td>
<td>-7.73 %</td>
</tr>
<tr>
<td>S2</td>
<td>40 000</td>
<td>79.40 ± 3.14</td>
<td>2.765</td>
<td>2.2328e+06</td>
<td>-7.46 %</td>
</tr>
</tbody>
</table>

Table 7.9: Data for solo trucks using the most efficient modes of operation.

Table 7.10 presents data for the lead trucks of platoons, controlled using generic Scania controllers. Table 7.11 presents data for the follower truck in the platoons.
Table 7.10: Data for the different two-vehicle platoon lead trucks. The quantity $\Delta_{ACC40}$ is shown only for HDVs with a mass of 40 t since the controlled follower truck is always a 40 t truck. Thus, the values $\Delta_{ACC40}$ for different mass HDVs are not of interest since they would not be useful for evaluating control performance.

<table>
<thead>
<tr>
<th>Contr.</th>
<th>Mass [kg]</th>
<th>Vel. [km/h]</th>
<th>Cons. [L/10km]</th>
<th>$E_b/s$ [J/10km]</th>
<th>$\Delta_{ACC40}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC</td>
<td>20 000</td>
<td>80.27 ± 1.20</td>
<td>2.170</td>
<td>7.3566e+05</td>
<td>-</td>
</tr>
<tr>
<td>ACC</td>
<td>40 000</td>
<td>80.19 ± 2.41</td>
<td>2.988</td>
<td>4.2426e+06</td>
<td>0 %</td>
</tr>
<tr>
<td>ACC</td>
<td>50 000</td>
<td>79.66 ± 3.54</td>
<td>3.419</td>
<td>6.3576e+06</td>
<td>-</td>
</tr>
<tr>
<td>LACC</td>
<td>20 000</td>
<td>80.12 ± 1.44</td>
<td>2.109</td>
<td>4.7579e+05</td>
<td>-</td>
</tr>
<tr>
<td>LACC</td>
<td>40 000</td>
<td>80.00 ± 2.45</td>
<td>2.831</td>
<td>2.9707e+06</td>
<td>-5.25 %</td>
</tr>
<tr>
<td>LACC</td>
<td>50 000</td>
<td>79.66 ± 3.28</td>
<td>3.223</td>
<td>4.4690e+06</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 7.11: Data for follower trucks with a mass of 40 t using Scania ACC.

<table>
<thead>
<tr>
<th>Lead</th>
<th>Vel. [km/h]</th>
<th>Cons. [L/10km]</th>
<th>$\tau_{hw}$ [s]</th>
<th>$E_b/s$ [J/10km]</th>
<th>$\Delta_{ACC40}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 t ACC</td>
<td>80.13 ± 2.56</td>
<td>2.966</td>
<td>5.16 ± 2.99</td>
<td>4.7799e+06</td>
<td>-0.74 %</td>
</tr>
<tr>
<td>40 t ACC</td>
<td>80.13 ± 2.77</td>
<td>2.878</td>
<td>2.30 ± 1.02</td>
<td>4.7724e+06</td>
<td>-3.68 %</td>
</tr>
<tr>
<td>50 t ACC</td>
<td>79.65 ± 3.55</td>
<td>2.849</td>
<td>1.51 ± 0.49</td>
<td>4.7764e+06</td>
<td>-4.65 %</td>
</tr>
<tr>
<td>20 t LACC</td>
<td>80.05 ± 2.55</td>
<td>2.948</td>
<td>3.52 ± 2.54</td>
<td>5.2859e+06</td>
<td>-1.34 %</td>
</tr>
<tr>
<td>40 t LACC</td>
<td>80.00 ± 2.54</td>
<td>2.856</td>
<td>2.05 ± 1.00</td>
<td>4.4696e+06</td>
<td>-4.42 %</td>
</tr>
<tr>
<td>50 t LACC</td>
<td>79.66 ± 2.98</td>
<td>2.816</td>
<td>1.63 ± 0.68</td>
<td>4.1544e+06</td>
<td>-5.76 %</td>
</tr>
</tbody>
</table>

7.6.2 MPC Solution

This section first presents simulation results for a follower truck controlled with the MPC and oracle. Data from this simulation is presented in Table 7.12. This data serves as an internal benchmark for the data presented in Table 7.13, which is data for a follower truck using the MPC along with the predictor systems designed in Chapter 5.

A visualization of relevant data from the complete system (MPC, predictor, corrector and classifier) in operation can be seen in Figure 7.10. The scenario in the figure is the most fuel efficient of the scenarios from Table 7.13 where the controlled 40 t truck is driven behind a 50 t LACC truck.

To demonstrate the performance of the predictor in the system, simulations of two identical scenarios are made with the MPC, but with one using the oracle and the other using the predictor systems. The data from these simulations is presented in Figure 7.11. The data in Tables 7.12 and 7.13 as well as the figure in question suggest that the differences in fuel consumption between the MPC with oracle and the MPC with predictors are very small, on the order of tenths of a percent.
Table 7.12: Data for follower trucks with a mass of 40 t using MPC and oracle.

<table>
<thead>
<tr>
<th>Lead</th>
<th>Vel. [km/h]</th>
<th>Cons. [L/10km]</th>
<th>$\tau_{hw}$ [s]</th>
<th>$E_b/s$ [J/10km]</th>
<th>$\Delta_{ACC40}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 t ACC</td>
<td>80.05 ± 4.11</td>
<td>2.790</td>
<td>2.93 ± 2.27</td>
<td>1.3982e+06</td>
<td>-6.63 %</td>
</tr>
<tr>
<td>40 t ACC</td>
<td>79.95 ± 4.31</td>
<td>2.786</td>
<td>2.81 ± 2.23</td>
<td>1.3986e+06</td>
<td>-6.76 %</td>
</tr>
<tr>
<td>50 t ACC</td>
<td>79.43 ± 4.86</td>
<td>2.763</td>
<td>2.71 ± 2.16</td>
<td>1.2370e+06</td>
<td>-7.53 %</td>
</tr>
<tr>
<td>20 t LACC</td>
<td>79.93 ± 4.42</td>
<td>2.777</td>
<td>2.94 ± 2.27</td>
<td>1.2206e+06</td>
<td>-7.06 %</td>
</tr>
<tr>
<td>40 t LACC</td>
<td>79.80 ± 4.50</td>
<td>2.762</td>
<td>2.78 ± 2.15</td>
<td>1.1632e+06</td>
<td>-7.56 %</td>
</tr>
<tr>
<td>50 t LACC</td>
<td>79.43 ± 4.98</td>
<td>2.751</td>
<td>2.67 ± 2.10</td>
<td>1.1877e+06</td>
<td>-7.93 %</td>
</tr>
</tbody>
</table>

Table 7.13: Data for follower trucks with a mass of 40 t using MPC with classification and predictor systems.

<table>
<thead>
<tr>
<th>Lead</th>
<th>Vel. [km/h]</th>
<th>Cons. [L/10km]</th>
<th>$\tau_{hw}$ [s]</th>
<th>$E_b/s$ [J/10km]</th>
<th>$\Delta_{ACC40}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 t ACC</td>
<td>80.05 ± 4.09</td>
<td>2.790</td>
<td>2.93 ± 2.26</td>
<td>1.4019e+06</td>
<td>-6.63 %</td>
</tr>
<tr>
<td>40 t ACC</td>
<td>79.96 ± 4.31</td>
<td>2.786</td>
<td>2.88 ± 2.26</td>
<td>1.3773e+06</td>
<td>-6.76 %</td>
</tr>
<tr>
<td>50 t ACC</td>
<td>79.44 ± 4.95</td>
<td>2.770</td>
<td>2.75 ± 2.14</td>
<td>1.3024e+06</td>
<td>-7.30 %</td>
</tr>
<tr>
<td>20 t LACC</td>
<td>79.94 ± 4.23</td>
<td>2.780</td>
<td>2.92 ± 2.25</td>
<td>1.2665e+06</td>
<td>-6.96 %</td>
</tr>
<tr>
<td>40 t LACC</td>
<td>79.79 ± 4.67</td>
<td>2.782</td>
<td>2.79 ± 2.16</td>
<td>1.4615e+06</td>
<td>-6.89 %</td>
</tr>
<tr>
<td>50 t LACC</td>
<td>79.43 ± 5.17</td>
<td>2.763</td>
<td>2.74 ± 2.14</td>
<td>1.3059e+06</td>
<td>-7.53 %</td>
</tr>
</tbody>
</table>
Figure 7.10: A visualization of relevant data from the simulation with the system implemented in this thesis. The preceding vehicle is a 50 t LACC truck. Refer to data in Table 7.13.
Figure 7.11: Visualization of two identical scenarios where one truck uses the oracle and one the predictor. In the control signals subplot, the blue line is the engine torque [Nm] when using an oracle, and the yellow line is the engine torque when using the predictor systems. The red line is the braking force [N] when using an oracle, and the purple line is the braking force when using the predictor systems. In the engaged gear subplot, the blue line is the selected gear when using an oracle, and the red line is the selected gear when using the predictor systems.
This chapter covers analysis and discussion regarding the positive and negative aspects of both individual subsystems and the system as a whole. It contains propositions for future work, as there are many aspects of non-V2V platooning in need of further examination. Finally, the chapter answers the questions raised in Section 1.2.

8.1 Discussion

8.1.1 System for Prediction, Correction and Classification

The system for prediction, correction and classification performs fairly well, even in tests where the subject HDVs have operated at a different set speed or been of a different power-to-weight ratio than of those encountered in the training of the system. However, it is important to note that the underlying cruise controller structures and vehicular control systems of the subject HDVs in these tests have been of kinds encountered in the training data. It is thus difficult to say anything about how well the prediction system would perform on trucks with different powertrains, cruise controllers and lower level control systems.

The classification system also shows good performance, and as can be seen in Figure 7.6 it mostly manages to correctly classify the trucks used in tests. A question that arises when it comes to the correction and classification system is how long the correction and classification horizon $H_c$ should be. In Figure 7.6 it can be seen that larger $H_c$ yield better classification while introducing a longer delay before the first classification is made. This means that using a longer $H_c$ will also result in a longer period of time using the assumption of constant velocity of the preceding truck, which may affect the control performance negatively.

The performance of the complete system for prediction, correction and clas-
sification, can be seen by looking at Figure 7.11 in which the MPC using this system is compared to the same MPC using an oracle, that is a controller where the true preceding vehicle velocity profile is available. The only significant difference between the two solutions can be seen by looking at the first 10 km of the time headway plot. The controller with the predictor system has not yet classified the preceding vehicle and assumes constant velocity, which affects the time headway since the preceding vehicle velocity is not actually constant. However, after the classification is made at around 10km, the profiles almost coincide indicating satisfactory performance from the system in question. A comparison of Tables 7.12 and 7.13, in which the fuel consumption for both cases are found and almost identical, is sufficient to support this statement.

**Applicability to Real Operations**

The systems for prediction, correction and classification, are all designed to work given that a number of assumptions hold, which may not be the case in real world operations.

One interesting question relevant to real world operations is; what happens in case disturbances in the form of traffic are introduced? Everytime a call to the prediction system is made, it computes an estimate of the preceding vehicle velocity profile for an upcoming window. It is however, important to realize that the prediction system only have access to one speed measurement, which is used for initialization. Since the models describe the speed profile of a given truck as a function of road topography, they cannot foresee changes in speed due to reasons other than road topography. An example of this can be seen in Figure 8.1, where irregularities in the preceding vehicle speed profile due to traffic result in poor estimations of the set speed $v^cc$ and gain $k$. 
8.1 Discussion

Figure 8.1: Traffic induced prediction errors. First, the raw, uncorrected predictions are computed with a guess on the preceding vehicle cruise controller set speed $v_{cc}$ of 90 km/h, for the complete correction window. When the end of the window has been reached, all measurements necessary for the correction step have been obtained. However, disturbances in the form of traffic have resulted in a speed profile which is non-representative of an HDV operating a conventional cruise controller on this particular stretch of road (The V shape in the beginning of the correction window). The correction system computes a new estimate of the set speed, $v_{cc}$, and a gain $k$ for the set speed deviations, $v^\delta$, to maximize the fit of the predictions with respect to the measurements of the correction window. However, looking at the outcome in the lower plot, it is apparent that the correction has produced somewhat erroneous estimates. In part due to the disturbances from traffic, the set speed $v_{cc}$ was estimated to be lower than the true value, and the gain $k$ is also too low.
Another question of interest concerns the assumption of constant cruise controller set speed $v^{cc}$ of the preceding vehicle, over any given correction window. In real world operations, this would not necessarily be true, since a driver could hypothetically change the set speed at any time, even if the state of the traffic or condition of the road does not call for it. If the set speed $v^{cc}$ were to be changed over course of a correction window, it would induce a bias error in the predictions of the preceding vehicle speed profile for the upcoming window. For an example of a situation where this occurs, refer to Figure 8.2.

Figure 8.2: Prediction errors due to set speed changes during a correction window. First, the raw, uncorrected predictions are computed using a guess on the preceding vehicle cruise controller set speed $v^{cc}$ of 90 km/h, for the complete correction window. When the end of the window has been reached, all measurements necessary for the correction step have been obtained. However, about midway into the correction window, the preceding vehicle changed its cruise controller set speed. Since the correction step consists of maximizing the fit of the predictions with respect to the measurements of the entire correction window, this set speed change will induce some errors. The set speed $v^{cc}$ has been estimated to be lower than the true value, which is apparent by looking at the outcome in the lower plot.
8.1 Discussion

The algorithms could be extended to remedy the shortcomings during such transitional phases. One could add an algorithm to detect set speed changes in the correction window data, and then discard the data before this change, or otherwise modify the correction algorithm so as to not introduce bias errors in the predictions. Without such a system, the bias error introduced by the set speed change would persist until the next correction horizon is reached, when a new correction is done and the bias is adjusted (given constant set speed in that window).

The assumption of constant set speed during any given correction window is however, not completely unreasonable. This since operation on highways are usually conducted using cruise controller with the set speed being constant over distances significantly larger than a correction window, meaning that such transitions is a relatively rare event.

8.1.2 Controller

As mentioned in Section 6.2, the main motivation for implementing an MPC was its ability to explicitly account for constraints and exogenous inputs in the problem. This led to a controller which initially always stayed within its limits. As can be seen in Figure 7.10 the time headway satisfies

\[ \tau_{hw} \geq \tau_{hw,min} \forall t \]

which is good, especially from a safety perspective. Due to feasibility issues however, slack variables were introduced as explained in Section 6.2. This means that the controller actually is allowed to exceed the limits, but to a cost. The cost becomes a tuning parameter which adds complexity to the problem and the controller tuning. This was deemed necessary though, due to simulation stability.

As for the prediction horizon, simulations showed that a horizon of 2.2 km was enough to get a satisfactory performance. What happens beyond this distance seems to have no significant effect on the early parts of the optimal control signal trajectory. The sampling time was initially set to 0.5 seconds, but later increased to 1 second. The benefit is that the effective prediction horizon becomes longer, meanwhile performance can be affected during fast dynamical situations. During the simulations however, such situations seldom occurred which motivated staying with a sampling time of 1 second.

Model Simplifications

The modelling work done in Chapter 2 consists of several simplifications and assumptions. The one that affects the behaviour of the controller the most is the linearization of the air drag reduction, refer to Figure 2.3 and discussion in Section 2.1.2, which leads to an underestimation of the platooning benefits. Section 6.2.1 deals with measures to emphasize platooning behaviour by introducing penalties on deviations from the linearization points. However, another alternate approach to this problem could be to linearize the air drag reduction curve, Figure 2.3, around a smaller distance leading to a much steeper reduction curve.
This would probably lead to a controller which really prefers to stay close behind the preceding vehicle, and therefore benefit from platooning. Experiments using this method were conducted, but proved to be less fuel efficient compared to a controller which fluctuates more in distance. This indicates that the benefits of LACC behaviour may sometimes outweigh the benefits of platooning. This evaluation was made on the Södertälje Norrköping section, thus this reasoning is only valid for somewhat hilly roads.

**Gear Shifting**

The introduction of shifting possibilities between the three highest gears led to a flexible controller which could handle more driving situations, and became more realistic. The drawback was that the problem went from an LP/QP to MILP/MIQP which are more complex and time demanding to solve. This also made the controller tuning more difficult. With the introduction of the mass of consumed fuel state however, the shifting behaviour became more realistic and energy efficient and the tuning became more intuitive. There were quite some challenges in getting it to work satisfactory. One step was the addition of (6.18), which solved some problems with high frequency shifting. Though this was sort of a quick fix, and a more realistic loss model may have worked better. The decision to include the three highest gears proved to be a good choice since the need for an even lower gear seldom occurred during simulations using the generic Scania controllers. Tests were made to include a possibility to engage neutral gear for longer periods of time. This makes sense since there is quite some inertia in the powertrain of HDVs which is reduced if neutral is engaged. It proved however to be a hard task in terms of yielding a sensible behaviour, and the idea was left out. More modelling work and tuning may have been the key to get the neutral gear selection to work properly.

**LP or QP**

When it comes to LP versus QP there is much that can be discussed. During this project the LP controller has in general performed better in terms of fuel consumption, even though the differences on that measure have been minor. One reason for the slightly lower fuel consumption in the LP case is the fact that the cost of loosing up the distance to the preceding vehicle becomes smaller if the same weight matrices are used. This means that the LP controller is more willing to exploit the possibilities of LACC behaviour, explained in Section 1.1.1. An example of this can be seen in Figure 7.8, in which the LP controller fluctuates more in both velocity and time headway preventing it from braking in a greater manner compared to the QP. This can also be noticed by looking at the brake energy in Tables 7.6 and 7.7. It shall also be pointed out that there have been numerical problems with the MIQP solver which in turn led to more investigations and tuning of the LP controller. Before introducing the gearbox, some tests were conducted with one gear models and standard LP/QP controllers (no control signals with integrality constraints present). These indicated a more smooth behaviour of
the QP whereas the LP solutions tended to Bang-Bang control\footnote{A control sequence of either maximum or minimum input magnitude.}. Even though experiments on Bang-Bang control have been conducted in vehicles, it was deemed infeasible for this project and rate limiting constraints were introduced to prevent it.

**Computational Complexity**

Regarding the complexity of the controller, it has been measured that one controller evaluation on a PC with a 3.3 GHz processor takes around 0.2 seconds using tuning parameters from Table A.3. This is not too bad considering the complex optimization problem that needs to be solved in each call. Looking at Figure 6.3 it becomes clear that the moveblocking constraints have a great impact on evaluation times. However, the measurements in the figure were averaged over only 5 controller evaluations which is too few for statistical reliability. It is enough to show the trends, even though some lines display a somewhat irregular curvature.

Implementing the MPC on a standard truck control unit (ECU) is not deemed possible because of the low computational power of today’s ECUs. With the development of new autonomous technology there will be a need for more computational power, and it is therefore not unlikely that the MPC can be implemented in a truck in a fairly close future.

**8.1.3 Complete System**

The complete system performed quite satisfactory and as can be seen in Table 7.13, the fuel consumption of the most efficient truck using the MPC is close to what is achieved with state of the art Scania controllers for a single HDV (data in Table 7.9). It is possible that even better performance of the MPC could be expected with more tuning and tweaking. The implemented MPC is after all, as mentioned above, quite complex and time demanding to tune and once a satisfactory tuning was found, it was rarely changed.

Most evaluations of the system were made at the road section between Södertälje and Norrköping, which is somewhat representative for a typical Swedish highway. The stretch is however quite hilly, and platooning is most efficient on flat roads. Perhaps an additional stretch of road should have been used for evaluation where the platooning effect could be exploited more.

**8.2 Future Work**

There are currently some questions as to the possibility of extending the presented solutions to allow for platoon sizes larger than two. Is it for example possible that all of the following trucks has a model of the first truck and then uses different offsets in reference distances. In longer platoons, questions of stability also arise.
Another question of interest which was lightly investigated but left out in this work is whether extending the PWA system with the possibility of engaging and using neutral gear, rather than just passing through it in the process of shifting between gears, would be more energy efficient than the current solution.

Finally the problem of disturbances in the form of traffic other than the platooning trucks has been briefly discussed but remains unsolved. As was shown in Figure 8.1, surrounding traffic undeniably affects the current system for velocity profile prediction and therefore also the complete control system. It seems likely that it is possible to detect when a predicted profile is very different from what is later measured, and then use some sort of mode selector to deal with those situations.

8.3 Conclusions

As for the questions asked in the problem formulation in Chapter 1 concerning the two scenarios, the haulage scenario and the catch up and follow scenario, there are now some answers and conclusions. We can start by noting that the classification system works satisfactory and is able to classify, or rather pick the best predictor for a truck with good precision, at least of those tested, as shown in Figure 7.6. Because of that, no significant advantage of information about the preceding vehicle held beforehand could be seen in this work except the ordering with respect to power-to-weight ratio. Therefore the conclusions on both scenarios can be merged.

We have in this work shown that by using artificial neural networks trained on simulated data it is possible to, with good accuracy, predict the velocity profile of an HDV driving a section of road with known topography using a cruise controller. Further, this profile can be used in an MPC together with relevant dynamical models to control an HDV in an energy optimal manner. By comparing the measures for the MPC truck, Table 7.13 with the benchmark in Table 7.11 it is easy to tell that the MPC is more fuel efficient, especially when following a light truck. This shows that a more efficient solution to the example in Section 1.1.1 has been found. It can therefore be concluded that energy performance can be improved in the case of platooning without V2V by utilizing models of both the controlled and the preceding truck, under the assumption of no surrounding traffic. However, the state of the art controlled solo driving HDVs are today very fuel efficient, refer to Table 7.9, and to beat their performance on non-flat sections of road by platooning without V2V is still a challenging task. This indicates that the advantages of utilizing a look-ahead control strategy in non-flat sections may sometimes outweigh the benefits of operating in a platooning fashion.
Appendix
### Table A.1: Numerical values of relevant constants in the vehicle model.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_w$</td>
<td>Wheel radius</td>
<td>0.522 [m]</td>
</tr>
<tr>
<td>$i_{fd}$</td>
<td>Final drive ratio</td>
<td>2.59 [-]</td>
</tr>
<tr>
<td>$i_{G,12}$</td>
<td>Gear 12 ratio</td>
<td>1.55 [-]</td>
</tr>
<tr>
<td>$i_{G,13}$</td>
<td>Gear 13 ratio</td>
<td>1.24 [-]</td>
</tr>
<tr>
<td>$i_{G,14}$</td>
<td>Gear 14 ratio</td>
<td>1.00 [-]</td>
</tr>
<tr>
<td>$\eta_{G,12}$</td>
<td>Gear 12 efficiency</td>
<td>0.974 [-]</td>
</tr>
<tr>
<td>$\eta_{G,13}$</td>
<td>Gear 13 efficiency</td>
<td>0.974 [-]</td>
</tr>
<tr>
<td>$\eta_{G,14}$</td>
<td>Gear 14 efficiency</td>
<td>0.974 [-]</td>
</tr>
<tr>
<td>$\eta_{fd}$</td>
<td>Final drive efficiency</td>
<td>0.98 [-]</td>
</tr>
<tr>
<td>$J_w$</td>
<td>Wheel Inertia</td>
<td>32.9 [kgm$^2$]</td>
</tr>
<tr>
<td>$J_e$</td>
<td>Engine Inertia</td>
<td>3.5 [kgm$^2$]</td>
</tr>
<tr>
<td>$A_a$</td>
<td>Vehicle Frontal Area</td>
<td>10 [m$^2$]</td>
</tr>
<tr>
<td>$\rho_a$</td>
<td>Density of Air (at STP)</td>
<td>1.2754 [kg/m$^3$]</td>
</tr>
<tr>
<td>$c_r$</td>
<td>Rolling resistance coefficient</td>
<td>1.5e-3 [-]</td>
</tr>
<tr>
<td>$c_D$</td>
<td>Aerodynamic drag coefficient</td>
<td>0.6 [-]</td>
</tr>
<tr>
<td>$\tau_G$</td>
<td>Gear shift time constant</td>
<td>1.5 [s]</td>
</tr>
<tr>
<td>$\theta_{th}$</td>
<td>Gear activation threshold</td>
<td>0.8 [-]</td>
</tr>
</tbody>
</table>
**Table A.2:** Numerical values of linearization points used in the modelling.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_0$</td>
<td>Linearization Velocity</td>
<td>80 [km/h]</td>
</tr>
<tr>
<td>$\tau_{hw,0}$</td>
<td>Linearization Time Headway</td>
<td>1 [s]</td>
</tr>
<tr>
<td>$m_{f,0}$</td>
<td>Linearization Mass of Fuel</td>
<td>0 [-]</td>
</tr>
<tr>
<td>$T_{e12,0}$</td>
<td>Linearization Torque Gear 12</td>
<td>266 [Nm]</td>
</tr>
<tr>
<td>$T_{e13,0}$</td>
<td>Linearization Torque Gear 13</td>
<td>332 [Nm]</td>
</tr>
<tr>
<td>$T_{e14,0}$</td>
<td>Linearization Torque Gear 14</td>
<td>412 [Nm]</td>
</tr>
<tr>
<td>$F_{b,0}$</td>
<td>Linearization Brake Force</td>
<td>0 [N]</td>
</tr>
<tr>
<td>$v_{p,0}$</td>
<td>Linearization Velocity, Preceding</td>
<td>80 [km/h]</td>
</tr>
<tr>
<td>$\alpha_0$</td>
<td>Linearization Road Angle</td>
<td>0 [rad]</td>
</tr>
</tbody>
</table>

**Controller Data**

**Table A.3:** Description of controller tuning parameters and nominal values.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_s$</td>
<td>Controller Sampling Time</td>
<td>1 [s]</td>
</tr>
<tr>
<td>$N$</td>
<td>Prediction Horizon</td>
<td>100</td>
</tr>
<tr>
<td>$Q$</td>
<td>State Penalty</td>
<td>$\begin{bmatrix} 1 &amp; 0 &amp; 0 \ 0 &amp; 10 &amp; 0 \ 0 &amp; 0 &amp; 1 \end{bmatrix}$</td>
</tr>
<tr>
<td>$R_\sigma$</td>
<td>Slack Penalty</td>
<td>$\begin{bmatrix} 1000 &amp; 1000 \end{bmatrix}$</td>
</tr>
<tr>
<td>$R_{\text{shift}}$</td>
<td>Shift Penalty</td>
<td>$\begin{bmatrix} 1000 &amp; 1000 &amp; 1000 \end{bmatrix}$</td>
</tr>
<tr>
<td>$M$</td>
<td>Move-blocking Horizon</td>
<td>50</td>
</tr>
</tbody>
</table>
### Constraints Data

*Table A.4: Numerical values of relevant constraints in the controller optimization problem.*

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_{\text{min}}$</td>
<td>Minimum velocity</td>
<td>40 [km/h]</td>
</tr>
<tr>
<td>$v_{\text{max}}$</td>
<td>Maximum velocity</td>
<td>85 [km/h]</td>
</tr>
<tr>
<td>$\tau_{\text{hw,min}}$</td>
<td>Minimum time headway</td>
<td>1.0 [s]</td>
</tr>
<tr>
<td>$\tau_{\text{hw,max}}$</td>
<td>Maximum time headway</td>
<td>10.0 [s]</td>
</tr>
<tr>
<td>$T_{\text{e,min}}$</td>
<td>Minimum engine torque</td>
<td>-200 [Nm]</td>
</tr>
<tr>
<td>$T_{\text{e,max}}$</td>
<td>Maximum engine torque</td>
<td>2400 [Nm]</td>
</tr>
<tr>
<td>$F_{b,\text{min}}$</td>
<td>Minimum brake force</td>
<td>0 [N]</td>
</tr>
<tr>
<td>$F_{b,\text{max}}$</td>
<td>Maximum brake force</td>
<td>120 [kN]</td>
</tr>
</tbody>
</table>
Bibliography


