CHALMERS
UNIVERSITY OF TECHNOLOGY

Vehicle dynamics control using dimensionless MPC

Applying dimensional analysis to lateral position control and torque vectoring

Master's thesis in Systems, Control and Mechatronics

Josip Kir Hromatko

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

Master's thesis 2021

Vehicle dynamics control using dimensionless MPC

Applying dimensional analysis to lateral position control
and torque vectoring

JOSIP KIR HROMATKO

Department of Electrical Engineering
Division of Systems and Control
Mechatronics research group
Chalmers University of Technology
Gothenburg, Sweden 2021

Vehicle dynamics control using dimensionless MPC
Applying dimensional analysis to lateral position control and torque vectoring
JOSIP KIR HROMATKO

Supervisor: Šandor Ileš, Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia
Examiner: Paolo Falcone, Department of Electrical Engineering, Chalmers University of Technology, Sweden

Master's Thesis 2021
Department of Electrical Engineering
Division of Systems and Control
Mechatronics research group
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A scaled model of a Mercedes-Benz GLA, taken from [1].

Typeset in LaTeX
Gothenburg, Sweden 2021

Vehicle dynamics control using dimensionless MPC
Applying dimensional analysis to lateral position control and torque vectoring
JOSIP KIR HROMATKO
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Modern electric cars often come with various advanced control systems. However, developing and testing these systems requires significant resources. In this thesis, the aim was to investigate if dimensional analysis, an approach used to simplify experiments and reduce expenses, can be applied to vehicle dynamics control. This was done by transforming two common vehicle models, the bicycle model and the dual-track model, into dimensionless ones. Then, model predictive controllers were designed for lateral position control and torque vectoring applications, based on the dimensionless models. The lateral position controller was simpler, based on the bicycle model, providing good results independent of vehicle parameters. The torque vectoring one, based on the dual-track model, requires some further testing, but the initial experiments look promising. Finally, further exploiting the benefits of dimensional analysis could lead to significant savings in both the academia and the automotive industry.

Keywords: vehicle dynamics, dimensional analysis, model predictive control, lateral position control, torque vectoring

Acknowledgements

I would like to thank Šandor Ileš for the patience, advice and guidance and everyone at the Laboratory for Mechatronic Systems in Zagreb for pleasant discussions about this project. Also, I would like to thank Paolo Falcone and the entire staff at Chalmers for their understanding and teaching; I learned a great deal during my studies in Sweden. Finally, I would like to thank my family and friends for their continuous support and all the laughs.

Josip Kir Hromatko, Zagreb, September 2021

Contents

List of Figures

List of Tables

1

Introduction

1.1 Background

With the advances in technology in the recent years, automobiles are becoming increasingly more complex. Modern cars contain a large number of electronic systems, many of which have security roles. Systems such as Anti-lock Braking System (ABS), Traction Control System (TCS), Electronic Stability Control (ESC) and similar aim to improve the stability and handling characteristics of the vehicle. Also, many modern vehicles contain systems which aim to reduce energy consumption and improve the overall driving comfort.

Although the mentioned systems are used also with internal combustion vehicles, probably the biggest recent change in the automobile industry was the introduction of electric vehicles. These vehicles often have several motors, sometimes even one for each wheel, which creates many new possibilities for control algorithms and independent wheel control. This thesis focuses on electric vehicles with four independent motors and lateral vehicle dynamics control using *model predictive control*, a type of optimal control.

Testing on full scale vehicles requires significant time and resources. Therefore, many researchers use high-fidelity simulation software or scaled versions of vehicles in experimental setups. This is also the case at the Laboratory for Mechatronic Systems at the Faculty of Electrical Engineering and Computing in Zagreb, Croatia. At the Laboratory, a testbed consisting of a treadmill and a scaled remote-controlled car is being developed with the aim of testing various vehicle dynamics control algorithms.

Finally, building a scaled model and using it in experiments is pretty much useless unless it relates to a certain degree to the corresponding full scale system. In mechanical engineering, scaled models have been used for a long time, mainly in applications related to fluid mechanics, such as vessel propeller design or aerodynamic testing. This thesis aims to apply the theory used in these areas to vehicle dynamics control in order to verify that the results obtained with a scaled system can provide insight into the behaviour of a full size one.

1.2 Related work

In research related to automotive applications, a book by R. Rajamani [2] is often used as a reference for theoretical background on vehicle dynamics. Vehicle dynamics control using model predictive control has been explored by many researchers and this thesis is based on the work done by P. Falcone et al. in [3], [4] and B. Spahija et al. in [5]. These papers have also focused on independent wheel control using MPC and lateral vehicle dynamics through *lateral position control* and *torque vectoring* algorithms.

The theory behind dimensional analysis was first presented by E. Buckingham in [6], while examples of applications can be found in books by Westine et al. [7] or Munson et al. [8] for fluid mechanics in particular. Using dimensional analysis with vehicle dynamics control has been explored by S. Brennan and A. Alleyne in [9] and [10]. Finally, the experimental setup at the Faculty in Croatia and some preliminary results have been presented in [11].

1.3 Thesis aim and scope

The main goal of the thesis was to investigate how dimensional analysis can be used for vehicle dynamics control. By comparing control performance of a full size vehicle in simulations with a lab scale one on the experimental setup, one could see if the dimensionless approach has potential for future applications. This was done by implementing two control algorithms, lateral position control and torque vectoring, and evaluating the performance of the dimensionless controller in comparison to the dimensional one.

Although they were initially planned as additional goals, embedded system implementation and testing for robustness were not done within the scope of this thesis. Also, the performance of the developed linear MPC controller was not compared with that of a nonlinear one. These goals remain to be of interest in future work.

2

Theory

2.1 Vehicle dynamics

This section describes the models commonly used for vehicle dynamics control. They are based on certain simplifications which make them unsuitable for simulations where very high fidelity is required. However, these simplifications also make them convenient for control system development and real-time applications.

2.1.1 The bicycle model

One of the most commonly used models for lateral vehicle dynamics control is the so-called *bicycle model* or *single-track model* [2]. This model, shown in Figure 2.1, considers two degrees of freedom - the vehicle's position with respect to the center of rotation and its orientation with respect to the global coordinate frame. Also, pairs of wheels are lumped into a single one and the longitudinal speed of the vehicle is assumed to be constant.

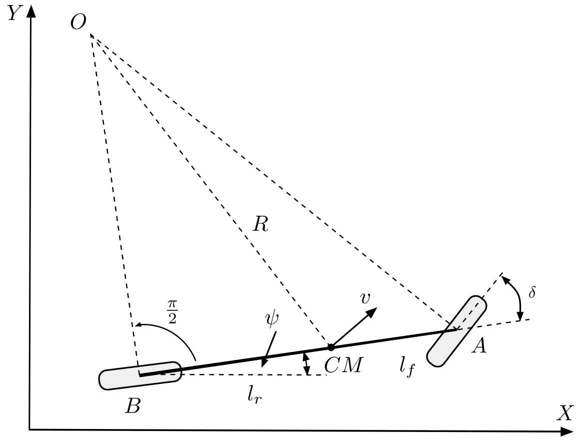


Figure 2.1: The bicycle model [11].

The state-space formulation of the model is linear and given by:

$$\frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & 0 & -V_x - \frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{I_zV_x} & 0 & -\frac{2C_{\alpha f}l_f^2+2C_{\alpha r}l_r^2}{I_zV_x} \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2C_{\alpha f}l_f}{I_z} \end{bmatrix} \delta \qquad (2.1)$$

with variables and parameters as shown in Table 2.1.

Table 2.1: Variables and parameters in the bicycle model.

y	lateral distance from the center of rotation to vehicle centerline
ψ	orientation w.r.t. the global coordinate system
$C_{\alpha f}$	front tire cornering stiffness
$C_{\alpha r}$	rear tire cornering stiffness
m	vehicle mass
I_z	vehicle moment of inertia around the z-axis
l_f	distance from CoG to front axle
l_r	distance from CoG to rear axle
V_x	longitudinal speed
δ	wheel steering angle

2.1.2 The dual-track model

The linear bicycle model is fairly simple and easy to use in control algorithms for many applications such as lane keeping or electronic stability control. However, for torque vectoring, a more complex model is needed. Since the aim with torque vectoring is to control the driven wheels independently, a four-wheel model must be used. Also, if wheel torque is to be controlled, wheel dynamics should be included. Figure 2.2 shows the *dual-track model*, which can be used for this purpose.

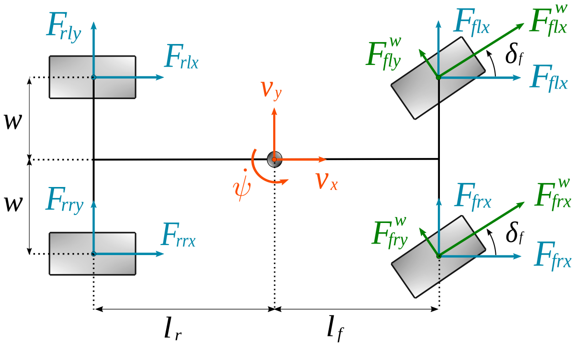


Figure 2.2: The dual-track model [11].

By setting up the equilibrium equations for vehicle body translation in x- and y-axes, rotation around the z-axis and wheel rotation, vehicle dynamics can be described by the following expressions:

$$
\begin{aligned}
m\dot{v}_x &= m\dot{\psi}v_y + F_{flx} + F_{frx} + F_{rlx} + F_{rrx} && \text{(2.2a)}\\
m\dot{v}_y &= -m\dot{\psi}v_x + F_{fly} + F_{fry} + F_{rly} + F_{rry} && \text{(2.2b)}\\
J_z\ddot{\psi} &= l_f(F_{fly} + F_{fry}) - l_r(F_{rly} + F_{rry}) + w(-F_{flx} + F_{frx} - F_{rlx} + F_{rrx}) && \text{(2.2c)}\\
J_w\dot{\omega}_{\bullet\star} &= T_{\bullet\star} - r_w F^w_{\bullet\star x}, \qquad \bullet \in \{f, r\}, \quad \star \in \{l, r\} && \text{(2.2d)}
\end{aligned}
$$

where v_x, v_y and $\dot{\psi}$ denote the longitudinal, lateral and angular velocities of the chassis. $T_{\bullet\star}$ and $\omega_{\bullet\star}$ denote the torque and angular velocity of each wheel. Also, m is the vehicle's mass, J_z the moment of inertia around the yaw axis, w half of the track width, l_f and l_r the distances from the front and rear axles to the center of mass. Wheel moment of inertia is denoted by J_w and radius by r_w.

The forces acting on the chassis are calculated as:

$$
\begin{aligned}
F_{f\star x} &= F^w_{f\star x}\cos\delta_{f\star} - F^w_{f\star y}\sin\delta_{f\star}, \qquad F_{r\star x} = F^w_{r\star x} && \text{(2.3a)}\\
F_{f\star y} &= F^w_{f\star x}\sin\delta_{f\star} + F^w_{f\star y}\cos\delta_{f\star}, \qquad F_{r\star y} = F^w_{r\star y} && \text{(2.3b)}
\end{aligned}
$$

where $\delta_{f\star}$ denotes the left/right wheel steering angle. The forces at the wheels are generally functions of tire slip s_x and slip angle α, road friction μ and vertical load F_z:

$$
\begin{aligned}
F^w_{\bullet\star x} &= f_x(s_{\bullet\star x}, \alpha_{\bullet\star}, \mu_{\bullet\star}, F_{\bullet\star z}) && \text{(2.4a)}\\
F^w_{\bullet\star y} &= f_y(s_{\bullet\star x}, \alpha_{\bullet\star}, \mu_{\bullet\star}, F_{\bullet\star z}) && \text{(2.4b)}
\end{aligned}
$$

Tire slips are defined as:

$$
s_{\bullet\star x} = \frac{r_w\omega_{\bullet\star}}{v^w_{\bullet\star x}} - 1, \qquad v^w_{\bullet\star x} \neq 0 \tag{2.5}
$$

and slip angles as:

$$
\alpha_{\bullet\star} = \arctan\left(\frac{v^w_{\bullet\star y}}{v^w_{\bullet\star x}}\right) \tag{2.6}
$$

Individual wheel velocities can be calculated as:

$$
\begin{aligned}
v^w_{f\star x} &= v_{f\star x}\cos\delta_f + v_{f\star y}\sin\delta_f, \qquad v^w_{r\star x} = v_{r\star x} && \text{(2.7a)}\\
v^w_{f\star y} &= -v_{f\star x}\sin\delta_f + v_{f\star y}\cos\delta_f, \qquad v^w_{r\star y} = v_{r\star y} && \text{(2.7b)}
\end{aligned}
$$

with

$$
\begin{aligned}
v_{flx} &= v_x - w\dot{\psi}, \qquad v_{fly} = v_y + l_f\dot{\psi} && \text{(2.8a)}\\
v_{frx} &= v_x + w\dot{\psi}, \qquad v_{fry} = v_y + l_f\dot{\psi} && \text{(2.8b)}\\
v_{rlx} &= v_x - w\dot{\psi}, \qquad v_{rly} = v_y - l_f\dot{\psi} && \text{(2.8c)}\\
v_{rrx} &= v_x + w\dot{\psi}, \qquad v_{rry} = v_y - l_f\dot{\psi} && \text{(2.8d)}
\end{aligned}
$$

If certain assumptions are made (small acceleration, flat road, low aerodynamic effects), load transfer can be neglected and the vertical tire forces can be approximated by the static load distribution on the two axles:

$$F_{f\star z} = \frac{mgl_r}{2(l_f + l_r)} \; , \qquad F_{r\star z} = \frac{mgl_f}{2(l_f + l_r)} \tag{2.9}$$

Finally, the model can be compactly written in state-space form as:

$$\dot{\xi} = f(\xi, u) \tag{2.10a}$$

$$\eta = h(\xi) \tag{2.10b}$$

with $\xi = [v_x \; v_y \; \dot{\psi} \; \omega_{fl} \; \omega_{fr} \; \omega_{rl} \; \omega_{rr}]^\top$ as the state vector, $u = [T_{fl} \; T_{fr} \; T_{rl} \; T_{rr}]^\top$ as the input vector and η as the output vector. The output mapping $h(\xi)$ depends on the tracked (referenced) states.

Although many other effects such as suspension dynamics, rolling resistance, combined slip and mechanical losses are ignored, this model describes the most important phenomena for passenger cars during normal driving. For simulating a vehicle, a more advanced model should be used, but the one presented here is suitable for designing a control system.

2.1.3 Tire modelling

The tire is one of the most important components of a vehicle, whose motion is mainly dictated by the forces in the four contact points between the tires and the ground. Also, tire models are generally nonlinear and describing the tire perfectly is very difficult. It is therefore necessary to understand the basics of how a tire behaves and be aware of the simplifications made during the modelling.

2.1.3.1 The linear model

The simplest tire model describes the tire with just two numbers. It assumes that the longitudinal and lateral tire forces are proportional to tire slips and slip angles:

$$F^w_{\bullet\star x} = C_x s_{\bullet\star x}, \qquad F^w_{\bullet\star y} = -C_y \alpha_{\bullet\star} \tag{2.11}$$

The proportionality constants C_x and C_y are commonly referred to as the *slip stiffness* and *cornering stiffness*. The negative sign for the lateral forces comes from the coordinate system convention. This model describes the tire well only in a certain region of small slips and slip angles. If the tire operates outside of this region, a more advanced and nonlinear model must be used.

2.1.3.2 The "magic" formula

Almost 40 years ago, Hans Pačejka proposed probably the most well-known empirical tire model [12]. It is formed as a combination of trigonometric functions and, in its simplest form, describes the tire forces with eight parameters:

$$F^w_{\bullet\star x} = D_x \sin(C_x \arctan(B_x s_{\bullet\star x} - E_x(B_x s_{\bullet\star x} - \arctan(B_x s_{\bullet\star x})))) \tag{2.12a}$$

$$F^w_{\bullet\star y} = D_y \sin(C_y \arctan(B_y \alpha_{\bullet\star} - E_y(B_y \alpha_{\bullet\star} - \arctan(B_y \alpha_{\bullet\star})))) \tag{2.12b}$$

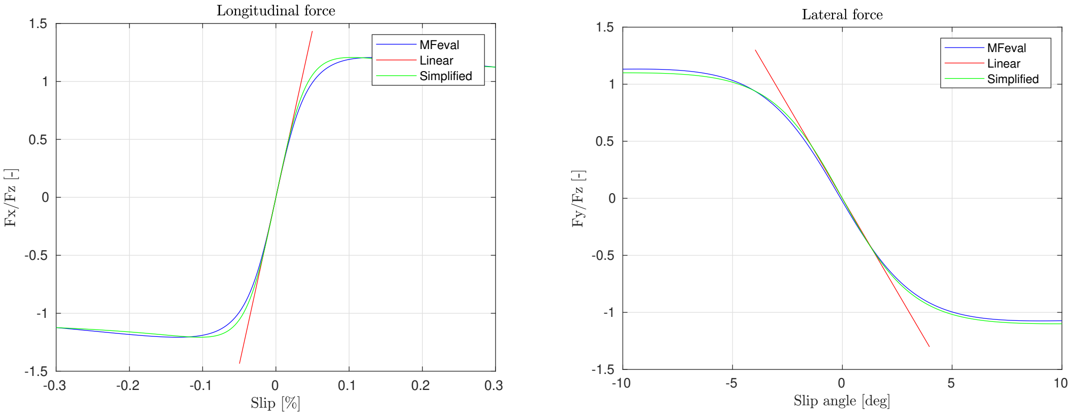
The full model contains many additional parameters which need to be fitted to the tire testing data. It also accounts for various effects needed for high-fidelity simulations, which made it popular among both the researchers and car manufacturers.

2.1.3.3 Linear approximation

If the parameters of a full, nonlinear tire model are available, the linear model parameters (2.11) can be found by looking at the tire characteristic around the origin. For small slips and slip angles, the two coefficients are equal to the derivatives of the force curves at the origin:

$$C_x = \left.\frac{\partial F_x}{\partial s_x}\right|_{s_x=0}, \qquad C_y = \left.\frac{\partial F_y}{\partial \alpha}\right|_{\alpha=0} \tag{2.13}$$

Figure 2.3 shows the full nonlinear tire model ("magic" formula, denoted with MFeval) for a certain tire and fitted approximations using the simplified "magic" formula and a linear model.



(a) Longitudinal force.

(b) Lateral force.

Figure 2.3: Tire forces - nonlinear model and approximations.

2.2 Dimensional analysis

The idea of conducting experiments on a small scale system to gain knowledge about a large scale one is often implemented in mechanical engineering, especially fluid mechanics. With dimensional analysis, the behaviour of a system is described in such a way that the equations hold true regardless of the absolute values of the system's parameters - only their relations are important. This allows for drawing conclusions about large systems by working on their scaled versions, which can in some cases lead to significant savings in time and resources.

2.2.1 The Buckingham Π theorem

In the early twentieth century, Edgar Buckingham sparked interest in the subject of dimensional analysis through his publications. Although he was not the first one to investigate it, he formulated the theorem which is the foundation of dimensional analysis, the so-called Π *theorem* [6].

Theorem (Buckingham Π theorem)**.** *If an equation involving k variables is dimensionally homogeneous, it can be reduced to a relationship among k-r independent dimensionless products, where r is the minimum number of reference dimensions required to describe the variables.*

The theorem mainly answers the question of the number of parameters needed to completely describe a system (degrees of freedom). Essentially, it states that any equation of the form:

$$u_1 = f(u_2, u_3, \dots, u_k) \tag{2.14}$$

can be rearranged into the form:

$$\Pi_1 = g(\Pi_2, \Pi_3, \dots, \Pi_{k-r}) \tag{2.15}$$

using combinations of u_i to form Π_i, which are dimensionless terms. The theorem does not provide the function g, but it might be significantly easier to obtain g than f through experimentation. Also, the obtained dimensionless terms Π_i indicate under which terms two systems follow the same dimensionless equations.

2.2.2 Dynamic similarity

The idea behind dimensional analysis can also be applied when differential equations describing the system dynamics are known. In that case, it is necessary to normalize (divide) all variables and parameters of one kind with a reference value of the same kind – speed with speed, time with time, etc. – in order to obtain relative variables. The reference values should be formed from fixed system parameters.

By doing so, physical quantities will be replaced by dimensionless ones, multiplied by dimensionless numbers, i.e., ratios. These ratios will give information about the conditions under which system similarity holds true. Finally, if two systems have the same dimensionless governing equations, they are *dynamically similar*. Several examples can be found in books on similarity in engineering [7] or those on fluid mechanics [8].

2.3 Model predictive control

Most of the controllers used today are PID controllers. They are easy to understand, implement and tune. However, for some applications, the requirements such as energy optimization or actuator constraints cannot be met efficiently by such a simple control law.

Model predictive control is a type of *optimal control*. As the name suggests, it uses a *model* of the system to make *predictions* about its state trajectory and find the best (optimal) *control* inputs in order to achieve the desired behaviour. It also allows for including more knowledge about the system in the model (compared to PID control) and extends easily to systems with multiple inputs and multiple outputs.

2.3.1 The moving horizon idea

The main idea behind MPC is to control systems by iteratively solving optimization problems online. In every time step, a prediction of the system's behaviour over a finite *prediction horizon* is obtained based on the current state and the future control inputs applied over the *control horizon*. The control sequence which gives the best performance with respect to a specified *objective* or *cost function* is found. Then, the first element in this optimal sequence is applied to the system. In the next time step, the whole process is repeated again. Figure 2.4 illustrates the main idea for one iteration.

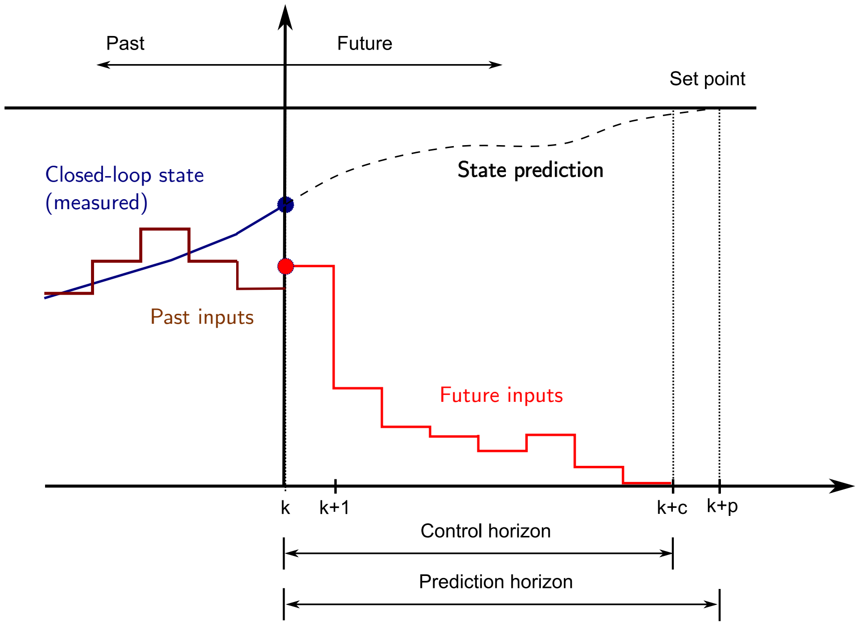


Figure 2.4: One instant of a moving horizon.

For an unconstrained system with nonlinear dynamics, the optimization problem to solve can generally be formulated as:

$$\min_{u} \quad J(\xi, u) \tag{2.16a}$$

$$\text{s.t.} \quad \xi^{+} = f(\xi, u), \quad \xi(0) = \xi_0 \tag{2.16b}$$

The objective function J can be specified in many ways, depending on the goal such as energy optimization, reference tracking etc. The state vector ξ and input vector u can (in theory) be arbitrarily large - the limitations on the optimization problem size are imposed by the hardware available for solving it. This also applies to the prediction and control horizon lengths.

2.3.2 Constraints

In most applications there are certain *constraints* that should not be violated. For example, actuators cannot operate beyond their limitations or certain states should not exceed allowed levels (especially important in safety-critical applications). In the MPC framework, it is relatively simple to include these limitations by adding constraints to the optimization problem:

$$\min_{u} \quad J(\xi, u) \tag{2.17a}$$

$$\text{s.t.} \quad \xi^{+} = f(\xi, u), \quad \xi(0) = \xi_0 \tag{2.17b}$$

$$\xi(k) \in \Xi, \quad u(k) \in \mathbb{U}, \quad \text{for all } k \tag{2.17c}$$

By doing this, the controller will never demand more than the actuator can give. On the other hand, hard constraints on system states can lead to infeasible optimization problems (e.g., if the system is already in the forbidden state), but there are methods for tackling this issue as well.

One possible solution is to use *soft constraints*. The main idea is to define new variables as deviations from the desired limits and penalize these variables in the objective function:

$$\min_{u,\varepsilon} \quad J(\xi, u, \varepsilon) \tag{2.18a}$$

$$\text{s.t.} \quad \xi^{+} = f(\xi, u), \quad \xi(0) = \xi_0 \tag{2.18b}$$

$$\xi(k) + \varepsilon(k) \in \Xi, \quad u(k) \in \mathbb{U}, \quad \text{for all } k \tag{2.18c}$$

With that, the optimization problem will have a solution and the controller will try not to exceed the allowed limits (how hard, depends on the objective function).

2.3.3 Linear time-varying MPC

In general, nonlinear optimization is computationally demanding despite the advances in optimization algorithms, especially in real-time applications. However, methods for solving convex optimization problems, and quadratic programs in particular, are already being used in embedded systems.

A common way to solve nonlinear optimization problems is to convert them to *linear time-varying* ones, where the nonlinear functions are iteratively approximated by linear functions and these approximations are used in the optimization instead. The solutions are accurate only in a certain region around the approximation point (i.e., locally) so special care needs to be taken.

Linearizing a nonlinear model is usually done with a first-order Taylor expansion:

$$f(\xi, u) \approx f(\xi_0, u_0) + \underbrace{\left.\frac{\partial f}{\partial \xi}\right|_{(\xi_0, u_0)}}_{A_0} (\xi - \xi_0) + \underbrace{\left.\frac{\partial f}{\partial u}\right|_{(\xi_0, u_0)}}_{B_0} (u - u_0) \tag{2.19a}$$

$$= A_0 \xi + B_0 u + \underbrace{d(\xi_0, u_0)}_{f(\xi_0, u_0) - A_0 \xi_0 - B_0 u_0} \tag{2.19b}$$

where the last term represents the difference between evaluating the nonlinear and linear models.

With LTV MPC, the system's dynamics are linear, but changing over time as well as the prediction horizon:

$$\xi_{k+1} = A_k(t)\xi_k + B_k(t)u_k + d_k(t) \tag{2.20a}$$

$$y_k = C_k(t)\xi_k \tag{2.20b}$$

First, a nominal trajectory $\hat{\Xi} = [\hat{\xi}_0, \hat{\xi}_1, \ldots, \hat{\xi}_N]^\top$ starting from the current state, $\hat{\xi}_0 = \xi_0$, and applying a known input u_0 (e.g., constant or shifted from the previous step) is obtained by integrating the nonlinear model (2.10). Then, the Jacobians A_k and B_k at time t can be calculated as:

$$A_k = \left.\frac{\partial f}{\partial \xi}\right|_{\xi_k, u_k}, \qquad B_k = \left.\frac{\partial f}{\partial u}\right|_{\xi_k, u_k} \tag{2.21}$$

The last term in (2.19) can be calculated by:

$$d_k = \hat{\xi}_{k+1} - A_k\hat{\xi}_k - B_k u_k \tag{2.22}$$

and the LTV model finally becomes:

$$\xi_{k+1} = A_k\xi_k + B_k u_k + d_k \tag{2.23}$$

where ξ_k and u_k represent the state predictions and optimal control inputs from the previous step.

In order to obtain a convex optimization problem, the objective function should be quadratic or linear and the constraints linear:

$$\min_{u,\varepsilon} \quad J(\xi, u, \varepsilon) = \sum_k \left\|\xi_k - \xi_k^{ref}\right\|_Q^2 + \|u_k\|_R^2 + \|\Delta u_k\|_S^2 + p\,\|\varepsilon_k\|^2 \tag{2.24a}$$

$$\text{s.t.} \quad \xi^+ = A_k\xi_k + B_k u_k + d_k, \quad \xi(0) = \xi_0 \tag{2.24b}$$

$$\xi(k) + \varepsilon(k) \in \Xi, \quad u(k) \in \mathbb{U}, \quad \text{for all } k \tag{2.24c}$$

where Q, R and S are positive semi-definite matrices, p is a scalar and A_k, B_k, d_k are calculated by linearizing the system around (ξ_0, u_0) [5].

Figure 2.5 illustrates the described process. Green circles mark the state predictions and optimal inputs from the previous step, which are used to calculate the Jacobian matrices A_i, B_i.

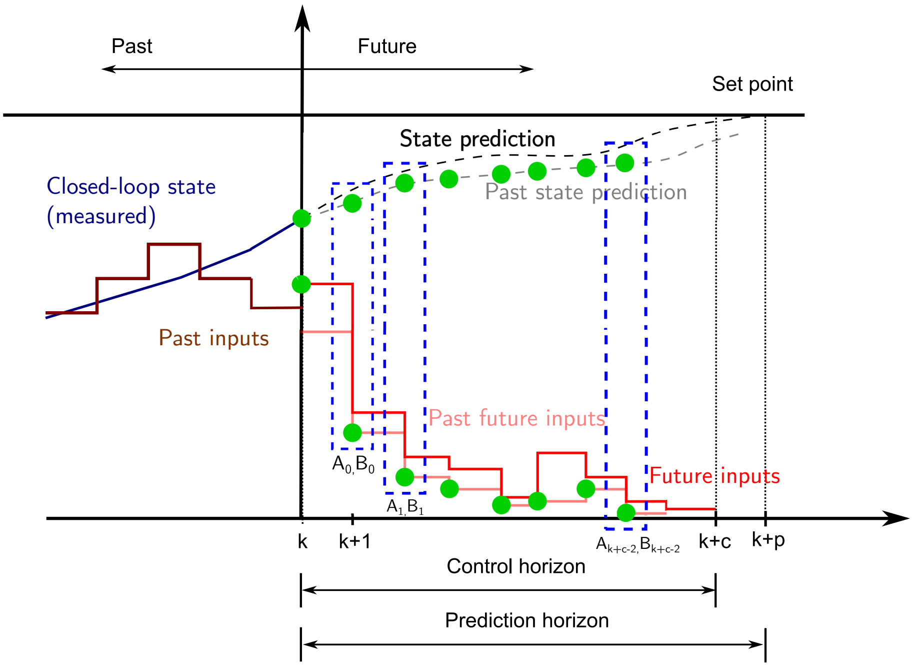


Figure 2.5: A visualization of LTV MPC.

2.3.4 Tuning

With a PID controller, the tuning parameters are the three gains, one for each of the proportional, integral and derivative terms. In MPC, the controller is tuned by changing the weighting matrices in the objective function. The bigger the weight for a specific term, the more emphasis is put on minimizing that term.

For example, if the control input weight matrix (R in (2.24)) is very large, even small inputs to the system will result in a large objective function value. This will result in the controller avoiding large control inputs and reaching the reference slowly. On the other hand, if the weights on the control input are small and the ones on referenced states are large, the controller will be more aggressive in following the reference despite the energy consumption. As usual, there will be a trade off between the response time and the used energy. Very often, the actual values of the weight matrices are less relevant compared to the ratios between them.

3

Methods

3.1 Lateral position control

Probably the first topic that comes to mind when talking about lateral vehicle dynamics is controlling the lateral position. This is a well researched problem and can be done relatively well with a simple bicycle model. It was therefore chosen as the first step in testing non-dimensional MPC.

3.1.1 Parameters

To verify the idea of dynamical similarity with a simple model, a bicycle model (2.1) of the vehicle was used. The required parameters for a full size vehicle were taken from a CarMaker's demo vehicle, while those for a lab size vehicle were taken from one of the lab vehicles developed at the Faculty in Zagreb, Croatia, shown in Figures 3.1 and 3.2. The longitudinal speeds were chosen so that the vehicles become dynamically similar (see next section). Both sets are given in Table 3.1.

Table 3.1: Bicycle model parameters for the full size and lab size vehicles.

symbol	description	full size	lab size
$C_{\alpha f}$	front tire cornering stiffness $[N/rad]$	72705	8.25
$C_{\alpha r}$	rear tire cornering stiffness $[N/rad]$	72705	8.25
m	vehicle mass $[kg]$	1600	1.173
I_z	vehicle moment of inertia around the z-axis $[kgm^2]$	2394	0.0337
l_f	distance from CoG to front axle $[m]$	1.311	0.1115
l_r	distance from CoG to rear axle $[m]$	1.311	0.141
V_x	longitudinal speed $[m/s]$	16.67	2.12
δ_{max}	maximal steering angle $[deg]$	27.69	15.01
$\dot{\delta}_{max}$	maximal steering angle rate $[deg/s]$	27.69	15.01

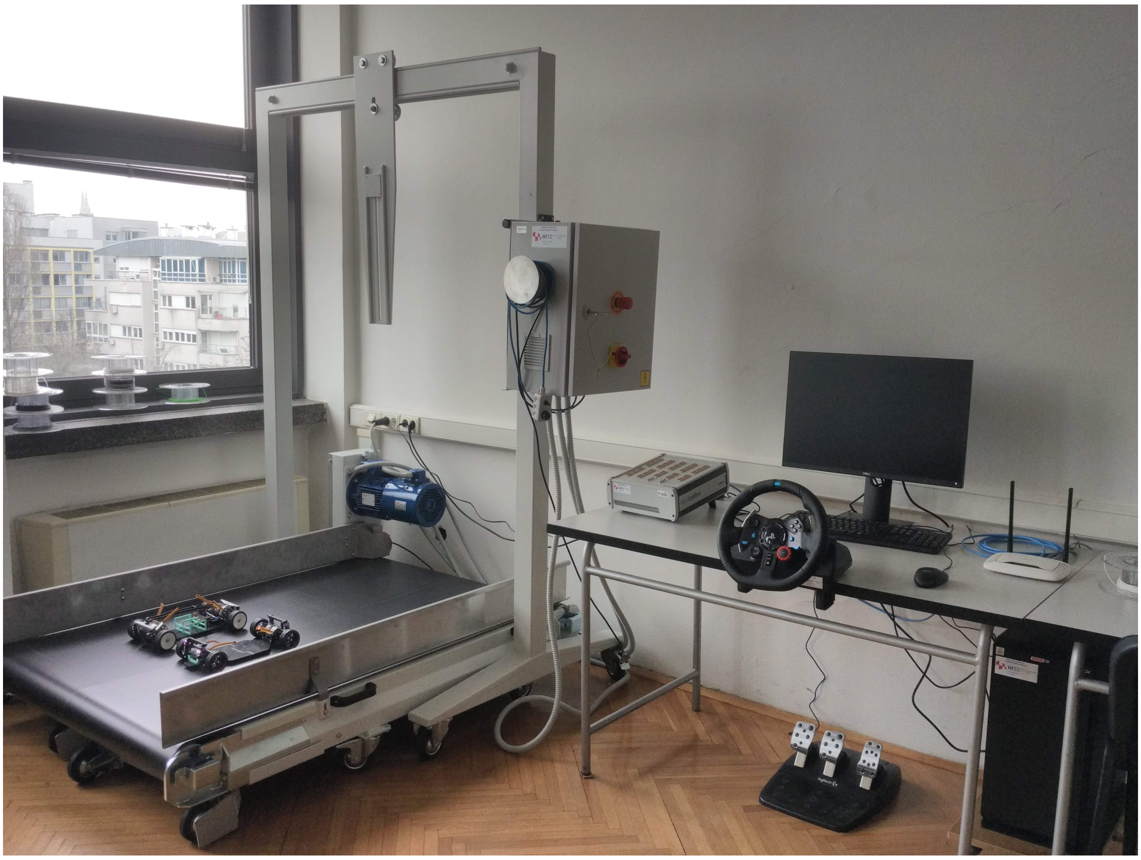


Figure 3.1: The experimental setup at the Faculty in Zagreb, Croatia [11].

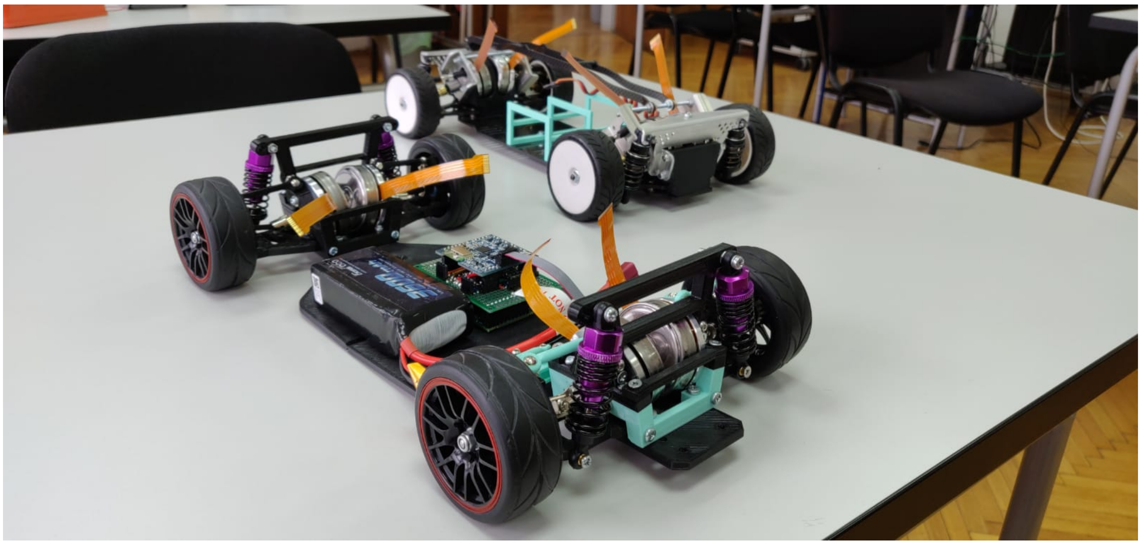


Figure 3.2: The two scaled models used in the experimental setup [11].

3.1.2 Similarity

As shown in [9], the bicycle model can easily be converted to the dimensionless form due to its linearity. If the vehicle's wheelbase, $L = l_f + l_r$, and longitudinal speed, V_x, are chosen as reference parameters, the state vector $\xi = [y\ \dot{y}\ \psi\ \dot{\psi}]^\top$ can be replaced by:

$$\xi = M_\xi \xi^*, \qquad M_\xi = \mathrm{diag}\,[L\ V_x\ 1\ V_x/L] \tag{3.1}$$

where M_ξ is the state transformation matrix and $\xi^* = [y^*\ \dot{y}^*\ \psi^*\ \dot{\psi}^*]^\top$ is the dimensionless state vector. Although given in radians, angles are commonly thought of as dimensionless (an angle in radians is actually the ratio of a circular arc and a radius) so input normalization is not required, i.e., $u^* = u$. The state derivative $\dot{\xi}$ can also be rewritten by normalizing time:

$$\frac{d\xi}{dt} = \frac{d(M_\xi \xi^*)}{dt} = \frac{V_x}{L} M_\xi \frac{d\xi^*}{dt^*} \tag{3.2}$$

Finally, the original system:

$$\frac{d\xi}{dt} = A\xi + Bu \tag{3.3}$$

can be written in the dimensionless form as:

$$\frac{d\xi^*}{dt^*} = \frac{L}{V_x} M_\xi^{-1} A M_\xi \xi^* + \frac{L}{V_x} M_\xi^{-1} B u^* = A^* \xi^* + B^* u^* \tag{3.4}$$

To check if the full size and lab size vehicles are dynamically similar, one can compare the corresponding π-groups, which are formed by normalizing the vehicle's parameters to obtain dimensionless terms [9]. The π-groups are given in Table 3.2.

Table 3.2: Bicycle model π-groups for the full size and lab size vehicles.

π-group	full size	lab size	lab size, +1 kg
l_f/L	0.5	0.449	0.449
l_r/L	0.5	0.551	0.551
$C_{\alpha f}\frac{L}{mV_x^2}$	0.395	0.402	0.396
$C_{\alpha r}\frac{L}{mV_x^2}$	0.395	0.402	0.396
$I_z\frac{1}{mL^2}$	0.218	0.439	0.237

The values in the first two columns are relatively similar, except for the last one. If the mass of the lab vehicle is increased by 1 kg, the values in the last column are obtained. With these parameters, dynamic similarity can be checked by comparing the eigenvalues of the non-dimensional state transition matrix (A^*), corresponding to the system's poles. The non-dimensional non-zero eigenvalues for the full size vehicle are -1.58 and -1.82 and for the lab size vehicle $-1.637 \pm 0.5563\jmath$, which suggests that their dynamics are similar. Although the lab size vehicle's poles form a complex conjugate pair, step response plots show that the oscillatory behaviour is negligible.

3.1.3 Controller design

3.1.3.1 Reference generation

For comparing the two differently sized vehicles, a test manoeuvre with parameters relative to the vehicle parameters was used. It consists of a sine steer after a short straight drive. The wavelength of the sinusoid was chosen as 32 times the vehicle's length, while the amplitude was equal to the vehicle's length. The wavelength was calculated from the assumed longitudinal speed and the desired period of around 5 seconds. The amplitude was chosen to be larger than track width to correspond to a highway manoeuvre, but also to make it slightly more challenging. Figure 3.3 shows the reference path in dimensionless coordinates.

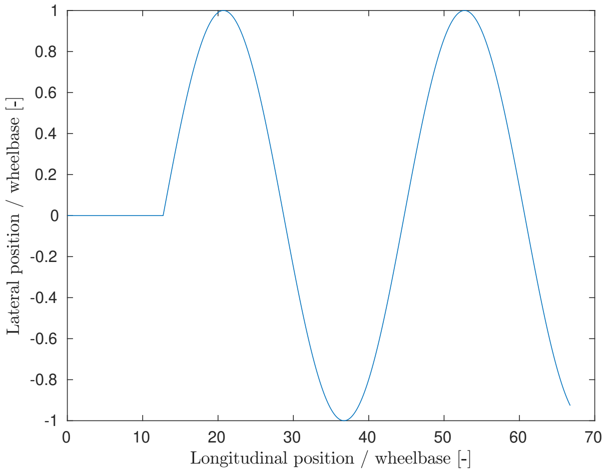


Figure 3.3: Normalized lateral position reference.

3.1.3.2 Linear quadratic regulator

The LQR is a *state-feedback* controller of the form:

$$u(t) = -K\xi(t) \tag{3.5}$$

which minimizes the quadratic cost function:

$$J(\xi, u) = \int_0^\infty (\xi^\top Q\xi + u^\top Ru + 2\xi^\top Nu)dt \tag{3.6}$$

subject to the system dynamics $\dot{\xi} = A\xi + Bu$. The controller gain K can be found by solving the associated *algebraic Riccati equation*:

$$A^\top S + SA - (SB + N)R^{-1}(B^\top S + N^\top) + Q = 0 \tag{3.7}$$

for S and calculating the gain from:

$$K = R^{-1}(B^\top S + N^\top) \tag{3.8}$$

If dimensionless forms of system dynamics are used, A^* and B^*, the LQR gains for both the full size and lab size vehicles should be almost the same. Indeed, this was the case for the two set of parameters considered - the resulting gains and closed loop poles are given in Table 3.3.

Table 3.3: LQR gains and closed-loop poles.

parameter	full size	lab size
K_1	1	1
K_2	0.5549	0.6176
K_3	1.2519	1.1958
K_4	0.4670	0.4963
pole pair 1	$-0.5210 \pm 0.7495j$	$-0.5260 \pm 0.7136j$
pole pair 2	$-1.8227 \pm 0.3587j$	$-1.7279 \pm 0.6199j$

3.1.3.3 MPC

With the bicycle model, designing an MPC controller is fairly straightforward. The dynamics are linear and the only additional constraints are the steering system limitations. Switching between dimensional and dimensionless models can easily be done according to (3.1). Since the input is an angle (dimensionless), no input transformation is needed before applying the optimization result to the system. The objective function and the constraints used were:

$$\min_{\xi,\delta} \quad J(\xi,\delta) = \sum_k \left\| \xi_k - \xi_k^{ref} \right\|_Q^2 + \|\delta_k\|_r^2 + \|\Delta\delta_k\|_s^2 \tag{3.9a}$$

$$\text{s.t.} \quad \xi^+ = A\xi_k + B\delta_k, \quad \xi(0) = \xi_0 \tag{3.9b}$$

$$|\delta_k| < \delta_{max}, \quad |\Delta\delta_k| < \Delta\delta_{max}, \quad \text{for all } k \tag{3.9c}$$

For the simulations, a sampling time of 10 ms was assumed. On a laptop, the optimization took 1-2 ms on average. The sampling time was chosen as a few times larger to simulate running on an embedded system. The bicycle model is linear with four states and one input, which allows for a fast optimization and a relatively small sampling time. The prediction horizon was set to $N = 50$ (corresponding to 0.5 s of preview) and the control horizon to $M = 5$ with a last value hold.

Lateral position was the only tracked state, with a weight of $Q_y = 10^6$. With the dimensionless model, the objective function term relating to lateral position was transformed using (3.1) and the weight becomes $Q^* = M_\xi^\top Q M_\xi$. Apart from the system dynamics, constraints were set on the input (steering angle) and its rate. Both were also penalized with the weight $r = s = 10^{-6}$.

3.1.4 Simulation environment

The bicycle model was generally used as a first step in testing the hypothesis of dynamical similarity. The focus was on obtaining qualitative results, i.e., roughly comparing the simulations instead of looking for a perfect match. Therefore, the simulations were done using YALMIP for optimization problem modelling [13] and OSQP [14] as the solver in MATLAB, with the same bicycle model as for controller design, which neglects many phenomena such as combined slip, load transfer, aerodynamics etc.

3.2 Torque vectoring

Along with lateral position control, another focus area in vehicle dynamics control is maintaining a desired yaw rate, i.e., ensuring stability and preventing the vehicle from drifting or spinning out. This can be done, e.g., by controlling the brakes independently or modifying the steering angle electronically. One of the main advantages of electric vehicles with multiple motors is the ability to control each of them independently, a strategy commonly known as *torque vectoring.* If this is done well, the torque acting on the vehicle around the z-axis can be modified and both the stability and performance of the vehicle can be improved.

3.2.1 Parameters

For torque vectoring with MPC and four independent motors, a dual-track model (2.2) was used. The parameters were again taken from CarMaker's demo vehicle and the lab vehicle in Zagreb. The linear model (2.11) was used for tire modelling (in the prediction model). The additional required parameters in comparison with the bicycle model 3.1 were the tire longitudinal stiffness, vehicle track width, wheel radius and inertia and motor limitations. The full sets of parameters for both vehicles are given in Table 3.4.

Table 3.4: Dual-track parameters for a full size and a lab size vehicle.

symbol	description	full size	lab size
m	vehicle mass $[kg]$	1600	1.173
I_z	vehicle moment of inertia around the z-axis $[kgm^2]$	2394	0.0337
l_f	distance from CoG to front axle $[m]$	1.311	0.1115
l_r	distance from CoG to rear axle $[m]$	1.311	0.141
w	vehicle track width $[m]$	1.586	0.163
C_x	tire slip stiffness $[N]$	116377	25
C_y	tire cornering stiffness $[N/rad]$	72705	8.25
r_w	wheel radius $[m]$	0.318	0.0325
I_w	wheel moment of inertia $[kgm^2]$	1.22	$4 \cdot 10^{-5}$
T_{max}	maximal motor torque $[Nm]$	250	0.05
$\dot{T}_{max}$	maximal motor torque rate $[Nm/s]$	1000	0.2

3.2.2 Similarity

In order to obtain a dimensionless model, a similar approach as with the bicycle model was used. First, the system dynamics are linearized as described in (2.19). Then, the linearized dynamics are transformed using (3.4) with the addition of input normalization, $u = M_u u^*$, to obtain a dimensionless linear model. The transformation matrices can be obtained by following the approach in [9], which results in:

$$M_\xi = \text{diag}\left[\sqrt{\frac{C_x L}{m}},\ \sqrt{\frac{C_x L}{m}},\ \sqrt{\frac{C_x m}{L}},\ \sqrt{\frac{C_x m}{L}},\ \sqrt{\frac{C_x m}{L}},\ \sqrt{\frac{C_x m}{L}},\ \sqrt{\frac{C_x m}{L}}\right] \tag{3.10a}$$

$$M_u = C_x L \cdot I_{n_u} \tag{3.10b}$$

where $L = l_f + l_r$ is the vehicle's wheelbase, I_n is the identity matrix of size n and n_u is the number of control inputs. This model is finally discretized with a dimensionless sampling time, $T_s^* = \sqrt{C_x/mL} \cdot T_s$, giving the discrete-time matrices needed for the LTV-MPC scheme.

3.2.3 Controller design

3.2.3.1 Reference generation

The first step in the control pipeline is to "translate" the driver commands (throttle, brake, steering) into state references. For standard driving situations, a kinematic bicycle model is used for this purpose. In most basic test maneuvers, the reference longitudinal speed, v_x^{ref}, is kept constant. Then, the desired yaw rate can be calculated from the steering angle δ_f as:

$$\dot{\psi}^{ref} = \frac{v_x^{ref}}{l_f + l_r} \tan \delta_f \tag{3.11}$$

Finally, the reference lateral speed, v_y^{ref}, is often set to zero, which completes the reference vector with three tracked states, $\eta^{ref} = [v_x^{ref}\ v_y^{ref}\ \dot{\psi}^{ref}]^\top$. With a controller that outputs motor torques, wheel speeds do not need to be tracked - this would be the task of a traction control system, for example.

3.2.3.2 MPC

With an LTV model, the optimization problem to be solved in every time step was formulated as in (2.24). After the solution to the problem was found, the first control input was applied to the system.

The first constraint in the prediction model were the state dynamics, which are described by the dual-track model. Then, the actuator limitations were taken into account by limiting the available torque on the driven wheels and its rate. Finally, since the tires are approximated with a linear model which is accurate only in a certain region of small slip, soft state constraints were added to limit the slip and avoid the nonlinear dynamics. This range was determined from the plot of the tire model approximation, shown in Figure 2.3. The optimization problem was formulated as:

$$\min_{u,\varepsilon} \quad J(\xi, u, \varepsilon) = \sum_k \left\| \xi_k - \xi_k^{ref} \right\|_Q^2 + \|u_k\|_R^2 + \|\Delta u_k\|_S^2 + p \, \|\varepsilon_k\|^2 \tag{3.12a}$$

$$\text{s.t.} \quad \xi^+ = A_k \xi_k + B_k u_k + d_k, \quad \xi(0) = \xi_0 \tag{3.12b}$$

$$|\xi_k| < \varepsilon_k, \quad |u_k| < u_{max}, \quad |\Delta u_k| < \Delta u_{max}, \quad \text{for all } k \tag{3.12c}$$

with $Q = \text{diag}[10^2 \ 0 \ 10^4]$, $R = 10^{-3} \cdot I_4$, $S = 10^{-2} \cdot I_4$ and $p = 100$. The sampling time was set to 50 ms to allow for running the optimization, the prediction horizon was set to $N = 10$ and the control horizon to $M = 3$. With the dimensionless model, the cost function matrices were transformed into $Q^* = M_\xi^\top Q M_\xi$, $R^* = M_u^\top R M_u$ and $S^* = \hat{t}^{-2} \cdot M_u^\top S M_u$, where $\hat{t} = \sqrt{mL/C_x}$.

3.2.4 Simulation environment

With the dual-track model and torque vectoring, the goal was to obtain results with a higher fidelity. Therefore, a full nonlinear tire model (the "magic" formula) was used, along with a 3DOF vehicle model. The tire model parameters were taken from the `.tir` file provided by CarMaker. In this initial work, aerodynamics were considered (also with parameters from CarMaker) and the suspension dynamics were neglected.

The simulations were done in Simulink using YALMIP [13] with OSQP [14] as the solver and the Vehicle Dynamics Blockset by Mathworks [15].

4

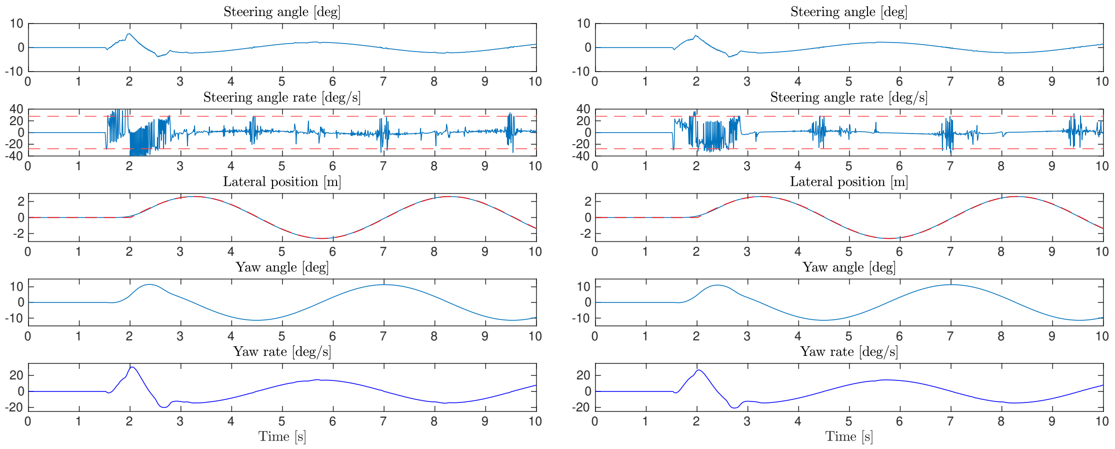
Results

4.1 Lateral position control

For testing lateral position control, a sine steer manoeuvre as shown in Figure 3.3 was used, with a sampling time of 0.01 s and simulation time of 10 s. First, *nondimensionalization* was tested by comparing simulations with a dimensional and a dimensionless prediction model for both the full size and lab scale vehicles. Then, *transferability* was evaluated by comparing the dimensionless states and inputs for the two vehicles, which should be similar according to the theory.

4.1.1 Nondimensionalization

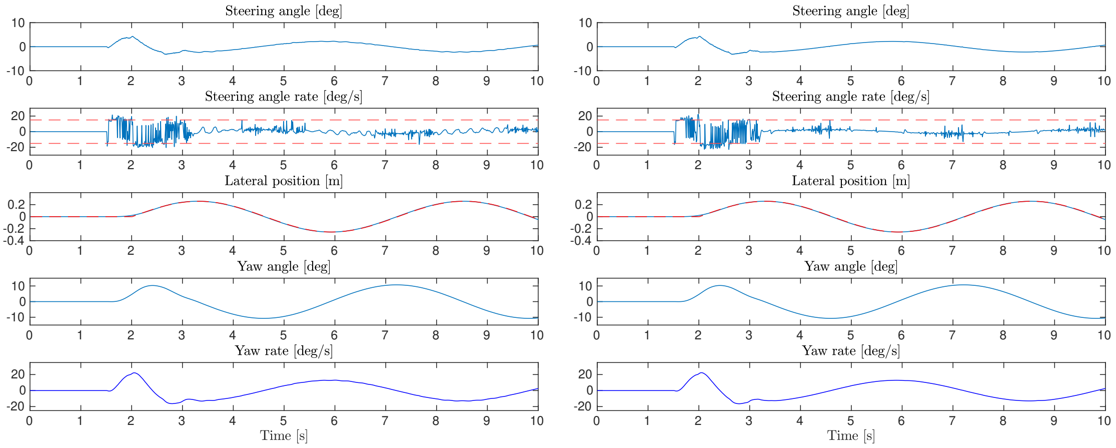
Figures 4.1 and 4.2 show the results obtained with using a dimensional and a non-dimensional prediction model in the controller, for both the full size and lab scale vehicle parameters. The manoeuvre consisted of driving straight for two seconds and then initiating a sine steer. Constraints on the input and input rate are shown with the dashed red lines, which is also used for the reference lateral position.



(a) The dimensional model.

(b) The non-dimensional model.

Figure 4.1: Lateral control comparison with the two models (full size).



(a) The dimensional model.

(b) The non-dimensional model.

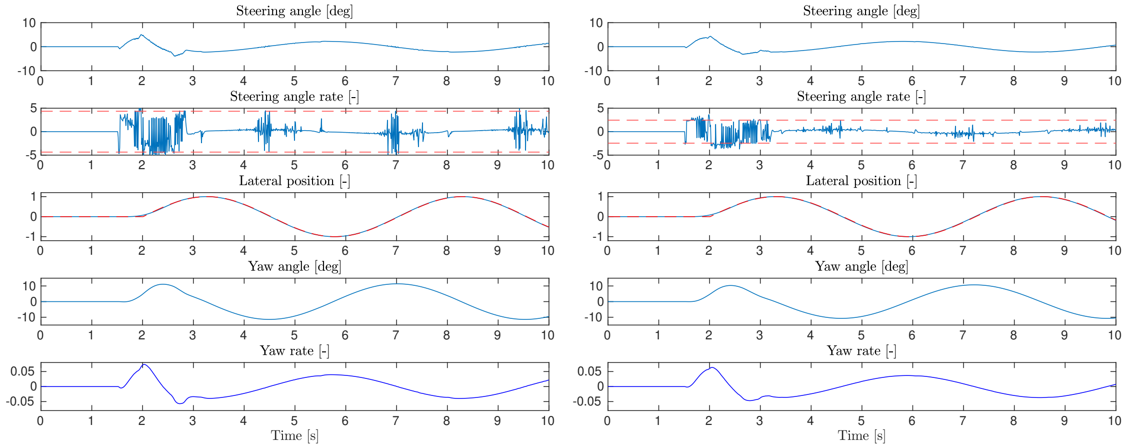
Figure 4.2: Lateral control comparison with the two models (lab scale)

The plots indicate that very similar results can be obtained with either model, both for the full size and the lab scale vehicle. Although the simulation in this case is not very realistic, the expected behaviour can be verified and the vehicle seems to follow the reference well.

It is interesting to note that the steering angle yaw rate exceeds the specified limits in some time steps. This is probably due to the solver's algorithm, which can sometimes "bend" the constraints slightly. A possible solution for this would be to decrease the allowed range so that the actual desired limits are not reached. Also, the cost function weights for the input rate could be increased to achieve a similar effect (although respecting the constraints is not guaranteed in that case).

4.1.2 Transferability

The next step in lateral position control evaluation was to compare the dimensionless states of the two vehicles, which should again be similar if dynamic similarity is achieved. Figure 4.3 shows the normalized states and inputs for the two vehicles, with a slight difference in steering angle rate limits due to different vehicle parameters. As expected, the dimensionless states are very similar, which indicates that the controller can operate independent of the vehicle scale.



(a) Full size parameters.

(b) Lab scale parameters.

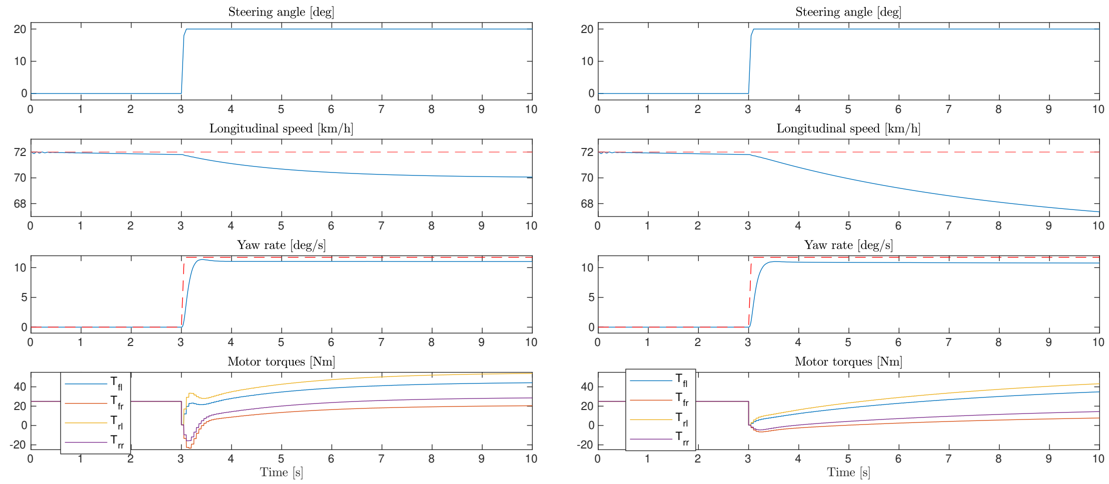
Figure 4.3: Inputs and non-dimensional states, full size and lab scale.

4.2 Torque vectoring

After verifying the basic assumptions with the bicycle model and lateral position control, the goal was to test the torque vectoring algorithm. The initial plan was to simulate the full size vehicle in Simulink or CarMaker and compare the results with the experimental setup built at the Faculty in Zagreb during the last year. The main reason was that there was no possibility of testing the algorithm on a real full size vehicle. On the other hand, modelling the lab-sized vehicle to a level needed for Simulink/CarMaker would require a significant amount of time and experimentation.

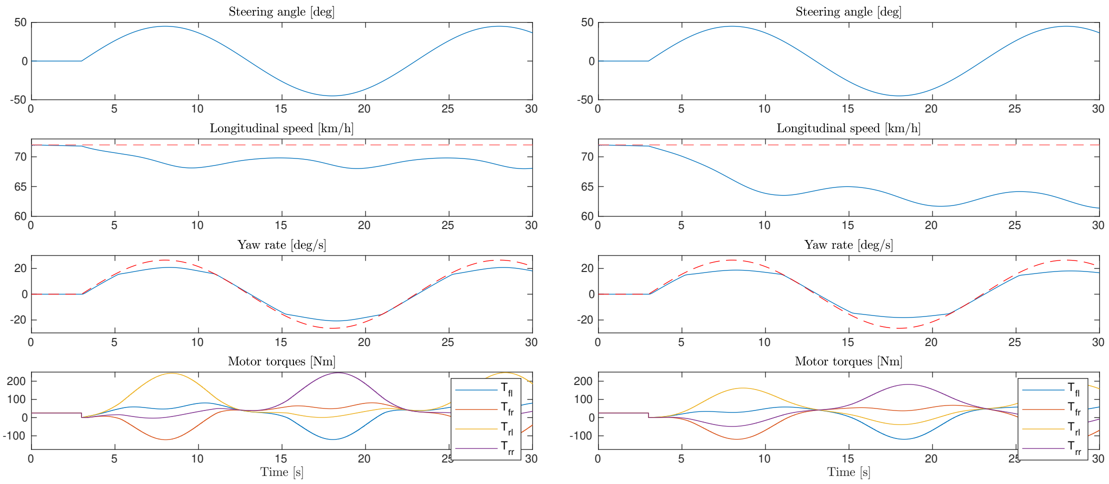
Unfortunately, hardware issues in the late stage of thesis work made it difficult to obtain experimental results on the lab-size vehicle in good time. Also, the full lab-size tire model is still unknown and modifying proprietary Simulink blocks is not straightforward. This section therefore presents only the results obtained for the full-size vehicle.

The torque vectoring algorithm was tested on two driving scenarios, a step steer and a sine steer from the lateral position control testing (Figure 3.3). The inputs and states of interest (longitudinal speed and yaw rate) are shown in Figures 4.4 and 4.5. The dashed red line marks reference values, which were held constant for the longitudinal speed and calculated according to (3.11) for the yaw rate. Note that, in the simulations done in Simulink, the coordinate system is oriented with the y-axis to the right and z-axis downwards.



(a) The dimensional model. **(b)** The non-dimensional model.

Figure 4.4: Torque vectoring comparison with the two models, step steer.



(a) The dimensional model. **(b)** The non-dimensional model.

Figure 4.5: Torque vectoring comparison with the two models, sine steer.

Although the yaw rate tracking seems similar, there are some differences in the optimal torque calculated by the controller. This could be due to the LTV MPC method, in which the nonlinear dynamics are linearized. This might, along with the state and cost function transformation, result in an optimization problem with a different minimum. Also, the longitudinal speed decreases more when dimensionless MPC is used.

The controllers generally perform well and manage to track the reference. Further testing is needed to find the cause of the mentioned differences, but the overall results indicate that nondimensionalization could be possible and applicable for torque vectoring.

5

Conclusion

5.1 Overall results

Within this project, dimensional analysis was applied to both the bicycle model and the dual-track model with realistic vehicle parameters. With the dimensionless models obtained by a linear transformation, it was shown that differently sized vehicles have similar dynamic equations if certain relations between their parameters are satisfied. Then, two MPC controllers were developed and tested. The first one, a lateral position controller, produced positive results similar to those in [11]. The second one, used for torque vectoring, is slightly more difficult to evaluate, but looks promising after a few initial experiments. Due to multiple reasons, the thesis work was not conducted exactly as planned. Several issues with implementation in both software and hardware caused delays and reduced the intended thesis output. However, the results that were obtained confirm that dimensional analysis can be used with MPC and vehicle dynamics control, demonstrate that the controllers give similar results independent of the scale of the vehicle and show the potential of further research.

5.2 Future work

As with almost every project, there are several things that can be considered as the next step in the direction of this thesis. The first one would be to complete the torque vectoring evaluation by either modelling the lab-size vehicle in more detail in Simulink/CarMaker or working on the experimental setup. Another area to investigate could be the comparison of a nonlinear MPC scheme (with a nonlinear dimensionless model) and the presented LTV MPC scheme.

Additionally, more extensive work could be done on the theoretical background of dimensional analysis and MPC, e.g., showing analytically under which conditions two dynamical closed-loop systems become equivalent. Finally, it would be interesting to look into the possibilities of robust or adaptive MPC with the dimensionless model. As shown in [9], designing a controller in the dimensionless state space with possibly smaller variations could allow for an easier use of the same controller in different vehicles.

Bibliography

[1] Oagana, A., 2014. *Mercedes-Benz GLA Gets Scaled Down as a Model Car.* [online] Autoevolution. Available at: `https://www.autoevolution.com/news/mercedes-benz-gla-gets-scaled-down-as-a-model-car-81517.html` [Accessed 26 August 2021].

[2] Rajamani, R., 2012. *Vehicle dynamics and control.* New York: Springer.

[3] Falcone, P., Borrelli, F., Tseng, H., Asgari, J. and Hrovat, D., 2008. Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation. *International Journal of Robust and Nonlinear Control*, 18(8), pp.862-875.

[4] Falcone, P., Borrelli, F., Asgari, J., Tseng, H. and Hrovat, D., 2007. Predictive Active Steering Control for Autonomous Vehicle Systems. *IEEE Transactions on Control Systems Technology*, 15(3), pp.566-580.

[5] Spahija, B., Švec, M., Matuško, J. and Ileš, Š., 2019. Successive Linearization Based Predictive Vehicle Torque Vectoring. *2019 International Conference on Electrical Drives & Power Electronics (EDPE)*, pp.267-271.

[6] Buckingham, E., 1914. On Physically Similar Systems; Illustrations of the Use of Dimensional Equations. *Physical Review*, 4(4), pp.345-376.

[7] Westine, P., Dodge, F. and Baker, W., 2014. *Similarity Methods in Engineering Dynamics.* Amsterdam: Elsevier Science.

[8] Munson, B., Okiishi, T., Huebsch, W. and Rothmayer, A., 2013. *Fundamentals of fluid mechanics.* 7th ed. John Wiley & Sons, Inc.

[9] Brennan, S. and Alleyne, A., 2001. Robust Scalable Vehicle Control via Non-Dimensional Vehicle Dynamics. *Vehicle System Dynamics*, 36(4-5), pp.255-277.

[10] Brennan, S. and Alleyne, A., 2005. Dimensionless robust control with application to vehicles. *IEEE Transactions on Control Systems Technology*, 13(4), pp.624-630.

[11] Makarun, P., Josipović, G., Švec, M. and Ileš, Š., 2021. Testing predictive

vehicle dynamics control algorithms using a scaled remote-controlled car and a roadway simulator. *2021 International Conference on Electrical Drives & Power Electronics (EDPE)* (Accepted for publication). IEEE.

[12] Pačejka, H., 2002. *Tyre and vehicle dynamics.*
Oxford: Butterworth-Heinemann.

[13] Löfberg, J., 2004. YALMIP: a toolbox for modelling and optimization in MATLAB. *In Proceedings of the CACSD Conference*, Taipei, Taiwan.

[14] Stellato, B., Banjac, G., Goulart, P., Bemporad, A. and Boyd, S., 2020. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4), pp.637-672.

[15] Mathworks.com. 2021. *Vehicle Dynamics Blockset Documentation.* [online] Available at: `https://www.mathworks.com/help/vdynblks/` [Accessed 30 July 2021].

[16] Boyd, S. and Vandenberghe, L., 2011. *Convex optimization.* Cambridge: Cambridge Univ. Pr.

[17] Rawlings, J. and Mayne, D., 2009. *Model predictive control.* Madison: Nob Hill Publishing.