

CZECH TECHNICAL UNIVERSITY IN PRAGUE

**Faculty of Mechanical Engineering
Department of Instrumentation and Control Engineering**



Master's thesis

**Controller design for improving lateral
vehicle dynamics**

Návrh regulátoru pro zlepšení příčné dynamiky vozidla

I hereby declare that this thesis is my own original work and has not been submitted before to any institution for assessment purposes. Furthermore, I have acknowledged all sources used and have cited these in the reference section.

Date:

.....

Signature

I. Personal and study details

Student's name: **Velínský Ondřej** Personal ID number: **473548**
Faculty / Institute: **Faculty of Mechanical Engineering**
Department / Institute: **Department of Instrumentation and Control Engineering**
Study program: **Automation and Instrumentation Engineering**
Specialisation: **Automation and Industrial Informatics**

II. Master's thesis details

Master's thesis title in English:

Controller design for improving lateral vehicle dynamics

Master's thesis title in Czech:

Návrh regulátoru pro zlepšení příčné dynamiky vozidla

Guidelines:

1. get acquainted with function of the torque vectoring system (used in FSE.X monopost)
2. create a simplified model of lateral dynamics of the race car
3. qualitatively validate the model
4. design a controller for (sub)optimal stabilization of lateral dynamics
5. evaluate function of the system

Bibliography / sources:

- [1] VLK, František. Dynamika motorových vozidel. 2. vyd. Brno: František Vlk, 2003. ISBN 80-239-0024-2.
[2] Kiencke, Uwe, and Lars Nielsen. Automotive control systems: for engine, driveline, and vehicle. 2000. ISBN 3-540-23139-0.

Name and workplace of master's thesis supervisor:

Ing. Jaroslav Bušek, Ph.D. U12110.3

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **29.04.2022** Deadline for master's thesis submission: **12.08.2022**

Assignment valid until: _____

Ing. Jaroslav Bušek, Ph.D.
Supervisor's signature

Head of department's signature

doc. Ing. Miroslav Španiel, CSc.
Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt

Student's signature

Abstract

With the present development in the field of electromobility, new possibilities and requirements for electronic assistance systems arise. The goal of this thesis is to propose a system for stabilization and control of lateral dynamics of a electric vehicle with four independent drives.

In order to develop a mathematical model, basic concepts and equations of vehicle dynamics are presented. The model parameters are estimated from the data measured on eForce FSE.X prototype vehicle and validated by simulation. The model is then used for synthesis of linear parameter varying model predictive controller (LPV-MPC). The performance of the controller is then successfully evaluated in Simulink as well as in driver-in-loop simulation in IPG CarMaker environment.

Anotace

Se současným rozvojem elektromobility přichází i nové možnosti a požadavky na asistenční a bezpečnostní systémy vozidel. Tato práce si klade za cíl navrhnout systém pro úpravu a stabilizaci boční dynamiky elektrického vozu s nezávislým elektrickým pohonem čtyř kol. Jsou shrnuty základní rovnice dynamiky vozidel nezbytné pro sestavení matematického modelu. Jeho parametry jsou potom zvoleny na základě dat naměřených na prototypovém voze eForce FSE.X a validovány simulacemi. Vytvořený model je použitý pro syntézu lineárního prediktivního řízení založeného na modelu s proměnnými parametry (LPV-MPC). Jeho funkce je ověřena na simulacích v programu Simulink a následně úspěšně testována v driver-in-loop (DIL) simulaci v testovacím prostředí IPG CarMaker.

Contents

1	Introduction	1
1.1	Torque vectoring	1
1.2	Formula Student	1
1.3	Vehicle overview	2
1.3.1	Parameters	3
1.3.2	System in use	4
2	Vehicle dynamics modeling	5
2.1	Equations of motion	5
2.1.1	Referential body frame	5
2.1.2	Body slip angle	6
2.2	Tires	6
2.2.1	Slip	6
2.2.2	Forces	7
2.2.3	Magic formula	8
2.2.4	Friction ellipse	8
2.2.5	Vertical tire force	9
3	Model	12
3.1	Assumptions and goals	12
3.2	Nonlinear single-track model	12
3.3	Model validation	13
3.4	Single track model in terms of β and $\dot{\varphi}$	14
4	Controller	17
4.1	Stabilization and changing dynamics	17
4.2	Reference signal	17
4.3	Model linearization	20
4.4	Discretization	22
4.5	MPC	22
4.5.1	Cost function	22
4.5.2	Prediction	23
4.5.3	Action change	24
4.5.4	Quadratic programming	24
4.5.5	Implementation	25
5	Results	27
5.1	Ramp change	27
5.1.1	LTI-MPC	29
5.2	Disturbance attenuation	31
5.3	Virtual test drive	33

6 Yaw torque distribution	37
7 Conclusion	41
List of symbols and abbreviations	45
List of used SW	47
List of attachments	48

1 Introduction

1.1 Torque vectoring

Torque vectoring is a commonly used term for a vehicle control system that is employed to vary torque bias between each half-shaft and thus create yaw torque on the chassis. Different manufacturers use different implementations of such system under various trademarks. Notable examples are Mitsubishi's S-AWC [1], ZF's Vector Drive [2] or Rimac All Wheel Torque Vectoring System [3]. While the first two use electromechanical differential, Rimac has four electric drives (one for each wheel) and therefore implements torque vectoring only in a software form.

In recent years, electric vehicles have gained a significant market share [4] and one can expect it to grow even more rapidly in the near future. Compared to a combustion engine, an electric drive can be controlled with faster response and can generate negative torque by energy recuperation. This becomes even more advantageous when each wheel can be controlled independently. Although such driveline configuration is rarely seen in production EVs, it is being pioneered by sports cars like the previously mentioned Rimac Nevera or Schaeffler's prototype for Deutsche Tourenwagen Masters electric series.

1.2 Formula Student

Formula Student is an international engineering competition for university students. The goal of the contest is to design and manufacture a new formula style vehicle every year. Each summer, multiple races take place around the world, where the teams compete in four dynamics disciplines - acceleration, skidpad, autocross and endurance.

The rules of the competition focus mainly on the safety and do not restrict any technical aspects of the car. Because of that, the teams often utilize novel technologies in order to gain an edge in the competition. Throughout the years, it was for example the use of additive manufacturing, active suspension, generative design of mechanical and aerodynamics structures, or active ground effect. Similarly, algorithms for active vehicle dynamics control are used by all of the top teams nowadays. Few examples can be found in [5], [6] and [7].

The team eForce FEE Prague Formula is based at the Czech Technical University in Prague, specifically the Faculty of Electrical Engineering. The team has developed its 10th generation of an electric formula for the 2021 season from the scratch. The formula proved itself to be very competitive by finishing 2nd in races in Spain and Croatia.

However, one of its weaknesses was its vehicle dynamics control unit (VDCU) because of the lack of development and personnel. For the 11th generation of eForce's electric vehicle,

the torque vectoring system should be revisited and the proposal of a new system is the main objective of this thesis.

1.3 Vehicle overview



Fig. 1: eForce FSE.11

The vehicle has four independently driven wheels, each implementing the same 25kW PMSM motor and gearbox. Two 2-channel Lenze Schmidhauser inverters are used to control the motors and a 600V self-designed accumulator is used to store the energy. The powertrain is capable of regenerative braking. Custom electronic control unit is used to send torque requests to the motor controllers. The core of this unit is TMS320F28377D processor which is programmed using MATLAB's Embedded Coder directly from Simulink. This unit is designed to run with period of 10 ms. CAN bus network is utilized to transmit all communication between the electronic units. All messages broadcasted on the network are logged and stored in a datalogger for further evaluation and debugging.

For lateral dynamics control it is crucial to know the vehicle's velocity vector and heading direction. These are provided by off-the-shelf inertial measurement unit (IMU) Ellipse-N from SBG Systems. It uses sensor fusion of gyroscope, accelerometers and GPS to provide precise measurement of vehicle's location including its orientation in 3D space. Since the IMU measures heading and velocity vector, it is also able to provide the body slip angle.

Steering angle is measured on the steering column by a magnetic rotary position sensor on a custom board. It measures the angular position with 0.1° resolution every 50 ms. The steering kinematics is known and the resulting steering angle of the front wheels can be calculated. The kinematic equations are rather complicated and usually simulated in a

dedicated software. However, on this particular vehicle, the steering is almost linear as can be seen on Figure 2. For simplicity, the steering geometry will be considered linear and described by its first order polynomial with equation $y = 0.284x$.

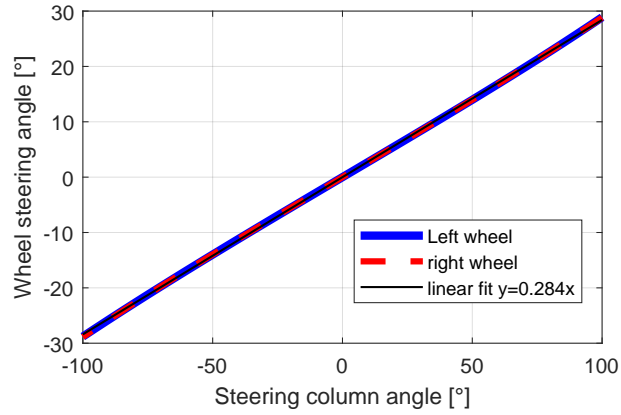


Fig. 2: Plot of the wheel steering angle as a function of the steering column angle

Throughout this thesis, following values will be used to parametrize the vehicle. Most of them have been measured by the author of this thesis, however some of them are difficult or impossible to measure without specialized equipment, so they were estimated from the data available and consulted with literature. That is the case of moment of inertia of the chassis whose value was chosen based on a CAD model and estimation validated by discussion with other Formula Student teams. Parameters of the tires were estimated based on the available Formula SAE Tire Testing Consortium dataset as will be discussed further.

1.3.1 Parameters

- $m = 260$ kg Mass of vehicle including driver
- $I_z = 80$ kg \cdot m² moment of inertia to Z axis
- $\text{CoG}_x = 0.49$ [-] relative position of center of gravity towards front axle
- $l_f = 0.747$ m Distance between CoG and center of the front axle
- $l_r = 0.778$ m Distance between CoG and center of the rear axle
- $t = 1.2$ m track, distance between left and right wheel. Is assumed to be the same on both axles.
- $\text{CoG}_z = 0.255$ m height of center of gravity
- $r_w = 0.2$ m wheel radius
- $i = 13.3$ [-] Gear ratio
- $\tau^{\max} = 29.1$ N \cdot m Single motor maximal torque
- $C_\alpha = 23000$ N/rad Cornering stiffness of tire

- $B^{\text{Pac}} = 10.55$ [-] Magic formula parameter
- $C^{\text{Pac}} = 1.347$ [-] Magic formula parameter
- $D^{\text{Pac}} = -1600$ [-] Magic formula parameter
- $E^{\text{Pac}} = 0.4464$ [-] Magic formula parameter

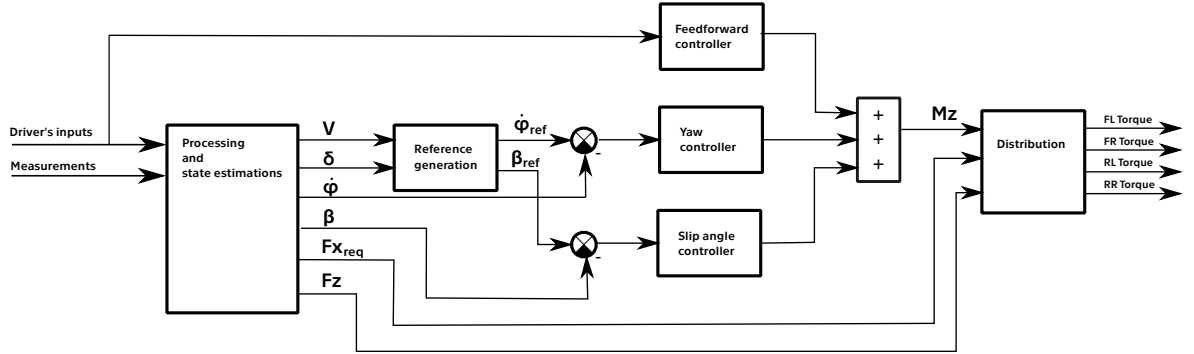


Fig. 3: Schematics of current control system

1.3.2 System in use

The system that is currently in use was primarily developed by Marek Laszlo, a former member of eForce. Detailed overview of his work can be found in his master's thesis [8]. The system follows generally used concept shown in schematic in Fig. 3 that is common for majority of implementations that can be found in literature.

Firstly, all the measured signals are processed and checked for their validity. IMU measurement is converted from north-east-down (NED) world coordinates to obtain the vehicle's speed, slip angle and yaw rate. Additional states, such as vertical wheel loads, are estimated. Signals from pedals are mapped according to the desired response.

The desired yaw rate and possibly also the side slip angle must then be calculated from the vehicle states and driver's inputs. Possible solution could be to rely solely on the steering wheel angle. However, this method would not yield good results since the vehicle lateral dynamics is also dependent on the velocity. There are several methods to predict the desired car behavior. Currently employed generator of reference is described in section 4.2.

The difference between these referential values and instantaneous measurements are then fed into yaw rate and slip angle controllers that adjust their outputs in order to minimize the error. The controller output is yaw torque M_z . PI control structure is implemented for both of these controllers.

An additional feedforward branch is added for faster response and as a backup if IMU measurement is not possible. Lastly, torque for each wheel is calculated so that the difference between left and right wheels creates requested yaw torque on the chassis.

2 Vehicle dynamics modeling

In order to establish a broader context for the controller synthesis, it is important to introduce the basic concepts and equations of vehicle dynamics. They are necessary for understanding the system and for the formulation of the dynamic model. The equations of motion are presented along with basic behavior of the tires. Estimation of vertical tire forces is also explained as the concepts are later applied to a proposal of an algorithm for yaw torque distribution in section 6.

2.1 Equations of motion

2.1.1 Referential body frame

Referential frame according to ISO 8855 is used for all the derivations in this thesis. This prevents any confusion in defining for example slip angles and accelerations and their positive directions. The system is right-handed and its positive directions and rotations can be seen on Figure 4.

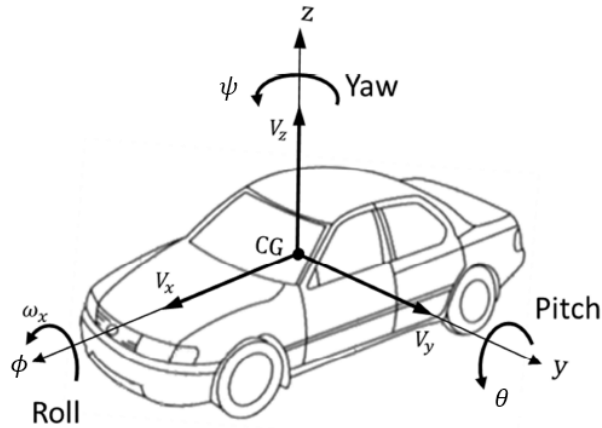


Fig. 4: Vehicle Axis System ISO 8855-2011. From [9]

Because the motion is described in referential frame which is rotating and thus non-inertial, an extra term considering the centripetal acceleration must be added to Newton's equations according to equation 1. This equation, also called transport theorem, describes the time derivative of a vector r defined in rotating coordinate system ρ in a non-rotating reference frame ξ .

$$\frac{dr}{dt}\Big|_{\xi} = \frac{dr}{dt}\Big|_{\rho} + \omega \times r \quad (1)$$

The resulting equations of motion are as follows. F_x and F_y denotes all the forces acting in x and y direction respectively.

$$m\ddot{x} - \dot{\varphi}\dot{y} = \sum F_x \quad (2a)$$

$$m(\ddot{y} + \dot{x}\dot{\varphi}) = \sum F_y \quad (2b)$$

$$I_z\ddot{\varphi} = \sum F \times r + M_z \quad (2c)$$

2.1.2 Body slip angle

Body slip angle is defined as the angle between where the vehicle is heading and the direction in which it is traveling. Mathematically this is formulated in the following equation. Note that the velocities are in the body reference frame.

$$\beta = \text{atan} \left(\frac{\dot{y}}{|\dot{x}|} \right) \quad (3)$$

2.2 Tires

Since all the driving forces acting on the vehicle body come from the tires, a solid model of their behavior must be developed. Three forces, longitudinal F_{lon} , lateral F_{lat} and vertical F_z are acting in the contact point between the tire and the road. Front tires are rotated from the body frame by steering angle δ , however the lateral and longitudinal force are described in a reference frame fixed to the tire. To obtain F_x and F_y forces parallel with axes of the chassis, they need to be rotated to match the vehicle frame.

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{bmatrix} \cos(\delta) & -\sin(\delta) \\ \sin(\delta) & \cos(\delta) \end{bmatrix} \cdot \begin{bmatrix} F_{lon} \\ F_{lat} \end{bmatrix} \quad (4)$$

2.2.1 Slip

Lateral and longitudinal forces are generally described in terms of a slip angle and a slip ratio. Both of these quantities reflect the stress in the rubber. One of them in the lateral direction, the other in the longitudinal direction. The slip angle, usually denoted α , is calculated as the angle between a heading vector and a velocity vector at the contact point between tire and ground. Velocity in the contact point consists of two components, the first one comes from the translation motion of the body and is identical for all four wheels, the other is generated by the yaw rotation of the vehicle. For the front left wheel, the formula is as follows.

$$\alpha^{FL} = \text{atan} \left(\frac{\dot{y} + l_f \cdot \dot{\psi}}{|\dot{x} - \frac{t}{2} \cdot \dot{\psi}|} \right) - \delta_{FL} \quad (5)$$

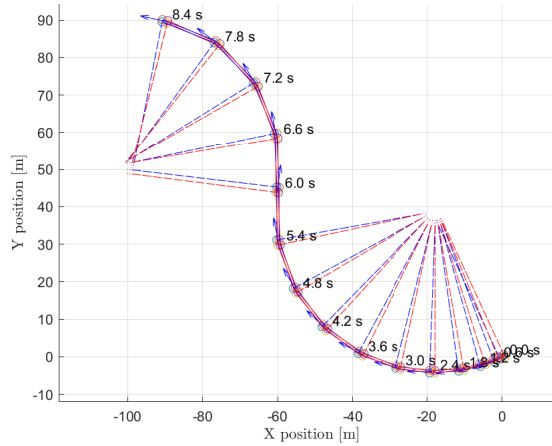


Fig. 5: Simulation showing normals to wheel speed vector during simulation

The longitudinal force of the tire depends on the so-called slip ratio which is according to SAE J670 defined by the following equation 6.[10]

$$SR = \frac{\dot{x}}{\omega_w \cdot r_w} - 1 \quad (6)$$

Slip angle is defined by a velocity vector and a referential wheel heading direction. In a steady state, the velocity vector should be tangent to instantaneous center of motion. This property was tested by a simple maneuver simulated on a twin-track vehicle model with three degrees of freedom, where constant longitudinal force was applied to all wheels and the steering angle was set to be -2 degrees with step change to 2 degrees after the first 6 seconds of simulation. Dashed line normal to vector of speed was plotted from a contact point of each wheel. Visualized time steps are chosen to be 0.6 s and do not reflect simulation step, which was variable and much finer.

As can be seen on Figure 5, all four lines in each time step meet in instantaneous center of rotation. Additionally, when the vehicle is in a steady state, normals in every time step meet in the center of the track radius. This simulation suggests that the wheel slip angle calculation is correct.

2.2.2 Forces

Forces generated by a tire can be to some extent predicted for example from finite element analysis, but for practical use they are generally physically tested in a specialized testing facility. However, it would not be feasible to conduct these tests as an individual or a single amateur racing team. That is why FSAE Tire testing consortium was founded to provide high quality data to Formula Student teams. For one time subscription fee, the teams can obtain data from various testings of racing tires measured in Calspan research facility. [11]

The test usually consists of series of slip angle or slip ratio sweeps with different values of pressure, inclination angle, load and velocity. The tires are only tested in pure slip conditions, that means lateral force is only described in dependence on the slip angle with no slip ratio, similarly the longitudinal force is measured only as the function of slip ratio with no slip angle. However, the reality is far more complex and these two forces cannot be simply added together.

2.2.3 Magic formula

So-called magic formula is an empirically developed set of equations designed by Dutch engineer Hans Bastiaan Pacejka. The equations fit the data measured on a wide range of tires. [12]. The main equation 7 is similar for the lateral and longitudinal force, only the argument changes.

For description of the force in lateral direction $\phi = F_{lat}$ and the argument of the function is the slip angle $\sigma = \alpha$. For longitudinal force $\phi = F_{lon}$ and the argument is slip ratio $\sigma = SR$. For pure slip conditions with constant vertical load and no inclination angle, B^{Pac} , C^{Pac} , D^{Pac} , E^{Pac} coefficients are constant. However, the full Pacejka's model can describe these parameters as a function of driving conditions and even works with a combined slip. Nevertheless, fitting all the parameters to the 78 equations is an exhausting process and requires specialized software.

Hoosier 10" LC0 tires utilized on FSE.X vehicle were tested by FSAE Tire testing consortium and the data are available. They were used to fit the magic formula for lateral tire force for single operating point defined by pressure 84 kPa, inclination angle 0° and $F_z = 660$ N. The result can be seen on Figure 6.

$$\phi(\sigma) = D^{Pac} \cdot \sin[C^{Pac} \cdot \text{atan}(B^{Pac} \cdot \sigma - E^{Pac}(B^{Pac} \cdot \sigma - \text{atan}(B^{Pac} \cdot x)))] \quad (7)$$

2.2.4 Friction ellipse

The simplest tool which can be used to work with combined slip is the circle of friction. It sets a boundary for total force calculated as squared sum of longitudinal and lateral force. The maximal force is the same in all directions and is equal to Coulombs friction force described by the following formula.

$$\sqrt{F_{lon}^{max} + F_{lat}^{max}} = \mu \cdot F_z \quad (8)$$

This is however just a rough approximation as due to the inner construction of tires, the friction coefficient varies in different directions. More precisely it can be approximated by an ellipse. Its size and shape can be obtained to some extent from the so-called g-g diagram

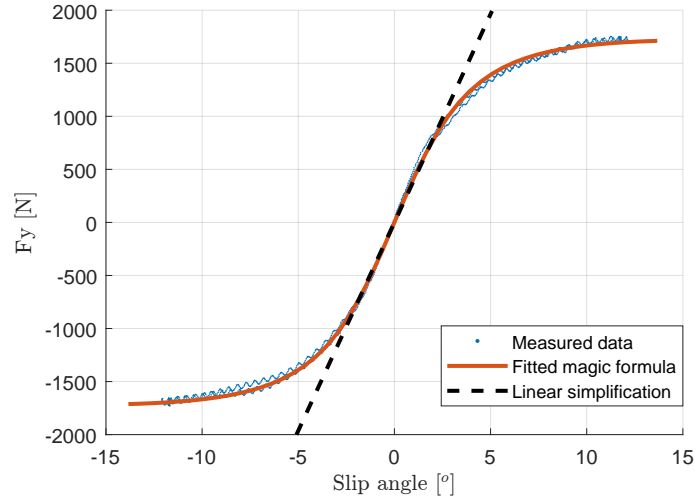


Fig. 6: Fitted tire lateral force data

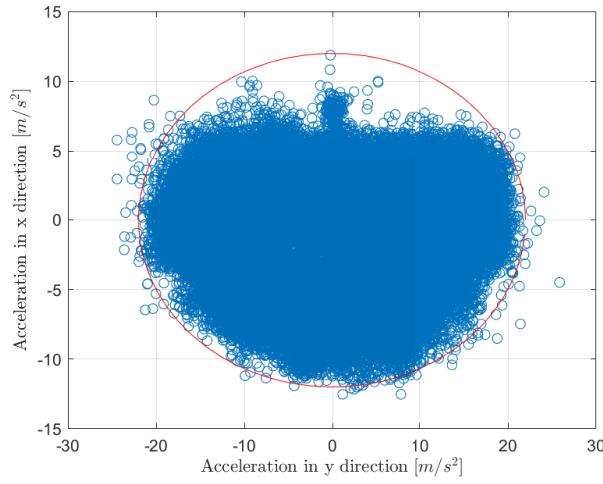


Fig. 7: G-G diagram of endurance race with estimated friction ellipse

(Figure 7), that is a plot of measured accelerations in y and x directions. In the upper part of the plot, there is an empty space caused by a power limitation.

This is a quick and easy way of obtaining some information about the tires. It would be better to measure this data without aerodynamic package, so that the total wheel load does not change with velocity. The friction ellipse can be described by the equation 9

$$\left(\frac{F_{lon}}{F_{lon}^{max}} \right)^2 + \left(\frac{F_{lat}}{F_{lat}^{max}} \right)^2 < 1 \quad (9)$$

2.2.5 Vertical tire force

Lastly, vertical force on the tires will be discussed. Three main components are assumed to be acting on a tire in vertical direction. First is the gravitational force that is distributed between the wheels of the front and rear axle in ratio given by $GoG_x = \frac{l_f}{l_f + l_r}$. Center of

gravity is located on a centerline of the vehicle, left-right static load distribution is neglected and both wheels on axle supports the same load.

$$F_{load}^{FL} = F_{load}^{FR} = (1 - CoG_x) \cdot \frac{m \cdot g}{2} \quad (10a)$$

$$F_{load}^{RL} = F_{load}^{RR} = CoG_x \cdot \frac{m \cdot g}{2} \quad (10b)$$

Formula Student car is typically designed with emphasis on aerodynamics package in order to obtain extra grip even in low speeds. In this case, the coefficient of lift C_L is obtained from CFD simulations and then validated by physical testing on a track. Lift coefficient is used in aerospace and when speaking about the downforce, its value should always be negative to avoid confusion.

For the purpose of this thesis, only the downforce generated by vehicle velocity in x direction acting in the center of pressure CoP is considered. Thanks to the symmetry of the vehicle, CoP lies on the x axis of the vehicle body and the aerodynamic force is distributed between the left and right wheel on same axle equally.

$$F_{lift} = \frac{1}{2} \cdot C_L \cdot \rho \cdot A_{ref} \cdot \dot{x}^2 \quad (11)$$

$$F_{aero}^{FL} = F_{aero}^{FR} = (1 - CoP_x) \cdot \frac{-F_{lift}}{2} \quad (12a)$$

$$F_{aero}^{RL} = F_{aero}^{RR} = CoP_x \cdot \frac{-F_{lift}}{2} \quad (12b)$$

Last considered source of tire vertical force is dynamic load transfer, both longitudinal and lateral. In both cases, the suspension dynamics are neglected. Vehicle is modeled only as a rigid body sliding on a flat plane. Tire forces act in the ground plane but the center of gravity is located above ground in the height of CoG_z . Acceleration of the CoG creates torque causing vertical reaction on the tires. The total load on wheels remains the same. For longitudinal direction, the load transfer is calculated by the formula

$$\Delta F_z^{FL} = \Delta F_z^{FR} = m \cdot \ddot{x} \cdot \frac{CoG_z}{2 \cdot (l_f + l_r)} \quad (13a)$$

$$\Delta F_z^{RL} = \Delta F_z^{RR} = -m \cdot \ddot{x} \cdot \frac{CoG_z}{2 \cdot (l_f + l_r)} \quad (13b)$$

During cornering, a centripetal force acts on the CoG and the outer wheels are more loaded than the inner ones.

$$\Delta F_z^{FL} = \Delta F_z^{RL} = m \cdot \ddot{y} \cdot \frac{CoG_z}{2t} \quad (14a)$$

$$\Delta F_z^{FR} = \Delta F_z^{RR} = -m \cdot \ddot{y} \cdot \frac{CoG_z}{2t} \quad (14b)$$

3 Model

3.1 Assumptions and goals

In order to obtain a usable model of the motion of a vehicle, significant simplifications need to be made as with any other complex system. Firstly, the vehicle will be considered to be single rigid body without suspension geometry.

Secondly, the tracks on which Formula Student competitions take place are usually flat and made from high quality tarmac without any significant bumps. Therefore, all vertical dynamics can be neglected without losing much precision.

Thirdly, following the common practice that can be seen in e.g. [13], only one wheel in the center of each axle is considered. Geometry of resultant vehicle model is on Figure 8.

Lastly, lateral dynamics are decoupled from longitudinal and the velocity is considered to be only a variable parameter. This simplification is used in literature when dealing only with lateral dynamics such as [14] or [15]. Motion equation in x direction then becomes

$$\ddot{x} = 0 \quad (15a)$$

$$\dot{x} = v_x. \quad (15b)$$

3.2 Nonlinear single-track model

By combining equations from the previous section, a nonlinear state space model can be developed. An additional yaw torque M_z that is generated by applying different torques to wheels on each side of the vehicle is added to equations. M_z is multiplied by 1000 to scale it down in order to get reasonable gain of the system. This is further beneficial for tuning of the yaw controller.

$$m(\ddot{y} + v_x \cdot \dot{\varphi}) = F_{yf} \cdot \cos(\delta) + F_{yr} \quad (16a)$$

$$I_z \ddot{\varphi} = F_{yf} \cdot \cos(\delta) \cdot l_f - F_{yr} \cdot l_r + 1000 \cdot M_z \quad (16b)$$

F_{yf} and F_{yr} are the tire forces in lateral direction generated by front and rear tire respectively, l_f and l_r describe chassis geometry according to Figure 8. The lateral tire forces depend on the wheel slip angles. In case of a single-track model they are calculated by the equations 17a and 17b.

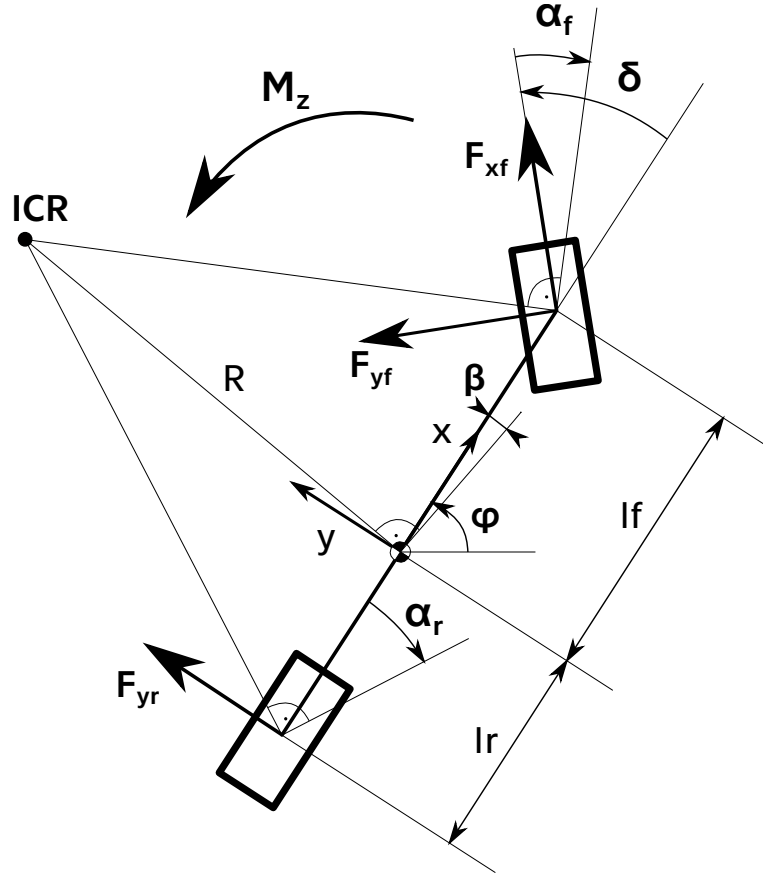


Fig. 8: Geometrical representation of a single-track model

$$\alpha_f = \text{atan} \left(\frac{\dot{y} + \dot{\varphi} \cdot l_f}{v_x} \right) - \delta \quad (17a)$$

$$\alpha_r = \text{atan} \left(\frac{\dot{y} - \dot{\varphi} \cdot l_r}{v_x} \right) \quad (17b)$$

The tire force is then calculated according to fitted magic formula. Note that in a single-track model two wheels are merged into one so the resultant lateral force is multiplied by two.

$$F_y = 2 \cdot D^{\text{Pac}} \cdot \sin[C^{\text{Pac}} \cdot \arctan(B^{\text{Pac}} \cdot \alpha - E^{\text{Pac}}(B^{\text{Pac}} \cdot \alpha - \arctan(B^{\text{Pac}} \cdot \alpha)))] \quad (18)$$

3.3 Model validation

This model was created in Simulink as a Matlab function. The v_x , δ and M_z data used to feed the simulation were collected during Baltic Open 2021 autocross race. The resulting lateral acceleration and yaw rate was compared with the collected data. The lateral acceleration

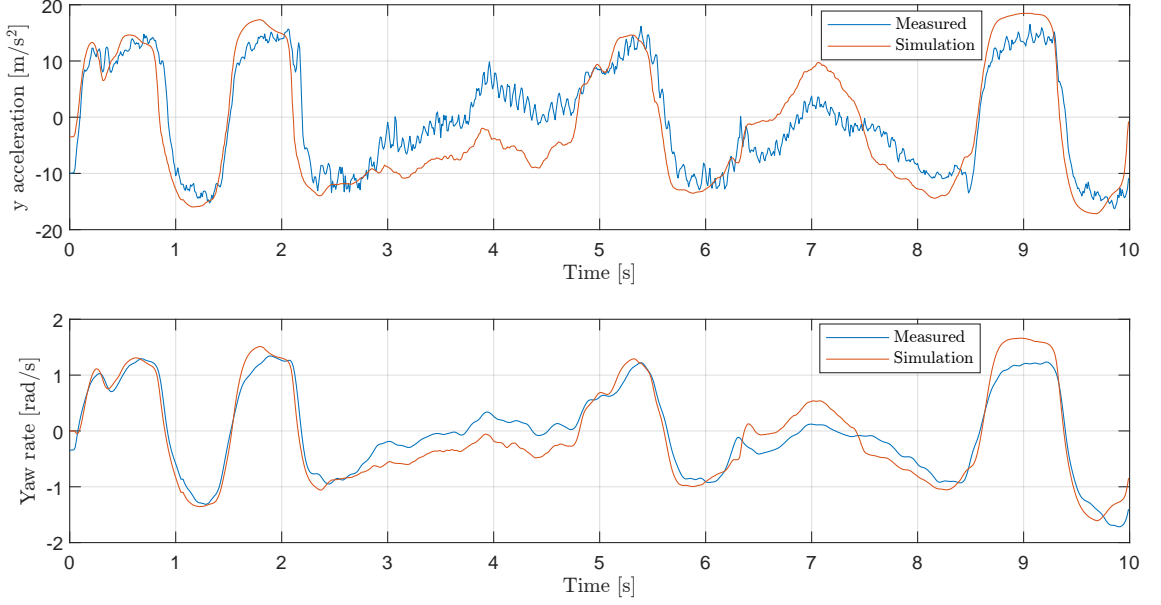


Fig. 9: Model validated on measured drive data

calculated by the model is in vehicle reference frame and cannot be directly compared with the data. An additional centripetal acceleration needs to be taken into account.

```
function [DDY, DDPSI] = model(DY, DPSI, delta, vx, Mz)
    SAf = atan((DY+DPSI * l_f)/vx)-delta;
    SAR = atan((DY-DPSI * l_r)/vx);
    Fyf = 2* D* sin(C* atan(B*SAf - E*(B*SAf-atan(B*SAf))));
    Fyr = 2* D* sin(C* atan(B*SAR - E*(B*SAR-atan(B*SAR))));
    DDY = 1/m*(Fyf*cos(delta) + Fyr) - vx*DPSI;
    DDPSI = 1/Iz*(Fyf*cos(delta)*l_f - Fyr*l_r + 1000*Mz);
end
```

3.4 Single track model in terms of β and $\dot{\varphi}$

The model developed in previous section has yaw rate and lateral acceleration as its state variables. Nevertheless, it is advantageous to choose body slip angle as the state variable instead of lateral acceleration. The lateral acceleration is defined in the body reference frame and so the centripetal acceleration must be subtracted, the measurement is noisy and arguably gives less insight into the behavior of the vehicle than the body slip angle (β).

Body slip angle is defined as

$$\beta = \text{atan} \frac{\dot{y}}{v_x}. \quad (19)$$

Editing the equation and calculating the time derivative, following expressions are obtained.

$$\dot{y} = \tan(\beta) \cdot v_x \quad (20a)$$

$$\ddot{y} = \frac{1}{1 + \tan^2 \beta} \cdot \dot{\beta} \quad (20b)$$

They can be then substituted for lateral acceleration in equations 17a, 17b, 16a and 16b.

After substitution, the equations for slip angle become

$$\alpha_f = \text{atan} \left(\tan \beta + \frac{\dot{\varphi} l_f}{v_x} \right) - \delta \quad (21a)$$

$$\alpha_r = \text{atan} \left(\tan \beta - \frac{\dot{\varphi} l_r}{v_x} \right) \quad (21b)$$

and the state space equations turn into

$$\dot{\mathbf{f}} = \begin{bmatrix} \dot{\beta} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} \frac{1}{m v_x \cdot (1 + \tan^2 \beta)} \cdot (F_{xf} \cdot \sin(\delta) + F_{yf} \cos(\delta) + F_{yr}) - \dot{\varphi} \\ \frac{1}{I_z} \cdot (F_{yf} \cdot \cos(\delta) \cdot l_f - F_{yr} \cdot l_r + 1000 \cdot M_z) \end{bmatrix}. \quad (22)$$

Model response to unit step of yaw torque under various conditions is presented on Figure 10 and 11. Notice how the response on Figure 10 changes with large steer angle. This is caused by large slip angle at which the lateral force has reached its saturation according to 7.

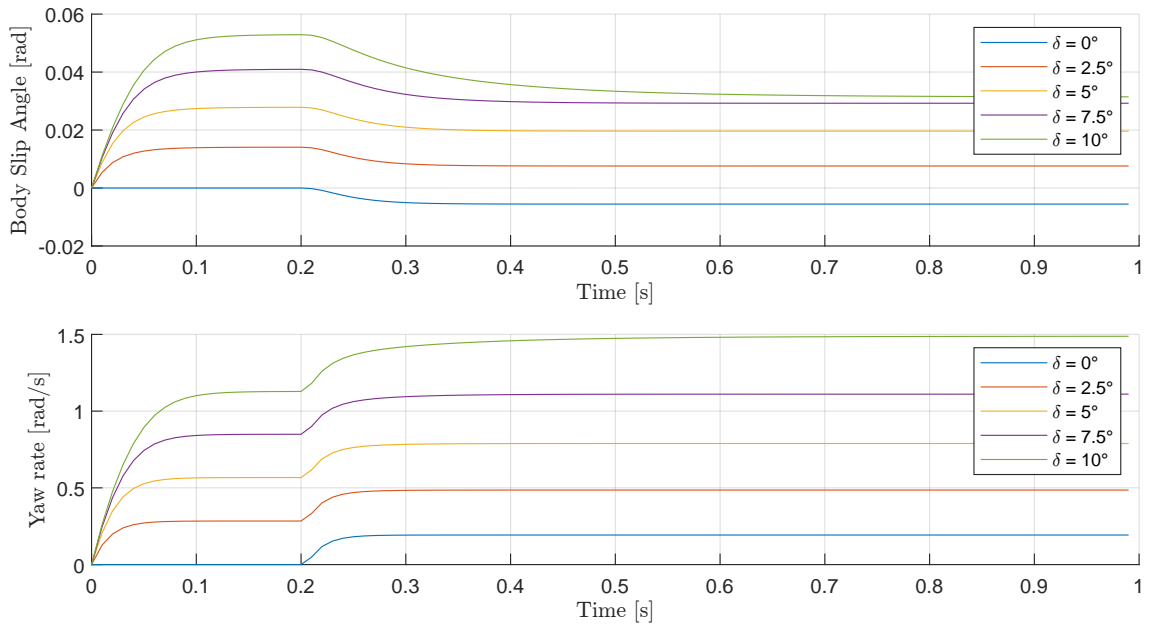


Fig. 10: Response of the model to unit yaw torque step at 0.2 s at various steer angle at $v_x = 10$ m/s.

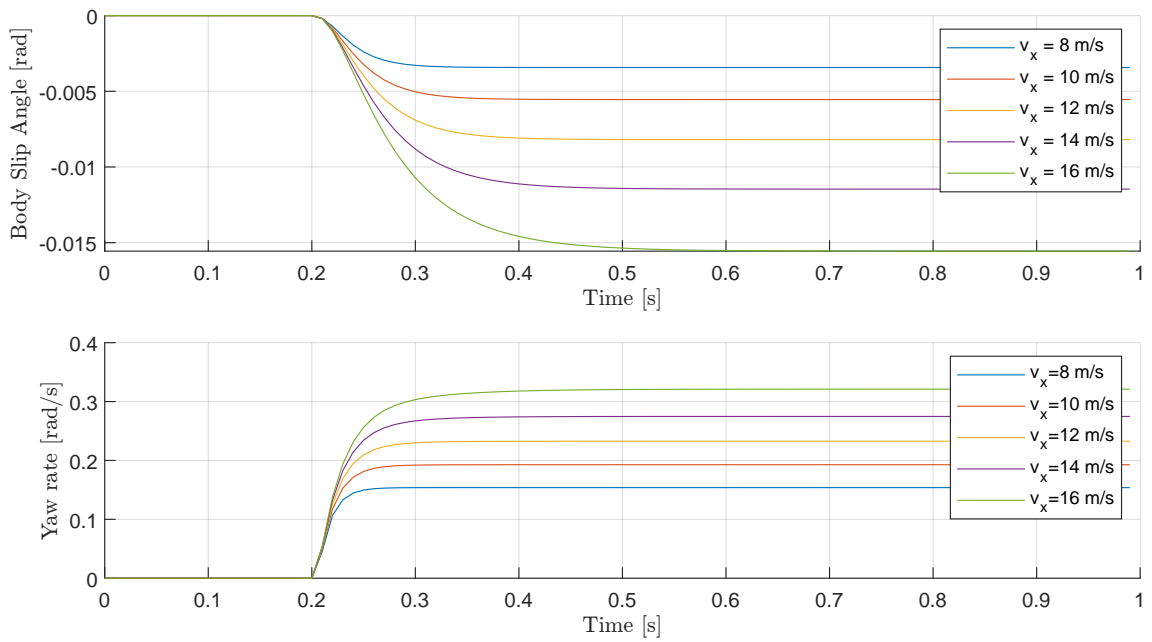


Fig. 11: Response of the model to unit yaw torque step at 0.2 s at various velocity at $\delta = 0^\circ$.

4 Controller

4.1 Stabilization and changing dynamics

One of the main design considerations when developing torque vectoring algorithm is how much to change the natural lateral dynamics. Ramp steer maneuver can be performed to plot static characteristic of the vehicle. During this maneuver, the driver tries to maintain constant velocity as he slowly turns the steering wheel at a constant rate. Steering angle and lateral acceleration is recorded and visualized in x-y plot. The shape of resulting curve determines if the vehicle is understeering or oversteering.

Production vehicles generally tend to understeer as it is safer for an inexperienced driver. However on the racetrack neutral steering characteristics is preferred. That means that the curve should be linear for as long as possible. Torque vectoring can change the shape of steering characteristics.

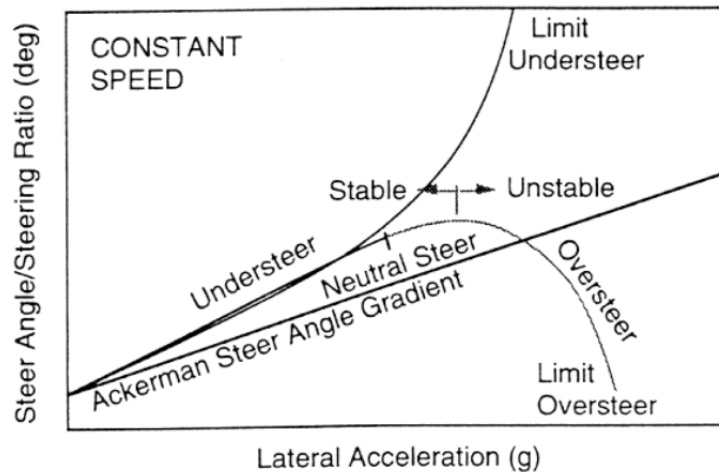


Fig. 12: Visualisation of possible steering characteristics.

[16]

The yaw controller should also be able to stabilize the vehicle from disturbances such as hitting bumps in the road and temporarily losing traction on one wheel and help the driver to follow desired trajectory.

These two objectives are achieved by adjusting the reference signal generator. The current system was designed to primarily stabilise the vehicle and keep its natural steering characteristics. That is why a nonlinear reference model was chosen to provide the requested yaw rate.

4.2 Reference signal

The whole concept of the torque vectoring system relies on a referential yaw rate that needs to be obtained from driver's inputs and vehicle states. Reference generator implemented in

current VDCU code operates in two modes. In low velocities (< 4 m/s) kinematic model is used as the reference. In higher velocities single-track model with three edgrees of freedom and nonlinear tires provides referential signals. This approach is rather complicated, since the dynamic behavior of the model can cause unwanted transient effects and tuning of this system requires fitting of the whole model.

One of the common approaches is to use steady state steering characteristics. The idea is to calculate the steady state that the vehicle would reach with current velocity and steering wheel angle. This approach was taken from [15] where author present derivation of the formula using geometry and force equilibrium. Needless to say, the same result can be obtained from equations of already developed dynamic model as the steady state yaw rate and body slip angle according to equation 23.

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\beta} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (23)$$

The formulae for steady state yaw rate and body slip angle are as follows.

$$\beta_{\text{ref}} = \frac{l_r - \frac{l_f \cdot m \cdot v_x^2}{2 \cdot C_\alpha \cdot (l_f + l_r)}}{K_u \cdot v_x^2 + l_f + l_r} \cdot \delta \quad (24)$$

$$\dot{\varphi}_{\text{ref}} = \frac{v_x}{l_f + l_r + K_u \cdot v_x^2} \cdot \delta \quad (25)$$

The K_u parameter is called understeering gradient. While it can be calculated from the parameters according to equation 26, it is suggested to adjust the behavior of the reference generator by changing the understeer gradient directly and comparing it with drive data. By tuning the parameters in a way that the desired yaw rate is different than the one measured on vehicle without torque vectoring, the dynamics of the car can be changed as mentioned in the previous section.

$$K_u = \frac{m \cdot (l_r - l_f)}{2 \cdot C_\alpha \cdot (l_r + l_f)} \quad (26)$$

Desired yaw rate calculation was validated on drive data, where recorded velocity and steer angle were used to calculate referential yaw rate which was compared with the vehicles actual measured yaw rate.

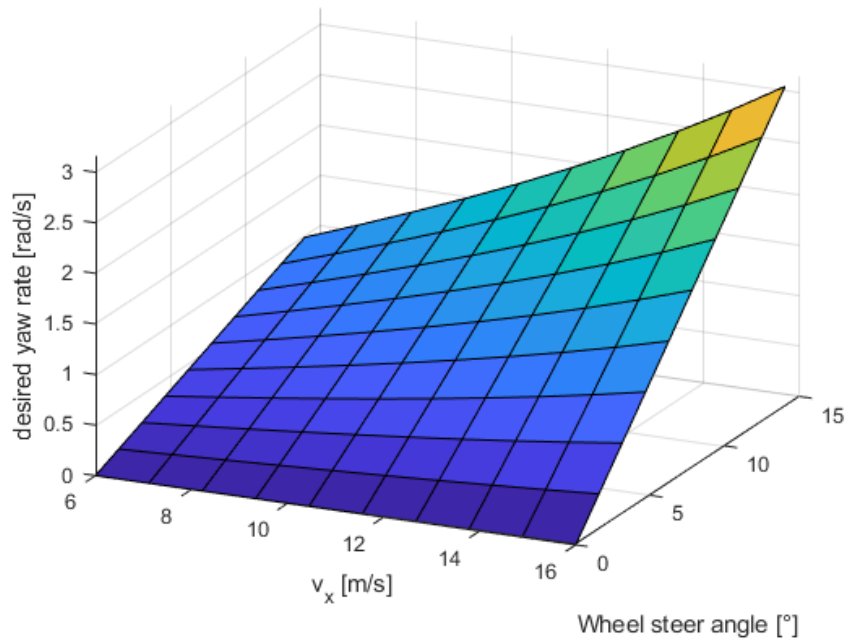


Fig. 13: Surface plot of desired yaw rate for oversteering vehicle

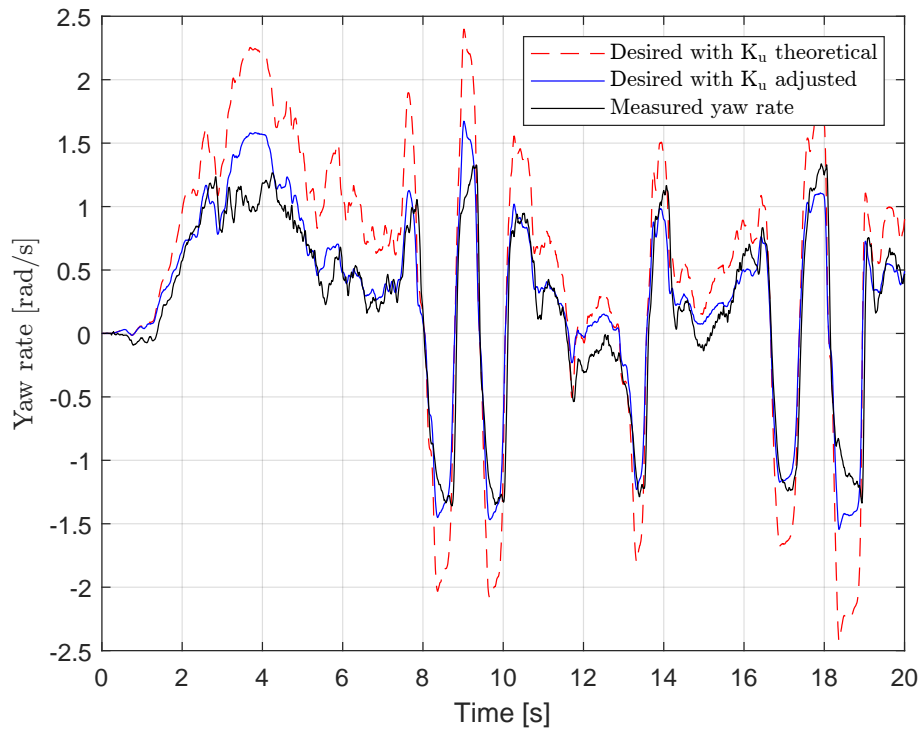


Fig. 14: Comparison between calculated referential yaw rate and actual yaw rate for different understeer gradients

4.3 Model linearization

For the controller synthesis, the next step is to obtain linearized model of the system in state space form. The indices c in the equation 27a differentiates the matrices of continuous model from the discrete ones.

$$\dot{\mathbf{x}} = A_c \mathbf{x} + B_c u + B_{dc} u_\delta \quad (27a)$$

$$\mathbf{y} = C_c \mathbf{x} + D_c u \quad (27b)$$

$$\mathbf{x} = \begin{bmatrix} \beta \\ \dot{\varphi} \end{bmatrix}$$

$$u = M_z$$

$$u_\delta = \delta$$

Since Pacejka's formula for lateral force has no physical background and is only used to fit the data, its linearization is omitted. First order polynomial is used to fit the data around $\alpha = 0$ instead. The tire force will be instead approximated by linear function of the slip angle. By linearizing magic formula around zero slip angle the same result would be achieved anyway. This is visualized on Figure 6. In order to avoid the complication of multiplying the coefficient by two every time, its value is assumed to be twice of what value that fitted the data on 6.

$$F_y = C_\alpha \cdot \alpha. \quad (28)$$

The parameter C_α is called cornering stiffness.

Matlab Symbolic Toolbox was used to carry out the derivations. Calculating the jacobian, the A matrix of state space representation is obtained.

$$A_c = \left. \frac{\partial f}{\partial x_i} \right|_{\mathbf{x}_0} = \begin{bmatrix} \frac{C_\alpha (\tan(\beta_0)^2 + 1)}{\left(\tan(\beta_0) - \frac{\varphi_0 l_f}{v_x}\right)^2 + 1} + \frac{C_\alpha \cos(\delta) (\tan(\beta_0)^2 + 1)}{\left(\tan(\beta_0) + \frac{\varphi_0 l_f}{v_x}\right)^2 + 1} & -\frac{C_\alpha l_f}{v_x \left(\left(\tan(\beta_0) - \frac{\varphi_0 l_f}{v_x}\right)^2 + 1\right)} - \frac{C_\alpha l_f \cos(\delta)}{v_x \left(\left(\tan(\beta_0) + \frac{\varphi_0 l_f}{v_x}\right)^2 + 1\right)} - 1 \\ \frac{C_\alpha l_f (\tan(\beta_0)^2 + 1)}{\left(\tan(\beta_0) - \frac{\varphi_0 l_f}{v_x}\right)^2 + 1} - \frac{C_\alpha l_f \cos(\delta) (\tan(\beta_0)^2 + 1)}{\left(\tan(\beta_0) + \frac{\varphi_0 l_f}{v_x}\right)^2 + 1} & \frac{C_\alpha l_f^2}{v_x \left(\left(\tan(\beta_0) - \frac{\varphi_0 l_f}{v_x}\right)^2 + 1\right)} + \frac{C_\alpha l_f^2 \cos(\delta)}{v_x \left(\left(\tan(\beta_0) + \frac{\varphi_0 l_f}{v_x}\right)^2 + 1\right)} \end{bmatrix}_{\mathbf{x}_0} \quad (29)$$

Substituting working point $\mathbf{x}_0 = [0 \ 0]^T$ into the Jacobi matrix, the matrix A of the state space representation is obtained. In [15] the author uses small angle approximation to get simplified model. He applies following assumptions and achieves the same result.

$$1 + \tan^2 \beta \approx 1 \quad (30)$$

$$\text{atan} \left(\tan \beta + \frac{l_r \dot{\varphi}}{v_x} \right) \approx \beta + \frac{l_r \dot{\varphi}}{v_x} \quad (31)$$

$$A_c = \begin{bmatrix} \frac{C_\alpha + C_\alpha \cos(\delta)}{m v_x} & -\frac{C_\alpha l_f - C_\alpha l_f \cos(\delta)}{I_z} - 1 \\ -\frac{C_\alpha l_f - C_\alpha l_f \cos(\delta)}{I_z} & \frac{C_\alpha l_f^2 + C_\alpha l_f^2 \cos(\delta)}{I_z v_x} \end{bmatrix}. \quad (32)$$

The system has two inputs - wheel steer angle and yaw torque. The steer angle is however not controlled and for the controller synthesis it is considered as a measured disturbance input. By calculating the derivatives of nonlinear state space with respect to u and u_δ , the B matrices are obtained.

$$B = \left. \frac{\partial f}{\partial u} \right|_{\mathbf{x}_0, u_0} = \begin{bmatrix} 0 \\ \frac{1000}{I_z} \end{bmatrix}$$

$$B_d = \left. \frac{\partial f}{\partial u_\delta} \right|_{\mathbf{x}_0, u_0} = \begin{bmatrix} \frac{-C_\alpha}{m \cdot v_x} \\ \frac{-C_\alpha l_f}{I_z} \end{bmatrix}$$

The states are directly measured, so the C_c matrix is identity matrix and there is no feedthrough dependence of the output on the input so D_c is zero.

$$C_c = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (35)$$

$$D_c = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (36)$$

4.4 Discretization

The controller runs in loop on hardware. The code is called periodically every 10 ms to match the frequency of the rest of the CAN bus network and so the model was discretized with sample time $T = 0.01$ s. Lower index notation will be used to describe element in prediction vector.

$$x_{k+1} = A x_k + B u_k \quad (37a)$$

$$y_k = C x_k + D u_k \quad (37b)$$

Equations 38a-38d, matrices A , B , C , D are obtained from their continuous counterparts using zero order hold. The derivation of the equations can be found in [17].

$$A = e^{A_c T} \quad (38a)$$

$$B = (A - I) A_c^{-1} B_c \quad (38b)$$

$$C = C_c \quad (38c)$$

$$D = D_c \quad (38d)$$

4.5 MPC

Model predictive controller was chosen to track the reference. Steering input is included as a measured disturbance and is assumed to be the same across the prediction horizon. Reference is also assumed to be same. This approach assumes, that the rate of change of steering angle and velocity is significantly slower than the lateral dynamics. To respect the changing of the dynamics with speed, the model updates to current conditions in each time step. In literature, this approach is called adaptive MPC or LPV-MPC (linear parameter varying). LPV systems can be considered subset of LTV-MPC (linear time varying). The LTV model is assumed to arbitrary change with respect to time, however for LPV there exists a set of time varying parameters, supposedly measurable, that influence the system behavior. [18]

4.5.1 Cost function

The idea of MPC is to find optimal series of inputs $\mathbf{u}_p = \begin{bmatrix} M_{z\ 1} & M_{z\ 2} & \dots & M_{z\ N} \end{bmatrix}^T$ to the system, that minimizes defined cost function over some prediction horizon N .

$$\text{minimize } J(\mathbf{u}_p) = \frac{1}{2}(\mathbf{y}_p - \mathbf{r})^T W_1 (\mathbf{y}_p - \mathbf{r}) + \Delta \mathbf{u}_p^T W_2 \Delta \mathbf{u}_p + \mathbf{u}_p^T W_3 \mathbf{u}_p \quad (39a)$$

The cost function was setup universally for case where action value as well as change of action value is penalized. The output of the system must be predicted based on current state and vector of inputs so that it can be substituted to cost function. W_1 is block diagonal matrix with weights for both states $[W_\beta \ W_\dot{\varphi}]^T$ on the diagonal. W_2 and W_3 are diagonal weight matrices penalizing change and value of action variable. $\mathbf{r} = \left[\beta_{\text{ref } 1} \ \dot{\varphi}_{\text{ref } 1} \ \beta_{\text{ref } 2} \ \dot{\varphi}_{\text{ref } 2} \ \dots \ \beta_{\text{ref } N} \ \dot{\varphi}_{\text{ref } N} \right]^T$ is a vector of referential values for both outputs and \mathbf{y}_p is predicted system trajectory.

4.5.2 Prediction

Trajectory of the system over set horizon is predicted based on current state, input and disturbance according to equation 40a. The wheel steering angle is uncontrolled input, so it is considered to be a disturbance. Since its progression over prediction horizon is not know, it will be considered to remain constant $\mathbf{u}_\delta = \left[\delta_0 \ \delta_0 \ \dots \ \delta_0 \right]^{N \times 1}$.

$$\mathbf{y}_p = P \mathbf{x}_0 + S \mathbf{u}_p + R \mathbf{u}_\delta \quad (40a)$$

Since the D matrix is zero, combining the equations 37a and 37b will result in following expression for next value of output

$$y_k = C(A x_k + B u_k + B u_{\delta k}) \quad (41a)$$

For next time step, previous value of state vector can be substituted according to equation 37a. Repeating this step, we get following pattern.

$$\begin{aligned} y_k &= C(A x_k + B u_k + B u_{\delta k}) \\ y_{k+1} &= C(A(A x_k + B u_k + B u_{\delta k}) + B u_{k+1} + B u_{\delta k+1}) \\ y_{k+2} &= C(A(A(A x_k + B u_k + B u_{\delta k}) + B u_{k+1} + B u_{\delta k+1}) + B u_{k+2} + B u_{\delta k+2}) \\ &\vdots \end{aligned} \quad (42)$$

Rewriting this in matrix form, P , S and R matrices from equation 40a are found.

$$\begin{aligned}
\mathbf{y}_p = & \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^n \end{bmatrix} \mathbf{x}_0 + \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ CA^{n-1}B & CA^{n-2}B & CA^{n-3}B & \dots & CB \end{bmatrix} \mathbf{u}_p \\
& + \begin{bmatrix} CB_\delta & 0 & 0 & \dots & 0 \\ CAB_\delta & CB_\delta & 0 & \dots & 0 \\ CA^2B_\delta & CAB_\delta & CB_\delta & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ CA^{n-1}B_\delta & CA^{n-2}B_\delta & CA^{n-3}B_\delta & \dots & CB_\delta \end{bmatrix} \mathbf{u}_\delta
\end{aligned} \tag{43}$$

4.5.3 Action change

The $\Delta \mathbf{u}$ term in cost function must also be expressed in matrix form. To do that, let's call the last applied input u_l .

$$\Delta \mathbf{u} = D \mathbf{u}_p + D_l u_l \tag{44}$$

$$\Delta \mathbf{u} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & & \\ \vdots & & \ddots & \ddots & \\ 0 & & & -1 & 1 \end{bmatrix} \mathbf{u}_p + \begin{bmatrix} -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u_l \tag{45a}$$

4.5.4 Quadratic programming

Quadratic programming is well understood optimization problem. Because the cost function is convex and has single minimum, it can be solved fast enough to enable the solver to run in real time. By combining equations 39a, 40a and 45a into equation 46 quadratic problem can be formulated 47a.

$$\begin{aligned}
J(\mathbf{u}_p) &= (\mathbf{y}_p - \mathbf{r})^T W_1 (\mathbf{y}_p - \mathbf{r}) + (D \mathbf{u}_p - D_l u_l) W_2 (D \mathbf{u}_p - D_l u_l) + \mathbf{u}_p^T W_3 \mathbf{u}_p \\
&= ((P \mathbf{x}_0 + S \mathbf{u}_p) - \mathbf{r})^T W_1 ((P \mathbf{x}_0 + S \mathbf{u}_p) - \mathbf{r}) \\
&\quad + (D \mathbf{u}_p - D_l u_l) W_2 (D \mathbf{u}_p - D_l u_l) + \mathbf{u}_p^T W_3 \mathbf{u}_p
\end{aligned} \tag{46}$$

$$\text{Minimize } J(\mathbf{u}_p) = \frac{1}{2} \mathbf{u}_p^T H \mathbf{u}_p + f^T \mathbf{u}_p \quad (47a)$$

$$H = S^T W_1 S + D^T W_2 D + W_3$$

$$f = -S^T W_1 \mathbf{r} + S^T W_1 P \mathbf{x}_0 + D^T W_2 D_l u_l$$

Subject to : $lb < u < ub$

4.5.5 Implementation

Proposed controller was developed as a Matlab function and simulated in loop with the developed model in Simulink.

The weights does not change during the run of the controller, so they can be calculated beforehand.

```

Qi = [W_beta; W_dpsi];
I = eye(N);
W1 = diag(kron(ones(N,1), Qi));
W2 = W_deltaMz * I;
W3 = W_Mz * I * 0.1;
Dif = I - diag(ones(N-1,1), -1);
Difl = [-1; zeros(N-1,1)];

```

The prediction matrices however change with current velocity so they need to be recalculated in each step. For implementation on HW this code could be optimized. The discrete dynamic matrices could be for example expressed directly with v_x parameter. Moreover, C is unity matrix so it could be ignored.

```

Ac = [(Ca + Ca*cos(delta))/(m*vx), ...
      - ((Ca*l_r)/vx - (Ca*l_f*cos(delta))/vx)/(m*vx) - 1; ...
      -(Ca*l_r - Ca*l_f*cos(delta))/Iz, ...
      ((Ca*cos(delta)*l_f^2)/vx + (Ca*l_r^2)/vx)/Iz];
Bc = [0; 1000/Iz];
Bdc = [-Ca/(m*vx); -(Ca*l_f)/Iz];
C = [1 0; 0 1];

```

```

A = expm(Ac*Ts);
B = (Ac) \ ((A-eye(size(A))) * Bc);
Bd = (Ac) \ ((A-eye(size(A))) * Bdc);

```

```
N = 15;
```

```

memS = cell([N,1]);
memR = cell([N,1]);
for i=0:N-1
    memS{i+1} = C*(A^i)*B;
    memR{i+1} = C*(A^i)*Bd;
end

for i=1:N
    P((i-1)*2+1:2*i, 1:2) = C*A^i;
    for j = 1:i
        index = i-j+1;
        S((i-1)*2+1:i*2, (j-1)+1:j) = memS{index};
        R((i-1)*2+1:i*2, (j-1)+1:j) = memR{index};
    end
end
end

```

Lastly, the H and f matrices are formulated and `quadprog` function is called. Fortunately, Matlab Coder supports code generation for this function [19].

```

ub = Mzmax*ones([N, 1]);
lb = -Mzmax*ones([N, 1]);
rp = kron(ones(N, 1), [betaref, dpsiref]');
udelta = delta*ones([N 1]);
H = S'*W1*S + Dif'*W2*Dif + W3;
Hsym = (H+H')/2;
f = -S'*W1*rp + S'*W1*P*x0 + Dif'*W2*Difl*ul ...
    + S'*W1*R*udelta;
optiopt = optimoptions('quadprog',...
    'Algorithm','active-set', "Display", "none");
[u_sol, cost] = quadprog(Hsym, f,...
    [], [], [], [], lb, ub, zeros([N, 1]), optiopt);
u = u_sol(1);

```

Prediction horizon was set to 15 steps, because at this value it covered the full dynamic response. β was set to zero as the behavior of body slip angle was found to be very sensitive to parameters and inconvenient to use for control. Maximal yaw torque was set to $M_{z\max} = 500$ Nm in both directions and further tuned by applying penalization to action. Weight of ΔM_z was set so that the controller did not oscillate rapidly however war kept generally as low as possible. Weights were adjusted according to need, but the default set of weights was $W_\beta = 0$, $W_{\dot{\varphi}} = 0.5$, $W_{\Delta M_z} = 0.1$ and $W_{M_z} = 0.5$.

5 Results

5.1 Ramp change

Closed loop system was tested on a simple step input of steering angle. To make the simulation more realistic, instead of step, the steering angle rises linearly from 0 to 5° during 0.2 s. The simulation was done three times at 6, 10 and 14 m/s. The results can be seen on Figures 15, 16 and 17. In the highest speed, it is apparent that the yaw rate does not reach the reference perfectly. This is caused by imperfect model ignoring the high slip angle tire saturation.

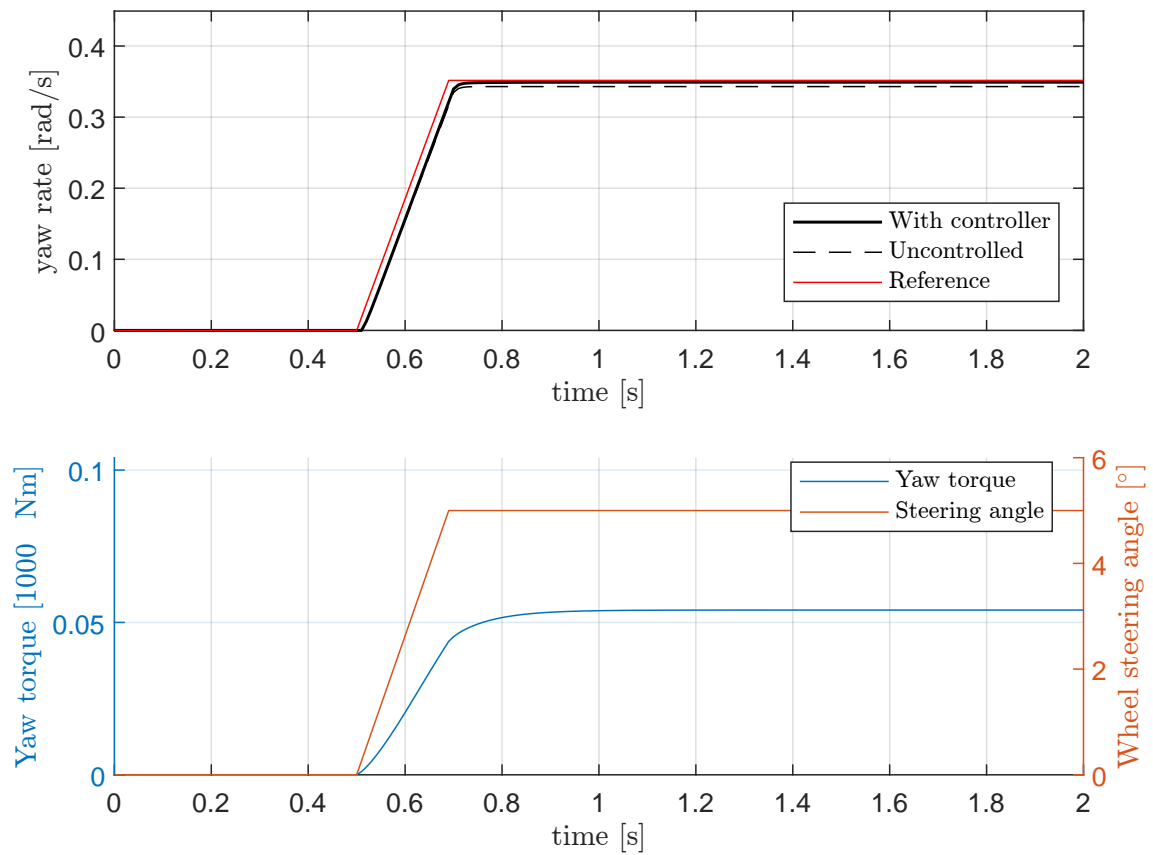


Fig. 15: Closed loop response to steering step at $v_x = 6$ m/s

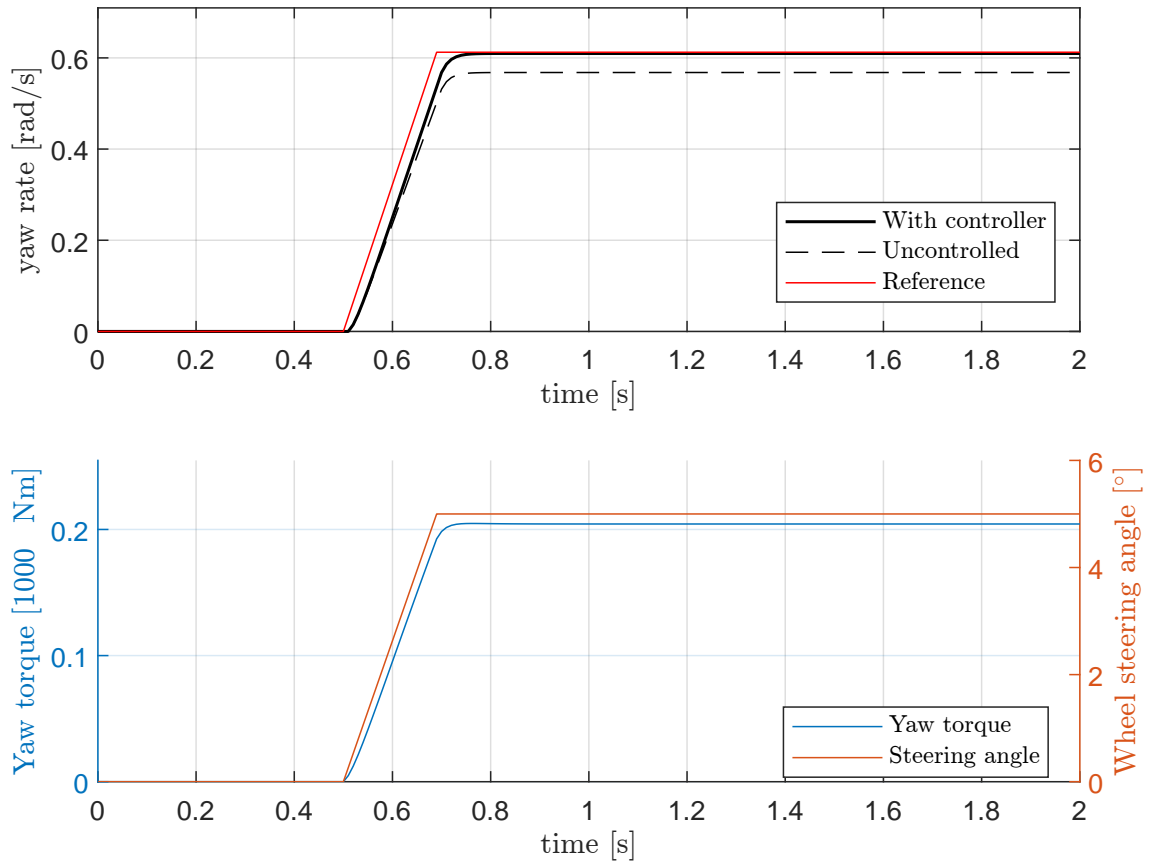


Fig. 16: Closed loop response to steering step at $v_x = 10$ m/s

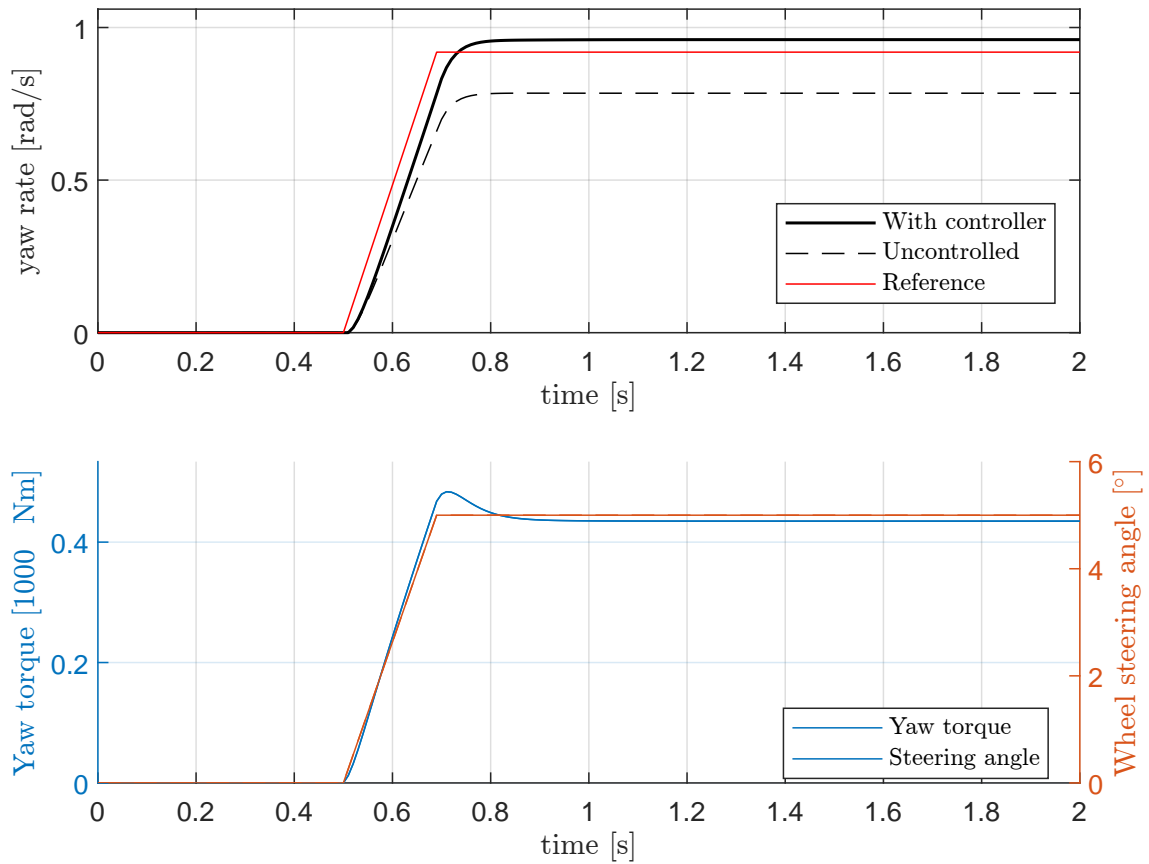


Fig. 17: Closed loop response to steering step at $v_x = 14$ m/s

5.1.1 LTI-MPC

For validation of the approach, the developed MPC was tested without an online model update. The velocity was set to $v_x = 10$ m/s and the same step steer test was conducted so the results could be compared.

As can be seen on the simulation results, the system does not work at all. At 6 m/s the yaw torque has negative value making the response worse than uncontrolled and at 14 m/s the vehicle loses stability.

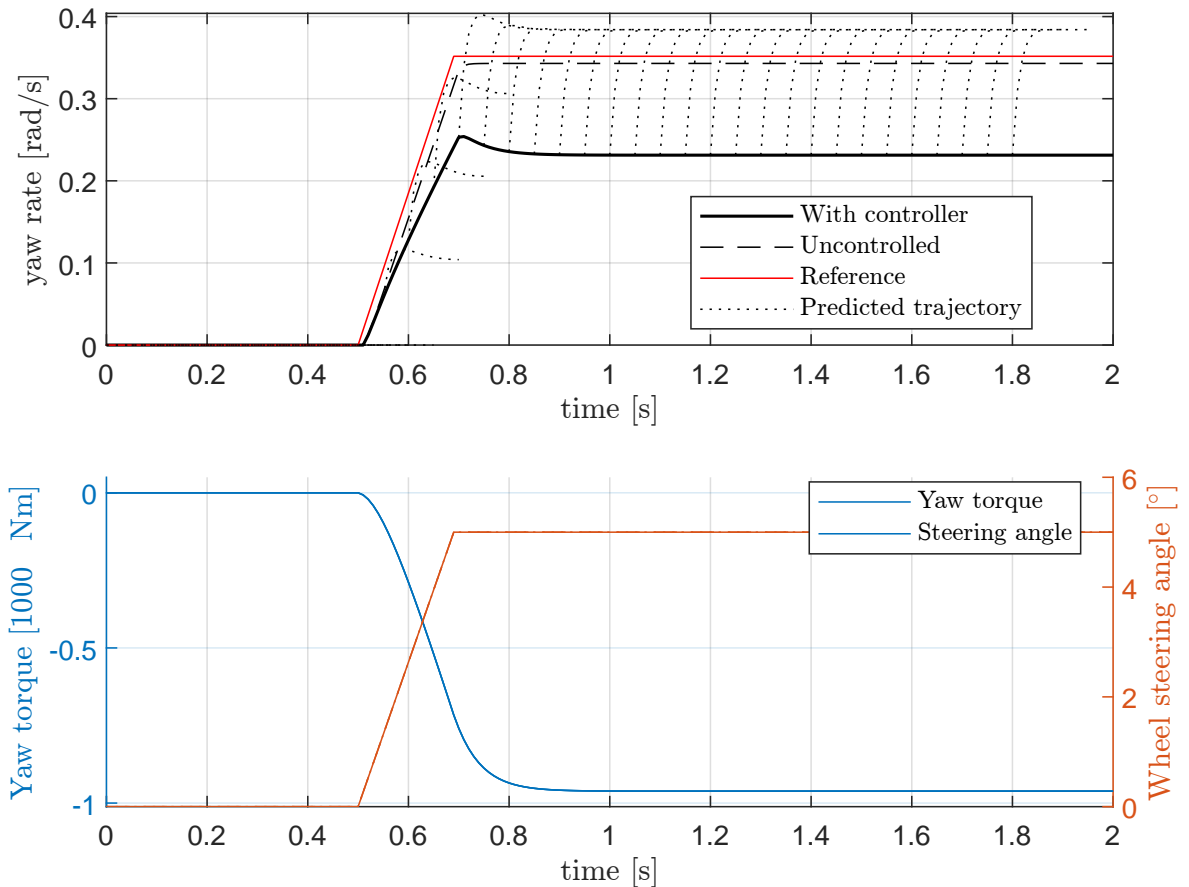


Fig. 18: Closed loop response of LTI-MPC to steering step at $v_x = 6$ m/s

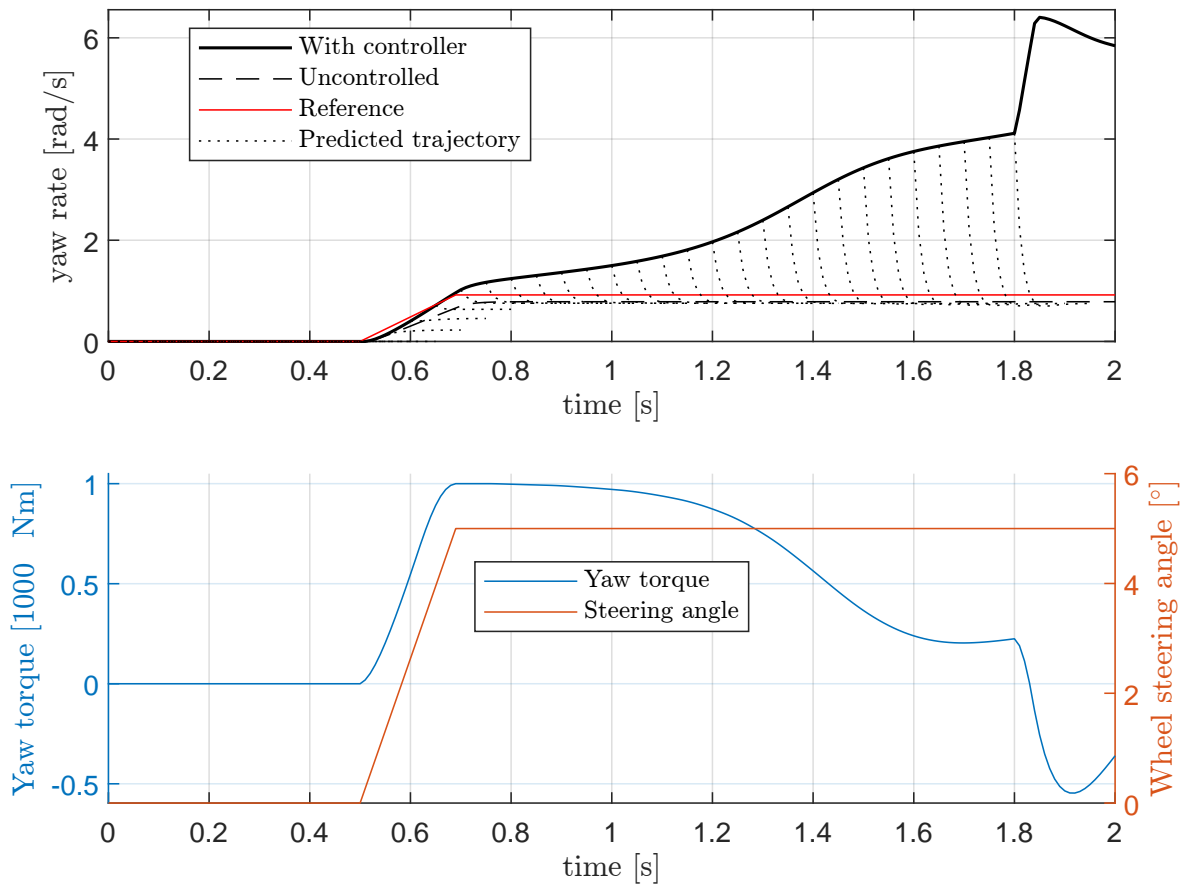


Fig. 19: Closed loop response of LTI-MPC to steering step at $v_x = 14$ m/s

5.2 Disturbance attenuation

Disturbance was simulated as an unmeasured step in a steering input of 5.73° (0.1 rad) for duration of 0.3 s applied at 0.5 s simulation time. The simulation was conducted for combinations of velocity 10 and 14 m/s and steering angle 0° and 5° . The ability to attenuate such disturbance depends significantly on the weight of rate of change of action variable.

The results show that while the controller does apply some counter torque to balance the error, the predictor expects the system to stabilize itself (as can be seen on Figure 20) and therefore it does not fully compensate for the error.

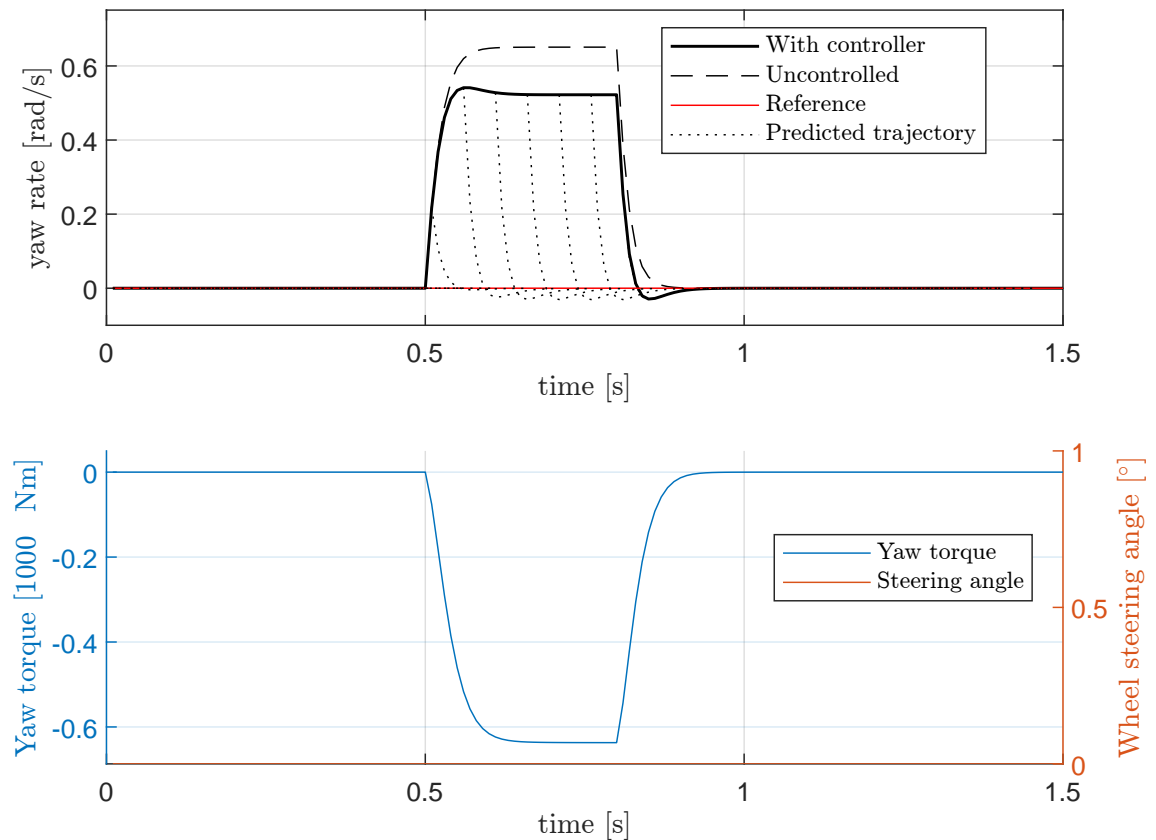


Fig. 20: Closed loop response to disturbance at $v_x = 10 \text{ m/s}$ and $\delta = 0^\circ$ with visualized prediction trajectories for every 5th simulation step

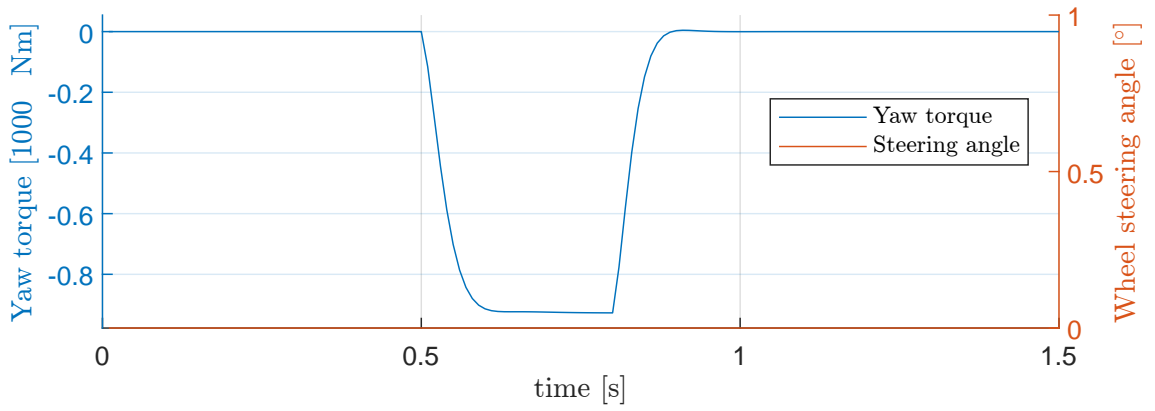
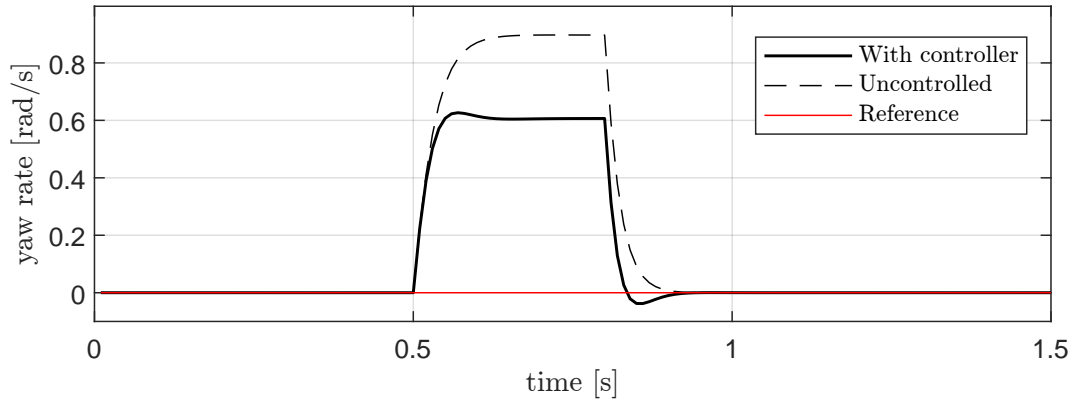


Fig. 21: Closed loop response to disturbance at $v_x = 14$ m/s and $\delta = 0^\circ$

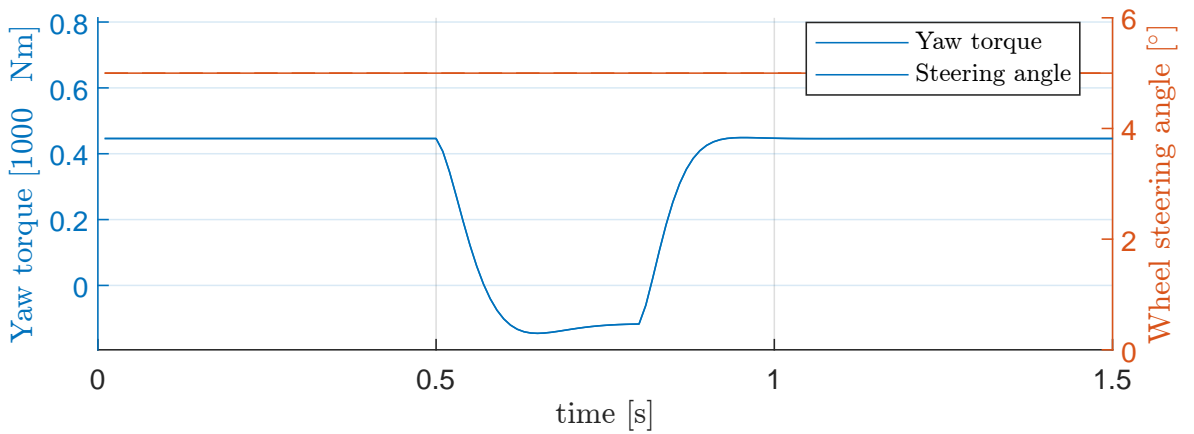
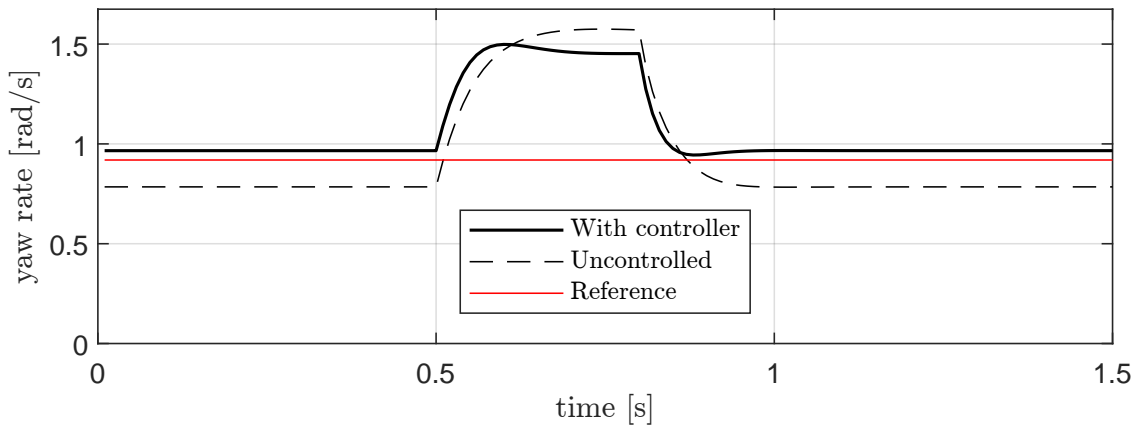


Fig. 22: Closed loop response to disturbance at $v_x = 14$ m/s and $\delta = 5^\circ$

5.3 Virtual test drive

Two experiments were conducted using IPG CarMaker virtual driving environment. In the first one, only the inputs were measured and then applied to closed loop simulation with the developed single track model. This way the response to real inputs was tested in order to validate the controller performance under changing velocity.

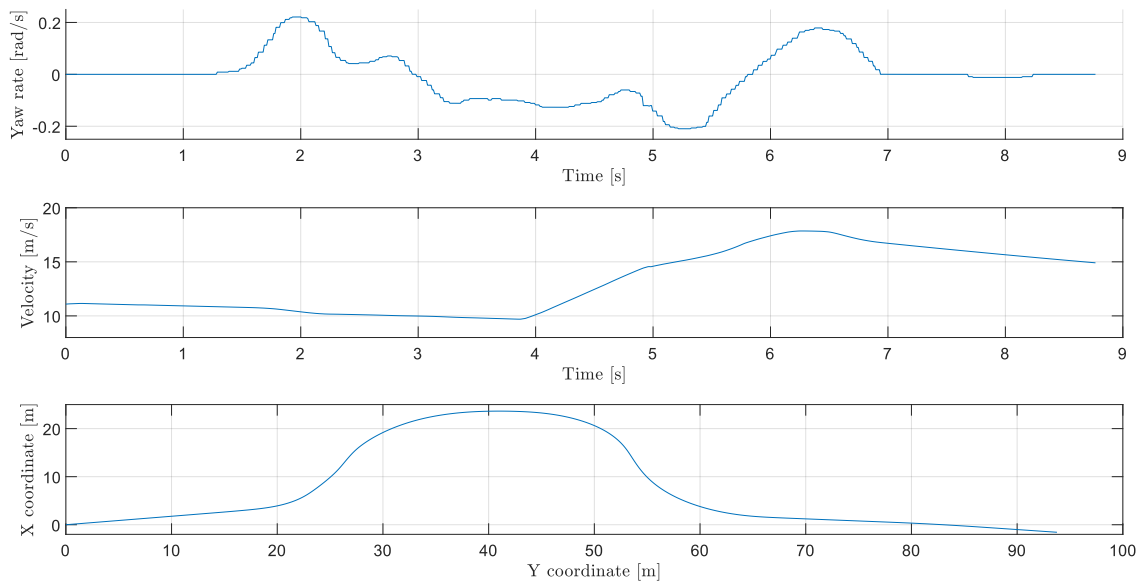


Fig. 23: Measured inputs for simulation

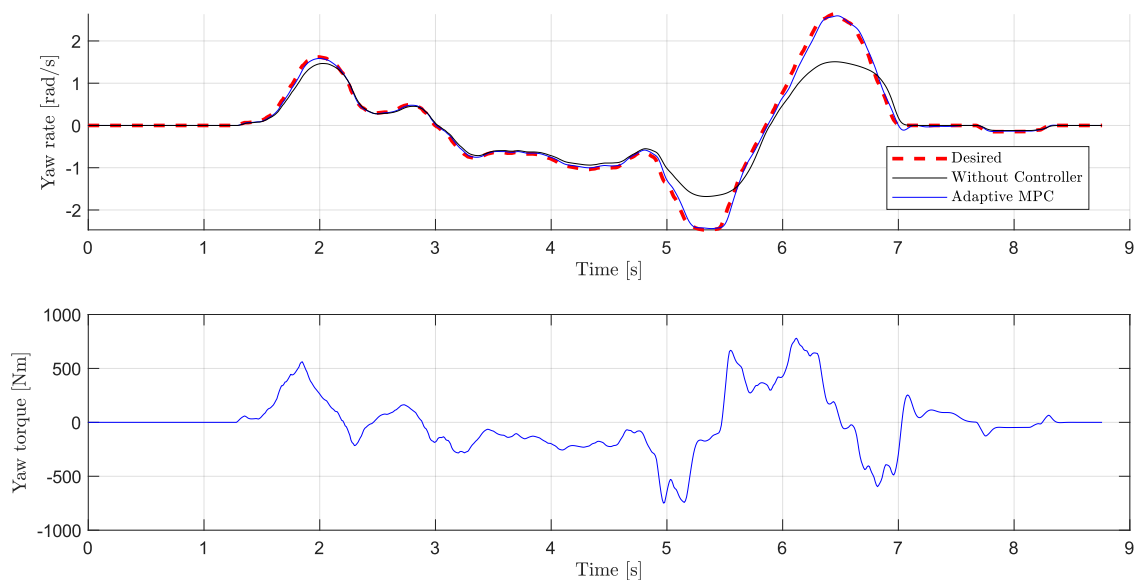


Fig. 24: Simulation of control loop with measured data

The second test was done with the controller running in real time with CarMaker. Using CarMaker for Simulink suite, it is possible to access the simulation and alter its variables in during the execution. CarMaker supports external input devices such as gaming steering wheel with pedals, which can be used to control the virtual vehicle.

Even though the virtual multibody vehicle model was developed according to the actual vehicle, the controller's prediction model had to be re-fitted to its dynamics that had turned out to be slightly different than model fitted in Section 3.2.

Driver-in-loop test drive was done with and without the controller on a model of a real Formula Student track. The very first realization was that the controller is way too aggressive and thus its weights had to be adjusted again. The ride with and without torque vectoring could not be directly compared, because inexperienced driver will not follow the same path and his actions will vary significantly between each lap. Subjectively, the main difference was that the car felt more sensitive and faster in corners. However, the loss of traction in high speed corners was more common with the controller turned on. It is difficult to make a conclusion from this, because the driver does not have a direct feedback and it gets hard to know the limits of the car without it.

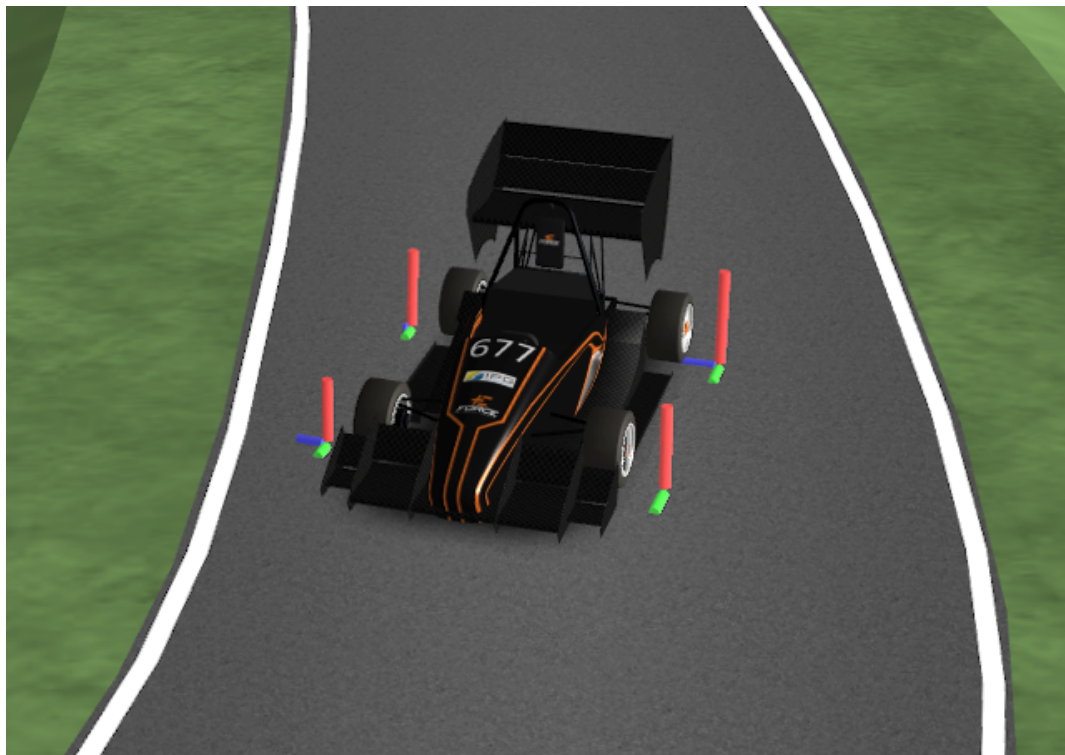


Fig. 25: CarMaker virtual driving environment

In order to obtain at least some objective results, the drive data with and without the controller had to be compared. It was decided to perform a simple double lane change maneuver on the same track which was used to record data on Figure 23. The test was performed 40 times in total, switching the controller on and off after each 10 repetitions starting with the controller

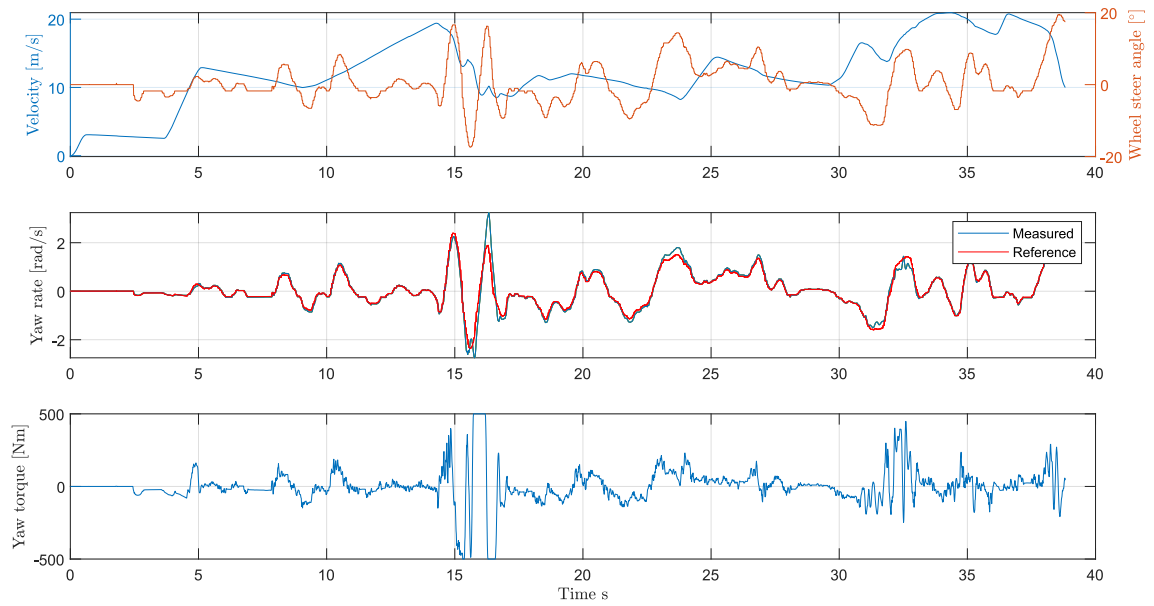


Fig. 26: Recorded data from virtual drive

turned on. Subjectively, with the controller turned off, the car was understeering and was much less responsive. The torque vectoring took a few rides to get used to however.

Comparison of the data is perhaps not as conclusive as one would hope for without performing a proper data analysis such as cropping the signals so that all of them start at the same time. Noticeable difference was however apparent when the steering angle and yaw rate signals were compared. With torque vectoring, less steering action was required and the yaw rate was overall faster.

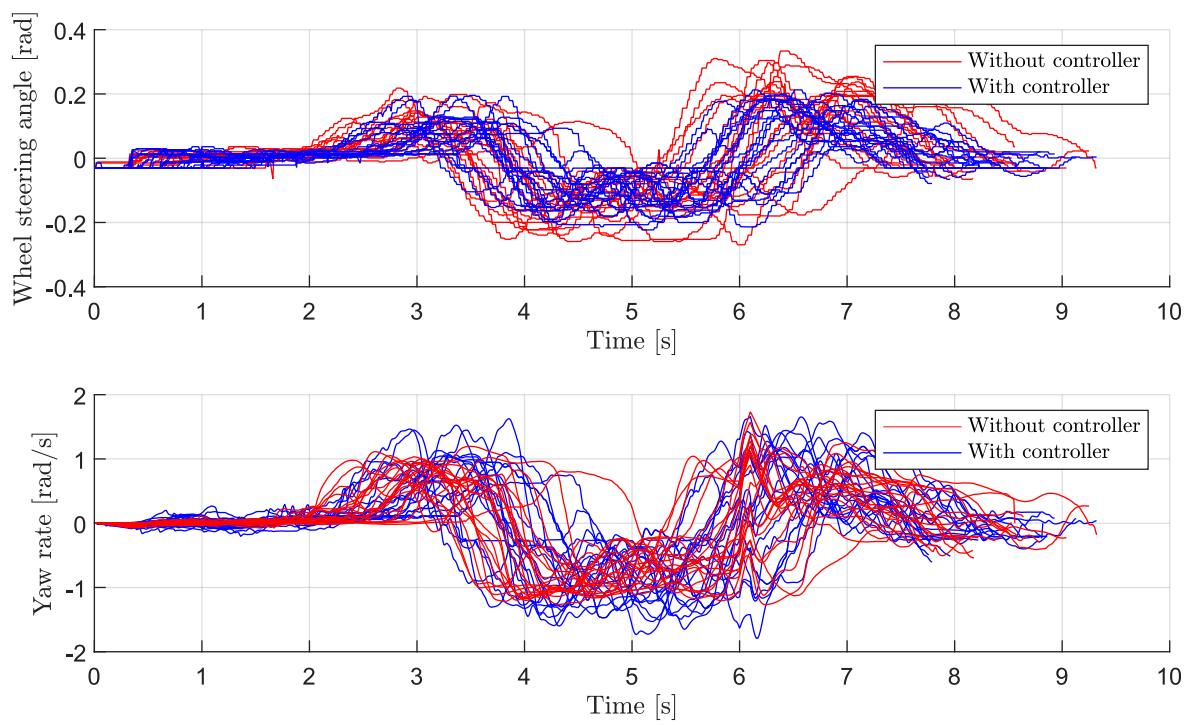


Fig. 27: Repeated double lane change test results

6 Yaw torque distribution

Last layer of the system is the actual application of the yaw torque on the chassis. In the simplest case, this torque can be evenly distributed between left and right wheels according to formula 48.

$$-\Delta F_x^{\text{FL}} = -\Delta F_x^{\text{RL}} = \Delta F_x^{\text{FR}} = \Delta F_x^{\text{RR}} = \frac{M_z}{2 \cdot t} \quad (48)$$

Currently implemented algorithm varies the longitudinal distribution of torque according to multiple adjustable parameters and limits the maximal torque, that can be applied to a single motor. If the saturation is reached, the algorithm starts to reduce the torque on the opposite side and thus reduces the total forward torque. However, the performance can be optimized by taking into account the wheel load. This way loss of traction can be prevented and the grip is effectively utilized.

Since it is not always possible to adhere to yaw torque request and forward force while staying within the tire grip limit, the idea is to formulate this as an optimization problem that weights these three requirements and finds according optimal solution. Because the requirements are formulated in terms of the forces acting on the chassis, instead of working with the motor torques, the vector of the longitudinal tire forces is defined as $\theta = \left[F_x^{\text{FL}} \quad F_x^{\text{FR}} \quad F_x^{\text{RL}} \quad F_x^{\text{RR}} \right]^T$. Once the optimal values of vector θ are found, they will be easily converted to the motor torque requests according to the equation 49.

$$\tau = \frac{F_x \cdot r_w}{i} \quad (49)$$

Maintaining the total forward force request $F_{x \text{ req}}$ means simply that the sum of all four torques equals the requested value. In matrix form it can be written as

$$\begin{aligned} S &= \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \\ S \theta &= F_{\text{req}}. \end{aligned} \quad (50a)$$

The yaw torque request can be written as

$$\begin{aligned} Y &= \begin{bmatrix} -\frac{t}{2} \cos \delta + l_f \sin \delta & \frac{t}{2} \cos \delta + l_r \sin \delta & \frac{t}{2} & \frac{t}{2} \end{bmatrix} \\ Y \theta &= M_z \end{aligned} \quad (51a)$$

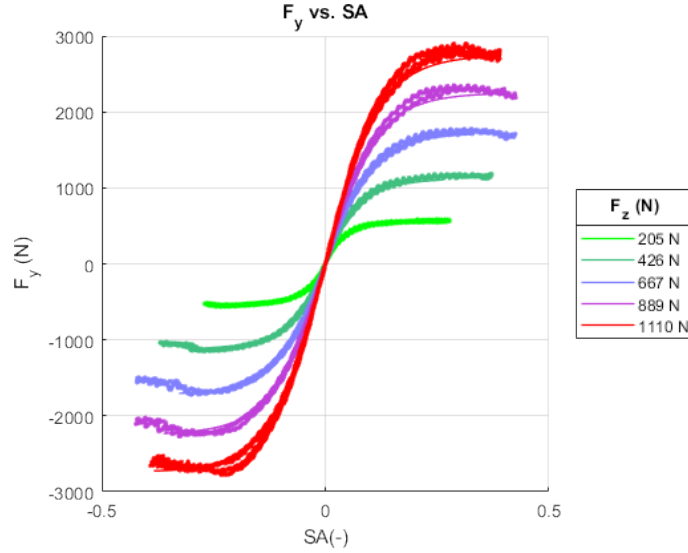


Fig. 28: Pure slip lateral force test data from Calspan fitted with Pacejka’s magic formula

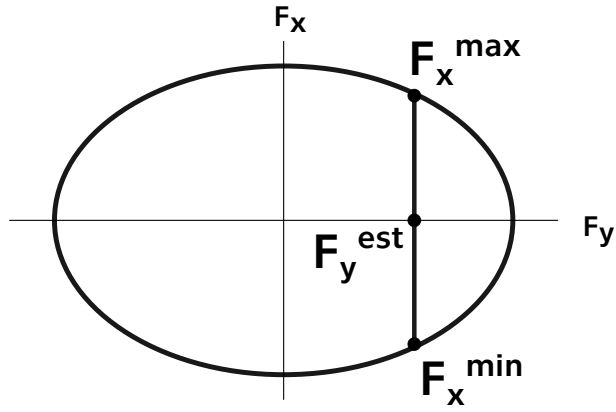


Fig. 29: Friction ellipse with maximal longitudinal force visualized

Lastly, the maximal longitudinal force must be calculated in order to know the limit of longitudinal force of each wheel. This could be done from the wheel slip angle and further tire lateral force estimation. In addition, the size and shape of friction ellipse for matching vertical load of each tire would have to be assessed. This could be quite easily approximated from available TTC data (Figure 28) and adjusted by scaling factor. Equations for vertical load estimation are presented in Section 2. If friction ellipse and lateral force is known, the maximal and minimal longitudinal force can be calculated from equation 9. This is visualized on Figure 29.

$$F_{x \text{ lim}} = \sqrt{1 - \left(\frac{F_y}{F_{y \text{ max}}}\right)^2} \cdot F_{x \text{ max}} \quad (52)$$

Question arises as how to formulate this request. The force should be smaller than its limit, so it can be formulated as a constraint for the optimization solver. In addition to that however the forces should ideally be distributed evenly between the wheels. If only a limit was set,

for example the solution $\theta = [500 \ 0 \ 0 \ 0]^T$ could be optimal. Instead of hard limits, the forces are minimized proportionally to their maximal value. This request could however go against the first one so attention must be paid to correct setting of the weight factors.

$$L = \begin{bmatrix} \frac{i}{F_{x\lim}^{FL} \cdot r_w} & 0 & 0 & 0 \\ 0 & \frac{i}{F_{x\lim}^{FR} \cdot r_w} & 0 & 0 \\ 0 & 0 & \frac{i}{F_{x\lim}^{RL} \cdot r_w} & 0 \\ 0 & 0 & 0 & \frac{i}{F_{x\lim}^{RR} \cdot r_w} \end{bmatrix} \quad (53)$$

Combining all three requirements, cost function can be formulated.

$$\begin{aligned} J(\theta) &= (S\theta - F_{x\ req})^T W_1 (S\theta - F_{x\ req}) \\ &+ (Y\theta - M_z)^T W_2 (Y\theta - M_z) \\ &+ (L\theta)^T W_3 (L\theta) \end{aligned} \quad (54)$$

Equation 54 was rewritten so that it can be solved by quadratic solver.

$$\begin{aligned} H &= \left[S^T W_1 S + Y^T W_2 Y + L^T W_3 L \right] \\ f &= -F_{x\ ref} W_1 - M_{z\ ref} W_2 Y \end{aligned} \quad (55a)$$

The torque distribution is then solved as a quadratic problem.

$$\begin{aligned} \min J(x) &= \frac{1}{2} x^T H x + f^T x \\ -F_{x\ lim} &< x < F_{x\ lim} \end{aligned} \quad (56a)$$

The proposed algorithm was realized in Matlab and tested in CarMaker's virtual driving environment. The manoeuvre consisted of acceleration from stand still to 50 km/h, lane change and acceleration to 60 km/h. Built-in virtual driver was used to control the steering and acceleration.

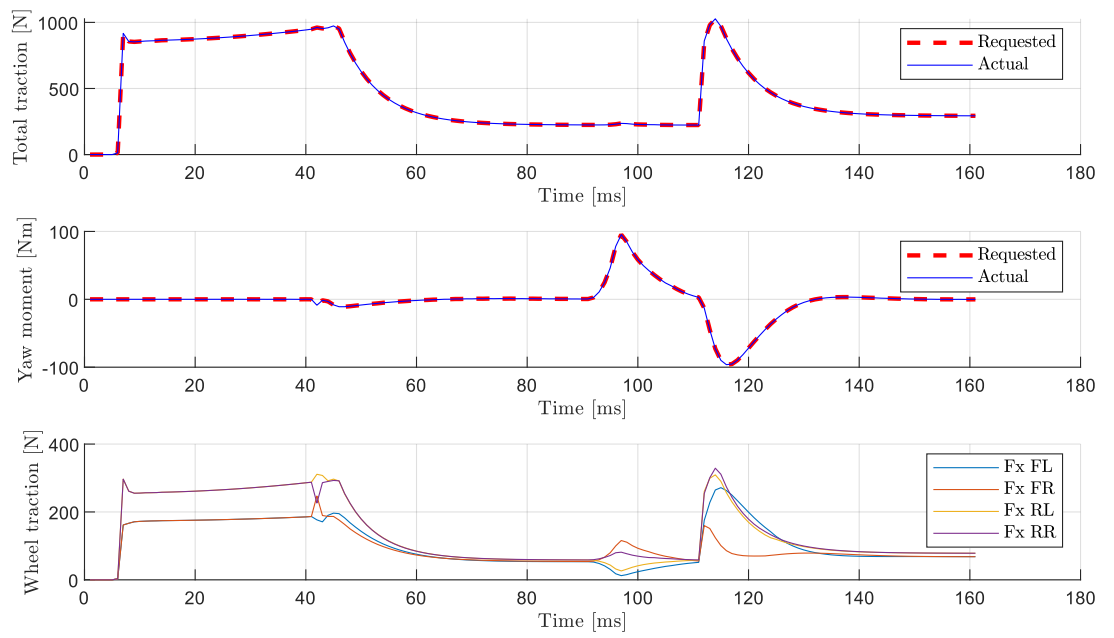


Fig. 30: Proposed traction distribution tested on lane change maneuver

7 Conclusion

The main objective of this thesis was to design a controller that would improve and stabilize lateral dynamics of a prototype vehicle.

In the beginning, a short introduction to the vehicle, its characteristics and available measurements are presented along with a description of a current state of the implemented torque vectoring system. Then vehicle dynamics equations and principles are presented to build mathematical background for model based design. Publicly available data as well as self-measured data were used to fit and validate the developed model.

Based on the model, a linear parameter varying model predictive controller (LPV-MPC) is proposed to improve the dynamics of the vehicle. The response to simple step input is presented to validate the controller and its variations. The performance of the system is then evaluated on drive data as well as in driver-in-loop simulation. Lastly, an algorithm for distribution of yaw torque between four wheels is presented.

The proposed controller adjusts itself to the dynamics that change with time. This is an advantage over the current solution which has a single PI controller. Alternatively, a gain scheduling could be implemented to current solution. A potential drawback could be the computation time on HW, however the implementation should be feasible, according to [20], where the authors investigated performance of a similar control structure on a similar hardware.

Although the algorithm works in a simulation, some potential problems have to be addressed to assure a complete safety under all conditions before its implementation to a real vehicle. For example, the velocity must always be greater than zero for the system to work correctly. The controller does not reflect the nonlinearities of the tires and thus might underperform while driving at the limits. Additional constraint on a maximal possible body slip angle could be added to the optimizer.

Alternatively, the full model including Pacejka's formula for tire force could be linearized in each controller cycle or even successively in each prediction step. This approach is successfully applied to a similar problem in [21]. Even though the weights of cost function were adjusted to achieve desired simulation results, in reality, they would need to be adjusted according to the driver's liking to be more or less aggressive.

The concept explored in this thesis is the combination of the yaw controller and its distribution into a single system which would directly return four wheel torques. This approach means that the yaw torque could only be applied if the tires would allow it. A similar system is proposed in [14] where the model also includes information about wheel angular velocity and thus also substitutes a traction control system. However, the idea was

rejected as such system would not be consistent with the current implementation and would therefore require its complete redesign.

Besides the original objective of this thesis, another significant contribution is the know-how that could be applied to develop a similar controller for following trajectory of an autonomous car. Team eForce also competes in the driverless category of Formula Student and a predictive lateral controller would improve the performance of the autonomous car significantly compared to the current reactive PI controller.

References

- [1] Mitsubishi Motors, “S-AWC,” [online] Available: <https://www.mitsubishi-motors.com/en/innovation/technology/SUVNewValue/sawc>, visited on 2022-05-14.
- [2] ZF Friedrichshafen AG, “Driving dynamics,” [online] Available: https://www.zf.com/master/media/en/corporate/m_zf_com/company/download_center/products/passenger_cars/dynamikvonmorgenzfachsantriebefrpkw.pdf, visited on 2022-05-14.
- [3] Rimac Automobili, “Rimac all wheel torque vectoring,” [online] Available: <https://www.rimac-automobili.com/media/press-releases/rimac-all-wheel-torque-vectoring/>, 2016, visited on 2022-05-14.
- [4] IEA, “Global ev outlook 2021,” [online] Available: <https://www.iea.org/reports/global-ev-outlook-2021>, Paris, 2021.
- [5] J. P. M. Antunes, “Torque vectoring for a formula student prototype,” http://www.dem.ist.utl.pt/poliveira/Ensino/paper_2017c.pdf, 2017.
- [6] MathWorks, “Torque vectoring: Controller design, tuning, and testing,” [online] Available: <https://www.mathworks.com/videos/series/student-teams-share-their-keys-to-success-101105.html>, Apr. 2015, visited on 2022-05-10.
- [7] C. Hahn, “"torque vectoring – kit team sharing vehicle controls success story",” [online] Available: <https://blogs.mathworks.com/student-lounge/2017/11/22/torque-vectoring-ka-race-ing/>, 11 2017, visited on 2022-05-10.
- [8] M. Laszlo, “Flight control solutions applied for improving vehicle dynamics,” Master’s thesis, Faculty of Electrical Engineering, CTU, January 2019.
- [9] Moad Kissai, “Adaptive robust vehicle motion control for future over-actuated vehicles,” https://www.researchgate.net/figure/Vehicle-Axis-System-ISO-8855-2011_fig1_332561752, 2019, [Online; accessed 27 Feb 2022].
- [10] W. F. Milliken and D. L. Milliken, *Race Car Vehicle Dynamics*. Great Britain: Society of Automotive Engineers Inc., 1996.
- [11] Milliken Research Associates, “Formula sae tire testing consortium,” [online] Available <https://www.millikenresearch.com/fsaettc.html>, 2022, visited on 2022-05-30.
- [12] H. B. Pacejka, *Tire and vehicle dynamics*. Warrendale, USA: SAE, 2002.
- [13] U. Kiencke and L. Nielsen, *Automotive control systems*. Springer, 2000.
- [14] G. Palmieri, O. Barbarisi, S. Scala, and L. Glielmo, *An Integrated LTV-MPC Lateral Vehicle Dynamics Control: Simulation Results*, 11 2010, vol. 402, pp. 231–255.

- [15] R. Rajamani, *Vehicle Dynamics and Control*. Austin, USA: Springer, 2006.
- [16] T. Gillespie, “Fundamentals of vehicle dynamics,” 1992.
- [17] V. Havlena and J. Štecha, *Moderní teorie řízení*. České vysoké učení technické, 1994.
- [18] Sename O. and Fergani S., “Lecture notes in linear parameter varying systems: from modelling to control,” [online] Available: https://hal.univ-grenoble-alpes.fr/hal-01661028/file/IVSS2017_LPV.pdf, July, 2017, visited on 2022-08-08.
- [19] Matlab, “Code generation for quadprog background,” <https://www.mathworks.com/help/optim/ug/code-generation-in-quadprog.html>, visited on 2022, visited on 2022-08-8.
- [20] G. Batista, G. Takács, and B. Rohal Ilkiv, “Application aspects of active-set quadratic programming in real-time embedded model predictive vibration control,” *IFAC-PapersOnLine*, vol. 50, pp. 11 625–11 631, 07 2017.
- [21] A. Katriniok and D. Abel, “Ltv-mpc approach for lateral vehicle guidance by front steering at the limits of vehicle dynamics,” in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 6828–6833.

List of symbols and abbreviations

- m [kg] Mass of vehicle including driver
- I_z [$\text{kg} \cdot \text{m}^2$] moment of inertia to Z axis
- CoG_x [m] relative position of center of gravity towards front axle
- CoG_z [m] height of center of gravity
- CoP_x [-] relative position of center of lift towards front axle
- l_f [m] length between front axle and CoG
- l_r [m] length between rear axle and CoG
- t [m] track, distance between left and right wheel. Is assumed to be the same on both axles.
- F_{lon} [N] longitudinal force acting on tire
- F_{lat} [N] lateral force acting on tire
- r_w [m] wheel radius
- ω_w [rad/s] wheel angular velocity
- δ [rad] steering angle
- μ [-] friction coefficient between tire and road
- C_α [N/rad] cornering stiffness of the tire
- $B^{Pac}, C^{Pac}, D^{Pac}, E^{Pac}, F^{Pac}$ [-] Magic Formula coefficients
- τ [Nm] motor torque
- C_L [-] coefficient of lift
- A_{aero} [m^2] front aerodynamic area

List of Figures

Image 1	eForce FSE.11	2
Image 2	Plot of the wheel steering angle as a function of the steering column angle	3
Image 3	Schematics of current control system	4
Image 4	Vehicle Axis System ISO 8855-2011. From [9]	5
Image 5	Simulation showing normals to wheel speed vector during simulation	7
Image 6	Fitted tire lateral force data	9
Image 7	G-G diagram of endurance race with estimated friction ellipse	9
Image 8	Geometrical representation of a single-track model	13
Image 9	Model validated on measured drive data	14
Image 10	Response of the model to unit yaw torque step at 0.2 s at various steer angle at $v_x = 10$ m/s.	16
Image 11	Response of the model to unit yaw torque step at 0.2 s at various velocity at $\delta = 0^\circ$	16
Image 12	Visualisation of possible steering characteristics.	17
Image 13	Surface plot of desired yaw rate for oversteering vehicle	19
Image 14	Comparison between calculated referential yaw rate and actual yaw rate for different understeer gradients	19
Image 15	Closed loop response to steering step at $v_x = 6$ m/s	27
Image 16	Closed loop response to steering step at $v_x = 10$ m/s	28
Image 17	Closed loop response to steering step at $v_x = 14$ m/s	28
Image 18	Closed loop response of LTI-MPC to steering step at $v_x = 6$ m/s	29
Image 19	Closed loop response of LTI-MPC to steering step at $v_x = 14$ m/s	30
Image 20	Closed loop response to disturbance at $v_x = 10$ m/s and $\delta = 0^\circ$ with visualized prediction trajectories for every 5th simulation step	31
Image 21	Closed loop response to disturbance at $v_x = 14$ m/s and $\delta = 0^\circ$	32
Image 22	Closed loop response to disturbance at $v_x = 14$ m/s and $\delta = 5^\circ$	32
Image 23	Measured inputs for simulation	33
Image 24	Simulation of control loop with measured data	33
Image 25	CarMaker virtual driving environment	34
Image 26	Recorded data from virtual drive	35
Image 27	Repeated double lane change test results	36
Image 28	Pure slip lateral force test data from Calspan fitted with Pacejka's magic formula	38
Image 29	Friction ellipse with maximal longitudinal force visualized	38
Image 30	Proposed traction distribution tested on lane change maneuver	40

List of used SW

- Matlab & Simulink R2021a
- IPG CarMaker 10.2
- L^AT_EX

List of attachments

- model_check.slx
- model_check_script.mlx
- closed_loop_simulation.slx
- close_loop_simulation_script.mlx
- derivations.mlx
- yaw_rate_generator.mlx
- IPGCarMaker_torque_vectoring_subsystem.slx
- signalsRun1.mat
- CarMaker_logged_data
 - noTV1.mat
 - noTV2.mat
 - ⋮
 - noTV20.mat
 - TV1.mat
 - TV2.mat
 - ⋮
 - TV20.mat