

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/329511870>

# Model Predictive Torque Vectoring Control for a Formula Student Electric Racing Car

Conference Paper · June 2018

DOI: 10.23919/ECC.2018.8550124

---

CITATIONS

13

READS

2,478

3 authors:



Erik Mikulaš

Freelancer

15 PUBLICATIONS 97 CITATIONS

[SEE PROFILE](#)



Martin Gulan

Slovak University of Technology in Bratislava

56 PUBLICATIONS 362 CITATIONS

[SEE PROFILE](#)



Gergely Takács

Garrett Motion

97 PUBLICATIONS 573 CITATIONS

[SEE PROFILE](#)

# Model Predictive Torque Vectoring Control for a Formula Student Electric Racing Car

Erik Mikuláš, Martin Gulan and Gergely Takács

**Abstract**—In this paper we present two torque vectoring control algorithms for an electric racing car with independent all-wheel drive. A nonlinear, two-track vehicle model is used for the design of a linear time-varying model predictive controller and a nonlinear model predictive controller, while the unknown system states are estimated by the unscented Kalman filter. The controller has been tested in various simulation scenarios and the obtained results are compared with the case assuming equal torque distribution, i.e. without torque vectoring. The results show that the proposed optimization-based torque vectoring control strategy may effectively stabilize the vehicle at the limits of handling and in this way increase its track performance.

## I. INTRODUCTION

The history of electronically controlled torque distribution goes back to the 1980s, when the first cars with such systems were released. Direct yaw control was achieved by activating brakes on individual wheels. Although this technique was effective for pro-active accident prevention, it was not suitable for racing cars, where the deterioration of the longitudinal performance would occur due to braking. The first work that showed a possibility of directly controlled drive torque distribution was presented by Sawase et al. [1] who achieved this by utilizing a special differential with electronically controlled clutches. However, the true power of such systems came to spotlight with the development of electric cars with independently driven wheels. This configuration enables one to control the driving torque precisely and without additional energy losses. Such electronic vehicle systems that directly control driving torque to increase vehicle performance and safety are generally referred to as torque vectoring (TV) systems.

The increasing computing power of microcontrollers and the growing computational efficiency of optimization algorithms has opened new possibilities for the implementation of more advanced control algorithms into vehicle electronic systems. A possible field of application for torque vectoring systems is, of course, in road cars. A well-designed control algorithm can enhance vehicle safety with a small additional cost compared to the improvement of mechanical parts, which is more expensive and usually can not bring a significant result. The other field of application for torque vectoring systems is the racing industry, where most of

\*This work was supported by the Slovak Research and Development Agency (APVV) under the contract APVV-14-0399.

<sup>1</sup>Erik Mikuláš, Martin Gulan and Gergely Takács are with the Institute of Automation, Measurement and Applied Informatics, Faculty of Mechanical Engineering, Slovak University of Technology in Bratislava, 81231 Bratislava, Slovakia erik.mikulas@stuba.sk; martin.gulan@stuba.sk; gergely.takacs@stuba.sk

the cars are already mechanically superb so any electronic system that can give an edge over rivals is highly appreciated.

The torque vectoring control problem is conventionally addressed by a proportional-integral-derivative controller (PID) [2], which provides satisfactory control quality for standard driving conditions. However, racing cars are often driven near the limits of handling, where vehicle behavior becomes highly nonlinear and therefore linear control strategies are not sufficient. Thus, nonlinear control algorithms are preferred for racing scenarios. Optimization-based strategies such as model predictive control (MPC) that employs constrained minimization of a quadratic cost function are an excellent match for vehicles on the edge of performance and physical possibilities. Thereby, the inherent constraint handling of MPC pushes the performance to the boundaries—just what a racing car requires. Several works utilize linear time-varying model predictive controllers (LTV-MPC) for this reason. An LTV-MPC scheme along with the unscented Kalman filter (UKF) for state estimation was developed for vehicles equipped with an independent rear-wheel drive in [3], [4], [5]. A similar design for all-wheel drive vehicles was presented by Vasiljevic and Bogdan in [6]. Two nonlinear model predictive control strategies (NMPC) were presented recently in [7], yet assuming only a rear wheel driven vehicle. Furthermore, fuzzy logic controllers that are otherwise seldom used for torque vectoring are presented in [2] and [8]. In 2013, Mercedes-Benz introduced SLS AMG Electric Drive [9] as a showcase of what can be practically achieved by torque vectoring. The control algorithms are not publicly known yet but the results are remarkable [9]. To bear on our knowledge MPC-based torque vectoring was not implemented on formula student race car.

Unlike previous works, this paper focuses on the design of a model predictive torque vectoring controller for a formula student electric racing car with all-wheel drive. Due to the utilization of extremely high performance tires and great driving torque, this car operates on the edge of its performance envelope, where the behavior of the tire becomes highly nonlinear. To this end we propose a LTV-MPC torque vectoring system, which is subsequently compared to an alternative NMPC torque vectoring approach.

This paper is organized as follows. Section II introduces a nonlinear two-track model of the car. Section III presents torque vectoring fundamentals and objectives. Section IV presents the proposed LTV-MPC and NMPC strategy with reference tracking and constraints. Finally, in Section V, the performance of the proposed control strategies is demonstrated in two simulation scenarios.

TABLE I  
PARAMETERS OF THE SGT-FE18 RACING CAR

Parameter	Symbol	Value	Unit
Weight	$m$	235	kg
Front track	$w_f$	1.22	m
Rear track	$w_r$	1.19	m
Wheel base	$l$	1.57	m
Center of gravity front arm	$l_f$	0.71	m
Center of gravity rear arm	$l_r$	0.86	m
Center of gravity height	$h_{cm}$	0.25	m
Wheel radius	$R$	0.22	m
Aero drag force at 25 ms <sup>-1</sup>		1100	N
Aero down force at 25 ms <sup>-1</sup>		380	N
Max torque per motor		21	Nm
Max power per motor		36	kW
Max combined power (regulated)		144	kW
Transmission ratio	$i$	13.9	-

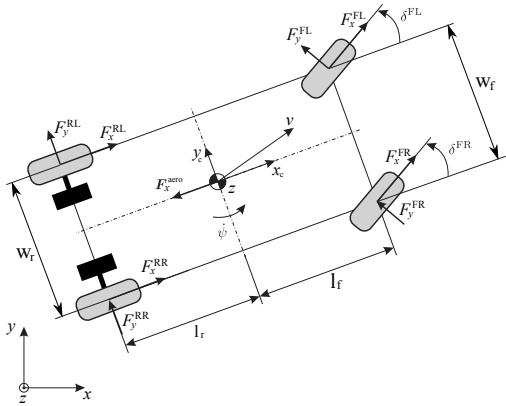


Fig. 1. Forces acting on the vehicle body.

## II. VEHICLE MODEL

A model was developed to aid simulations and controller design. The real-world vehicle is a formula student electric (FSE) racing car named 'SGT-FE18'. The car is currently being built by FSE team STUBA Green Team at the Slovak University of Technology in Bratislava, for the 2018 racing season. The vehicle has a carbon fiber monocoque structure, lightweight aerodynamic package to create down force and all four wheels are driven by independent electric motors through two-stage planetary gearboxes.

In order to simplify the vehicle model, let us assume that the vehicle is moving only on a flat surface, i.e. the effects of suspension are omitted; therefore there is no roll and pitch and the tire rolling resistance is negligible. The model itself is divided into three parts: equations of motion for the rigid vehicle body, wheel dynamics and tire model. The parameters of the vehicle are summarized in Table I.

### A. Planar two-track vehicle model

The equations of motion for the vehicle body contain the forces acting on the vehicle, and these are shown in Fig. 1: the aerodynamic drag  $F_x^{\text{aero}}$  and the tire/road contact forces  $F_k^{ij}(\ddot{x}, \ddot{y}, \ddot{\psi}, \dot{x}, \dot{y}, \dot{\psi})$ , where the superscript  $i$  stands for the front or the rear of the vehicle, i.e.  $i = \{\text{F}, \text{R}\}$ , while the superscript  $j$  is used to distinguish the right and the left side, i.e.  $j = \{\text{R}, \text{L}\}$ , and  $k = \{x, y\}$  indicates the direction in

which the variable is defined. The component equations for longitudinal, lateral and yaw acceleration are as follows [6], [10]:

$$\dot{v}_x = \frac{1}{m}(F_x^{\text{FR}} \cos \delta^{\text{FR}} + F_x^{\text{FL}} \cos \delta^{\text{FL}} - F_y^{\text{FR}} \sin \delta^{\text{FR}} - F_y^{\text{FL}} \sin \delta^{\text{FL}} + F_y^{\text{RR}} + F_x^{\text{RL}} - F_x^{\text{aero}}) - \dot{\psi} v_y, \quad (1)$$

$$\dot{v}_y = \frac{1}{m}(F_x^{\text{FR}} \sin \delta^{\text{FR}} + F_x^{\text{FL}} \sin \delta^{\text{FL}} + F_y^{\text{FR}} \cos \delta^{\text{FR}} + F_y^{\text{FL}} \cos \delta^{\text{FL}} + F_y^{\text{RR}} + F_x^{\text{RL}}) + \dot{\psi} v_x, \quad (2)$$

$$\begin{aligned} \ddot{\psi} = & \frac{1}{I_z}(l_F(F_x^{\text{FR}} \sin \delta^{\text{FR}} + F_y^{\text{FR}} \cos \delta^{\text{FR}} + F_x^{\text{FL}} \sin \delta^{\text{FL}} + F_y^{\text{FL}} \cos \delta^{\text{FL}}) - l_R(F_y^{\text{RR}} + F_y^{\text{RL}}) + \frac{w_F}{2}(-F_x^{\text{FR}} \cos \delta^{\text{FR}} \\ & + F_y^{\text{FR}} \sin \delta^{\text{FR}} + F_x^{\text{FL}} \cos \delta^{\text{FL}} - F_y^{\text{FL}} \sin \delta^{\text{FL}}) + \frac{w_R}{2}(+F_x^{\text{RR}} - F_x^{\text{RL}})), \end{aligned} \quad (3)$$

where all the parameters are according to Tab. I. The steering angles  $\delta^{F,j}$  are considered only for the front wheels. The toe angle on the rear wheels is omitted. The rotational dynamics for each wheel is given by

$$\dot{\omega}^{ij} = \frac{1}{I_{\text{red}}^i}(M_p^{ij} - M_b^{ij} - F_x^{ij}R), \quad (4)$$

where  $M_p$  denotes the propulsion torque generated by the motors and  $M_b$  is the torque created by the brake.

### B. Tire model

In order to determine the longitudinal and lateral tire forces, version 5.2 of the Pacejka's well-known 'Magic Formula' tire model is used [11]. This model can be expressed as set of semi-empirical functions given in the form of

$$F_x^{ij} = F_{x0}^{ij}(\kappa^{ij}, F_n^{ij}) G_{y\alpha}^{ij}(\alpha^{ij} \kappa^{ij}, F_n^{ij}), \quad (5a)$$

$$\begin{aligned} F_y^{ij} = & F_{y0}^{ij}(\alpha^{ij}, F_n^{ij}) G_{x\kappa}^{ij}(\alpha^{ij} \kappa^{ij}, \gamma^{ij} F_n^{ij}) + \\ & S_{V\gamma\kappa}(\alpha^{ij} \kappa^{ij}, \gamma^{ij} F_n^{ij}), \end{aligned} \quad (5b)$$

where  $F_{x0}^{ij}$  and  $F_{y0}^{ij}$  denote forces for pure lateral and pure longitudinal slip,  $G_{y\alpha}^{ij}$  and  $G_{x\kappa}^{ij}$  are the weighting functions for combined slip,  $S_{V\gamma\kappa}$  is vertical shift in the lateral force due to camber and normal force for combined slip and  $\kappa^{ij}$  is the longitudinal slip ratio of individual wheels given by

$$\kappa = \frac{\omega R - v_{x,w}}{v_{x,w}}. \quad (6)$$

Furthermore,  $\alpha^{ij}$  denotes the slip angle of individual wheels and is given by

$$\alpha = \arctan \frac{v_y + \dot{\psi} l_i}{|v_x - \dot{\psi} \frac{w_i}{2}|} - \delta, \quad (7)$$

where  $\gamma$  is the camber angle of the wheel that is assumed constant while  $F_n^{ij}$  is the normal force acting in the contact point between the tire and the road for individual wheels, and is obtained from static load distribution and load transfer induced by longitudinal and lateral acceleration.

### C. State-space model formulation

Let us now assume that the dynamics is represented by the nonlinear state-space model given by Eqs. (1)–(7) as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad \mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)), \quad (8)$$

where  $\mathbf{x}(t) = [v_x, v_y, \dot{\psi}, \omega^{\text{FR}}, \omega^{\text{FL}}, \omega^{\text{RR}}, \omega^{\text{RL}}]^T$  is the state vector and  $\mathbf{u}(t) = [M_p^{\text{FR}}, M_p^{\text{FL}}, M_p^{\text{RR}}, M_p^{\text{RL}}]^T$  is the input vector.

For the purpose of controller design the fast wheel dynamics was neglected, in order to reduce execution time, allow for fairly long sampling periods and enhance numerical stability of controller similarly as in [3]. This simplification may increase the feasibility of real-time implementation on computing devices with limited computational performance, note that even nominal linear MPC required more than 1ms execution time on microcontrollers. After this simplification the state vector will be  $\mathbf{x}(t) = [v_x, v_y, \dot{\psi}]^T$  and input vector  $\mathbf{u}(t) = [M_p^{\text{FR}}, M_p^{\text{FL}}, M_p^{\text{RR}}, M_p^{\text{RL}}]^T$ . Subsequently, the simplified version of the model in Eq. (8) was linearized and discretized to get the LTV state-space model described by

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k), \quad (9)$$

where the discrete-time state-transition matrix  $\mathbf{A}_k \in \mathbb{R}^{3 \times 3}$  and input matrix  $\mathbf{B}_k \in \mathbb{R}^{3 \times 4}$  are evaluated for the current operating point in each sample.

## III. CONTROLLER DESIGN

The fundamentals of the proposed LTV-MPC and NMPC control strategies along with torque vectoring objectives and a state estimation framework are presented in this section.

### A. Torque vectoring principles

A torque vectoring system should manage driving torque distribution between the driven wheels. In some cases, it is even possible to apply a negative torque to brake individual wheels, which is instrumental in small radius cornering. If the battery management system enables braking energy recuperation, it is possible to apply a greater braking power than driving power. In case energy recuperation is not possible—like it is assumed here—the total power must be zero or greater.

The driving torque should be distributed in a way that will ensure a stable behavior of the vehicle. The torque vectoring system cannot allow torque resulting in a large tire longitudinal slip ratio. The maximum torque that the tires can transfer to the road surface is obtained form the mathematical model of the tire in Eq. (5). Since we neglect wheel dynamics, we assume that lateral tire forces correspond to the amount of torque applied to the wheels while the maximum longitudinal forces that the tires can provide are taken into account on the level of input constraints for both cases.

### B. State estimation

We have chosen to implement the unscented Kalman filter (UKF) for state estimation [12]. The UKF enabled to utilize the full nonlinear vehicle model in Eqs. (1)–(7) while sustaining lower computational effort compared to the extended Kalman filter (EKF). The 6-step Euler method was used for the discretization of continuous-time dynamics. We assumed that the vehicle is equipped with wheel speed sensors and an inertial measurement unit (IMU), the data provided by these sensors are taken as the output of the system. The simulated output vector with additive measurement noise is given as  $\mathbf{y}(k) = [\dot{v}_x, \dot{v}_y, \ddot{\psi}, \omega^{\text{FR}}, \omega^{\text{FL}}, \omega^{\text{RR}}, \omega^{\text{RL}}]^T$ . Consequently, the state vector estimated by the UKF can be written as  $\hat{\mathbf{x}}(k) = [\hat{v}_x, \hat{v}_y, \hat{\psi}, \hat{\omega}^{\text{FR}}, \hat{\omega}^{\text{FL}}, \hat{\omega}^{\text{RR}}, \hat{\omega}^{\text{RL}}]^T$ .

### C. Reference calculation

Both control strategies share the same state reference formulated similarly as in [6] and calculated according to the current state, the driver input and the handling limits of the vehicle as  $\mathbf{x}_{\text{ref}} = [v_{x,\text{ref}} \ v_{y,\text{ref}} \ \dot{\psi}_{\text{ref}}]^T$ .

The longitudinal speed reference  $v_{x,\text{ref}}$  is chosen according to the current speed, motor torque requested by driver  $M_{\text{dr}}$ , length of prediction horizon  $n_p$  and the maximum allowable speed  $v_{\max}$  that is dependent on the maximal lateral force that tires can handle  $F_{y,\max}$ , determined from the tire model [11] and the steady-state cornering radius  $R_{\text{ss}}$ , which in turn is determined from the car's geometry and steering input as

$$R_{\text{ss}} = \frac{l_F + l_R}{\frac{\delta^R + \delta^L}{2}}, \quad (10)$$

$$v_{\max}^2 = \frac{R_{\text{ss}} F_{y,\max}}{m}, \quad (11)$$

$$v_{x,\text{ref}} = \min \left( \sqrt{|v_x^2 + v_y^2|} + \frac{i M_{\text{dr}}}{R m} n_p T_s, \sqrt{|v_{\max}^2 - v_y^2|} \right). \quad (12)$$

Lateral speed reference  $v_{y,\text{ref}}$  is chosen as the actual lateral speed being saturated at the value that results in the maximal allowable vehicle slip angle  $\beta_p$  that is determined from tire characteristics provided by the tire manufacturer, and is given by

$$v_{y,\text{ref}} = \text{sign}(v_y) \min(|v_y|, \tan(\beta_p) v_x). \quad (13)$$

The yaw rate reference  $\dot{\psi}_{\text{ref}}$  is calculated from the geometry of the car, actual longitudinal speed, steering angle and the under-steer coefficient  $K_{\text{us}}$ , which can be chosen with respect to the desired car behavior. Positive values correspond to under-steer behavior while negative values imply over-steer behavior. Usually, the desired behavior for racing car is neutral-steer, then corresponding value of the coefficient will be  $K_{\text{us}} = 0$ . Thus, the reference yaw rate will be

$$\dot{\psi}_{\text{ref}} = \frac{v_x \tan(\frac{\delta_R + \delta_L}{2})}{(l_R + l_F) + K_{\text{us}} v_x^2}. \quad (14)$$

#### D. Linear MPC

The linear MPC formulation assumed here is given by the following constrained optimization problem

$$\min_{\mathbf{U}} \sum_{i=0}^{n_p-1} (\|\mathbf{x}_{k+i} - \mathbf{x}_{\text{ref}}\|_{\mathbf{Q}}^2 + \|\mathbf{u}_{k+i}\|_{\mathbf{R}_1}^2), \quad (15a)$$

$$\text{s.t. } \mathbf{x}_k = \hat{\mathbf{x}}(k), \quad (15b)$$

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k, \quad (15c)$$

$$\underline{\mathbf{u}} \leq \mathbf{u}_{k+i} \leq \bar{\mathbf{u}} \quad i = 0, 1, \dots, n_p - 1, \quad (15d)$$

where  $n_p$  is the length of the prediction horizon and the optimal solution of this problem is the sequence of optimal future inputs  $\mathbf{U}^* = [\mathbf{u}_k^*, \mathbf{u}_{k+1}^*, \dots, \mathbf{u}_{k+n_p-1}^*] \in \mathbb{R}^{n_p n_u \times 1}$ . The current state  $\hat{\mathbf{x}}_k$ —as estimated by the UKF—is chosen as the operating point for LTV state-space model. The states are weighted by  $\mathbf{Q} \in \mathbb{R}^{n_x \times n_x}$  within the quadratic objective Eq. (15a), while inputs are penalized  $\mathbf{R} \in \mathbb{R}^{n_u \times n_u}$ . In our simulation study, (15a)–(15d) are solved by the qpOASES solver, in the form of a quadratic programming (QP) problem [13] in MATLAB/Simulink. Note, that our choice of QP solver is not arbitrary, as qpOASES have been proven to be real-time feasible on limited computational hardware before [14].

Input constraints are recalculated at every sampling period according to the maximal forces that tires can handle at the current state and motor limitations. To maintain a predictable vehicle behavior, there are constraints for the sum of all torques that cannot exceed the torque requested by the driver. This ensures that the torque is only distributed between wheels or reduced to decrease vehicle speed when the cornering radius is infeasible according to Eq. (12).

#### E. Nonlinear MPC

The considered nonlinear MPC scheme is based on solving the following optimal control problem

$$\min_{\mathbf{U}, \mathbf{X}} \frac{1}{2} \int_{t_0}^{t_0 + n_p T_s} (\|\mathbf{x} - \mathbf{x}_{\text{ref}}\|_{\mathbf{Q}}^2 + \|\mathbf{u} - \mathbf{u}_{\text{ref}}\|_{\mathbf{R}_2}^2) dt, \quad (16a)$$

$$\text{s.t. } \mathbf{x}(t_0) = \hat{\mathbf{x}}(t_0), \quad (16b)$$

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (16c)$$

$$\mathbf{h}_{\text{ineq}} \leq 0, \quad (16d)$$

where (16d) represent inequality constraints and  $\mathbf{X} \in \mathbb{R}^{n_p n_x \times 1}$  vector of predicted states. In order to numerically solve the nonlinear control scheme presented above we make use of the ACADO Code Generation tool [15], exploiting direct multiple-shooting, real-time iteration scheme and sequential quadratic programming using its MATLAB interface. The mathematical model used in this controller is the natively nonlinear state-space model in Eq. (8) which is internally discretized by ACADO. Since the rotational dynamics of the wheels was neglected, there is no way to calculate  $\kappa^{i,j}$  internally, so we used simplified inverted tire model to calculate  $\kappa^{i,j}$  from the current state and the applied torque. To prevent creating an algebraic loop, normal forces were

assumed constant during the prediction horizon. To ensure a maximum possible longitudinal performance, we introduced an additional state  $\dot{s} = v_x$ , representing the distance traveled in the longitudinal direction. The augmented state vector takes form  $\mathbf{x}(t) = [v_x, v_y, \dot{\psi}, s]^T$ . To achieve this effect, we chose the reference distance larger than it is possible to travel during the prediction horizon. The constraints were applied to inputs, slip angles  $\alpha^{i,j}$  and to the sum of power.

#### F. Driver Command

In order to emulate driver inputs, a simple path following driver model was designed based on PID controllers. This model controls the accelerator and brake pedals—that directly represents the amount of requested driving or braking torque—according to a defined speed reference. Furthermore, the driver model controls steering input according to the distance of the car from the track reference and yaw angle. Subsequently, the steering angle  $\delta$  for the left and right wheel is calculated according to the Ackerman geometry:

$$\cot \delta_I = \cot \delta - \frac{l}{2w_f}, \quad \cot \delta_O = \cot \delta + \frac{l}{2w_f}, \quad (17)$$

where  $\delta_I$  is the inner wheel angle and  $\delta_O$  is the outer wheel angle during cornering.

## IV. SIMULATION RESULTS

In this section, we present simulation results and the comparison of vehicle behavior with torque vectoring and with equal torque distribution.

The setup of the LTV-MPC controller was the same for all simulations. The sampling period for the controller and the estimator was  $T_s = 0.002$  s, the prediction horizon length was chosen  $n_p = 20$  steps and the weighting matrices were set to

$$\mathbf{Q} = \text{diag}(100, 10, 10),$$

$$\mathbf{R}_1 = \text{diag}(0.002, 0.002, 0.002, 0.002).$$

In order to ensure a fair comparison of the two torque vectoring strategies, most parameters were set in such way that the effect of them on the closed-loop response is comparable. This included, horizon length and state penalty, while in the interest to fine-tune performance the input penalty of the NMPC controller was chosen as

$$\mathbf{R}_2 = \text{diag}(0.1, 0.1, 0.1, 0.1).$$

The sampling time of NMPC was chosen as  $T_s = 0.01$  s to enable the practical tractability of simulations with 20 integrator substeps within one sample. The interested reader may find more details on the simulation settings, in particular the UKF, in [16].

#### A. U-turn scenario

##### U-turn scenario with 10 m radius at the limit of handling

This scenario demonstrates how torque vectoring prevented vehicle spin, then later during the acceleration after the apex, helped to maintain the cornering radius. As it can

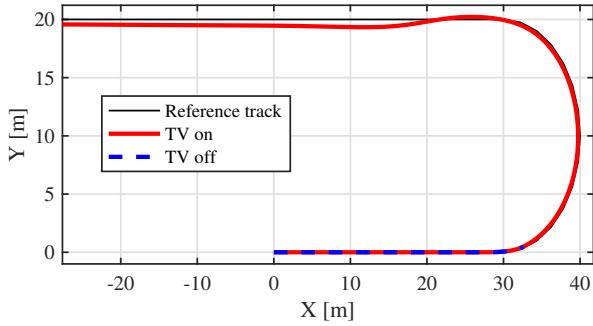


Fig. 2. Trajectory for the U-turn with a 10 m corner radius at the limit of handling.

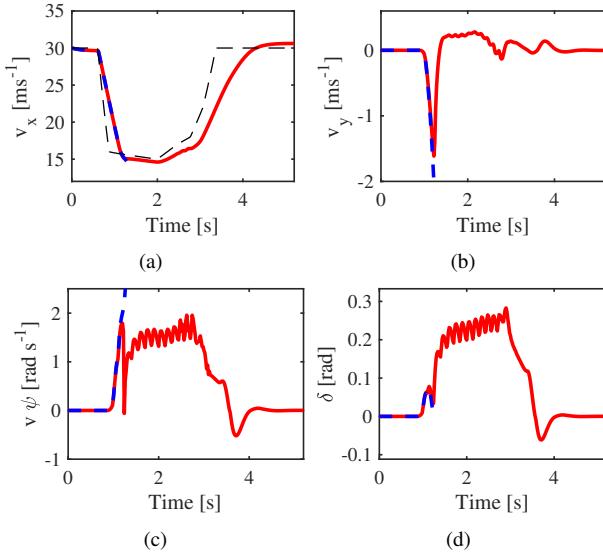


Fig. 3. States for the U-turn with a 10 m corner radius with LTV-MPC at the limit of handling, (a) longitudinal velocity, (b) lateral velocity, (c) yaw rate, (d) steering input.

be observed in Fig. 2, the vehicle trajectory with active torque vectoring (red line) precisely copied the reference track and the trajectory of the vehicle with equal torque distribution (dashed blue line) ended up in a spin shortly after entering the corner. Note that the lateral offset is due to the imperfect driver model.

Figure 4 shows the torque distribution for the previously introduced simulation study. The reader may note that between the time 1.00–1.25 s more torque was distributed to the left—inner—wheels, while the torque to the right—outer—wheel was actually negative in order to prevent vehicle spin while entering the corner. During the first half of the corner, torque was distributed almost equally and subsequently, after reaching the apex and when speed started to increase, more torque was delivered to the outer wheels to reach an otherwise infeasible cornering radius.

Moreover, the imperfection and the limitation of the simple driver model can be observed in Fig. 3 showing the state trajectories. The oscillating steering input in Fig. 3(d) caused corresponding oscillations in the yaw rate shown in Fig. 3(c) and lateral velocity in Fig. 3(b). The torque distribution was

also affected on a smaller scale and this is demonstrated in Fig. 4, between the times 1.25–2.00 s.

#### *U-turn scenario with a 10 m radius at 16 ms<sup>-1</sup>*

This simulation compares the LTV-MPC and the NMPC controlled torque distribution strategies. It is interesting to observe the effect of sampling speed on the performance of the controllers: NMPC showed the best results at a sampling period set to  $T_s = 0.01$  s, while the LTV-MPC controller was numerically more stable at  $T_s = 0.002$  s. However, to ensure a fair comparison of the methods, both controllers were sampled at  $T_s = 0.01$  s for this test.

The reader may note that the trajectories for the two torque vectoring strategies shown in Fig. 5 are almost identical. However, one may also see significant differences in the model states shown in Fig. 6, the torque distribution for the LTV-MPC in Fig. 7 and the NMPC in Fig. 8.

The NMPC strategy reduced the longitudinal velocity in Fig. 6(a) more than LTV-MPC, but at the end of the corner it also increased the speed immensely. Figure 6(b) shows that the NMPC controller actually caused marginal understeer behavior rather than the oversteer induced by the LTV-MPC controller. This is actually beneficial at high-speed cornering, where understeer is more stable.

The difference between the torque distribution for both torque vectoring strategies is evident from Fig. 7 (LTV-MPC) and Fig. 8 (NMPC).

In the case of the LTV-MPC strategy, a greater amount of torque goes to the inner wheels during the entire duration of the cornering maneuver. However, in the case of NMPC-based torque vectoring, more torque is supplied to the inner wheels in the first part of the cornering maneuver, but the controller is fluently increasing the torque on the outer wheels as the vehicle passes through the corner.

#### *B. Steering step response*

This simulation scenario demonstrates the torque vectoring response to a step change of the steering angle. The reference speed for the driver model is constant, hence any changes of speed are caused by torque vectoring systems only.

#### *Steering step response at 17.5 ms<sup>-1</sup> for LTV-MPC*

This simulation shows how the LTV-MPC controller reacted to a step change of the steering angle, along with a corresponding change of yaw rate reference. Figure 9 shows the trajectory of two simulated vehicles, where one may observe that the vehicle with equal torque distribution would eventually end up in a spin, while a vehicle with LTV-MPC torque vectoring will manage to stay on track.

The corresponding torque distribution chart shown in Fig. 10 demonstrates an immediate response of the controller that stabilizes the vehicle and subsequently applies negative torque to decrease the longitudinal speed. The states in Fig. 13 hold another interesting detail. As for the given cornering radius the tire forces will be less than the centrifugal force (i.e. grip is lost), the LTV-MPC torque vectoring strategy must decrease the longitudinal speed reference in order for

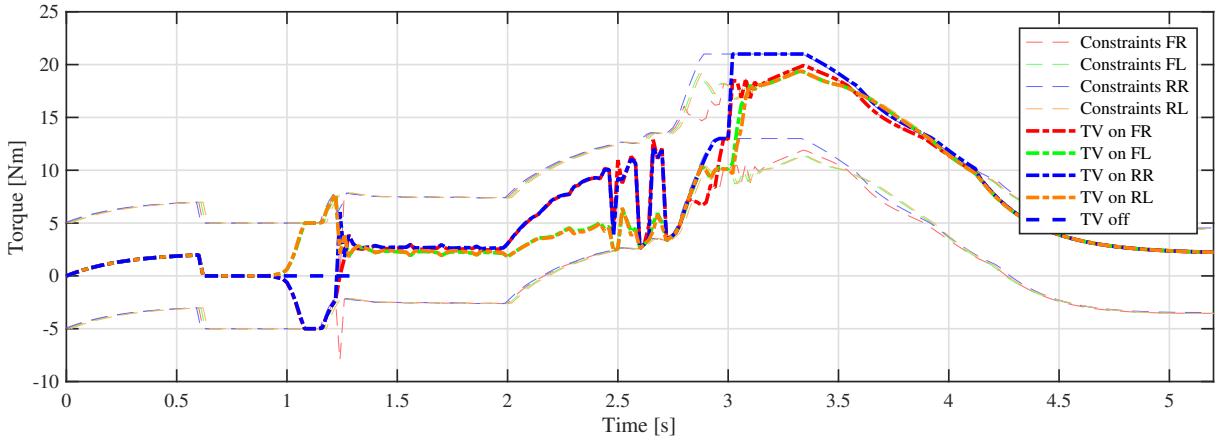


Fig. 4. Torque distribution for the U-turn with 10 m corner radius at the limit of handling.

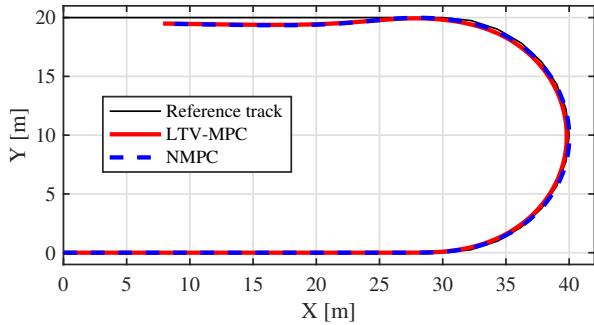


Fig. 5. Trajectory for the U-turn with a 10 m corner radius, assuming LTV-MPC (red) and NMPC (blue dashed) torque vectoring at  $16 \text{ ms}^{-1}$ .

the tire forces to be less than the centrifugal force. This behavior can be observed in Fig. 13(a). For the remainder of the simulation the torque distribution remained steady.

#### *Steering step response at $17.5 \text{ ms}^{-1}$ for NMPC*

This simulation is performed under the same conditions as the previous one, but this time using the NMPC controller. The reader may note that the results are largely similar to the previous case.

Nevertheless, the most significant difference between the two proposed strategies is in the torque distribution shown in Fig. 11, where the initial controller reaction was less aggressive but eventually managed to increase the longitudinal speed (Fig. 12(a)). This caused oscillations of torque distribution and subsequently the oscillations of lateral velocity (Fig. 12(b)) and yaw rate (Fig. 12(c)).

## V. CONCLUSION

We have presented two model predictive torque vectoring control systems for a racing car with independent all wheel drive. One strategy was based on LTV-MPC and the other on the NMPC framework. A nonlinear two-track mathematical model of the vehicle with nonlinear tire model was employed, which was linearized at every sampling period for the LTV-MPC controller. The reference and constraints were generated according to handling limits and the capabilities of the propulsion system for both controllers.

The simulation results showed that the proposed torque vectoring systems were able to stabilize the vehicle near the limits of the handling, prevented vehicle spin and enabled to reach a smaller cornering radius than one possible with equal torque distribution at given speed. Furthermore, the steering step response simulation demonstrated a decreasing speed that was limited by the controller in order to prevent lateral acceleration hitting the constraints.

While it is clear that implementing a natively nonlinear TV strategy would be more demanding from the viewpoint of computational efficiency, the inherent advantage of NMPC over its linear counterpart is better state trajectory predictions and ultimately a vehicle behavior that is absolutely on the

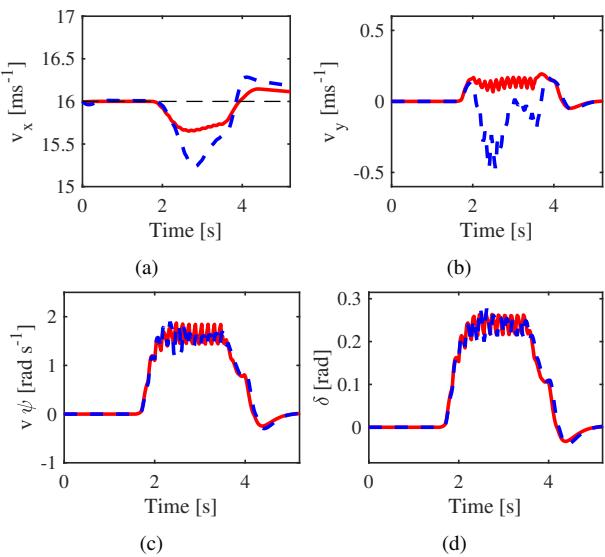


Fig. 6. States for the U-turn with 10 m corner radius with LTV-MPC (red) and NMPC (blue dashed) at  $16 \text{ ms}^{-1}$  (a) longitudinal velocity where dashed black line is reference for driver, (b) lateral velocity, (c) yaw rate, (d) steering input.

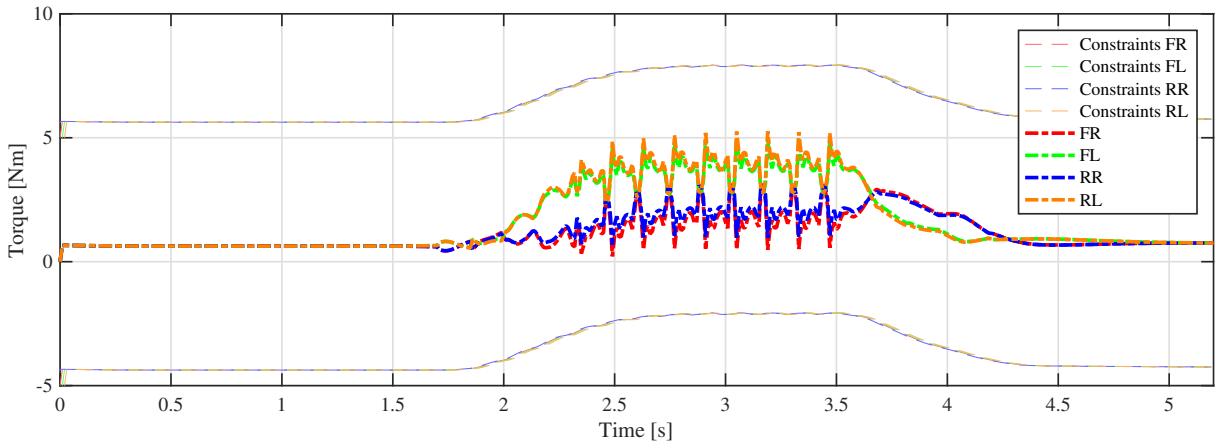


Fig. 7. Torque distribution for the U-turn maneuver with 10 m cornering radius achieved with LTV-MPC at  $16 \text{ ms}^{-1}$ .

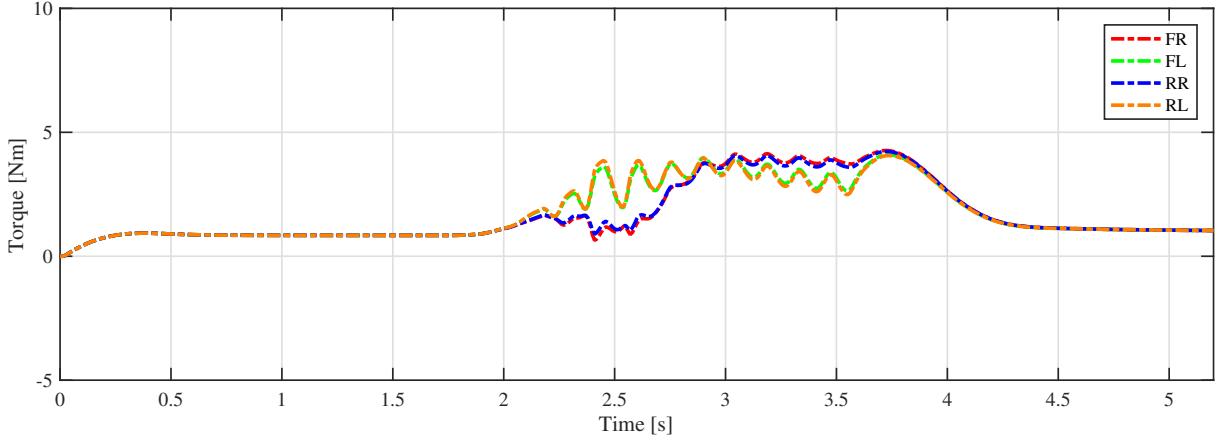


Fig. 8. Torque distribution for the U-turn maneuver with 10 m cornering radius achieved with NMPC at  $16 \text{ ms}^{-1}$ .

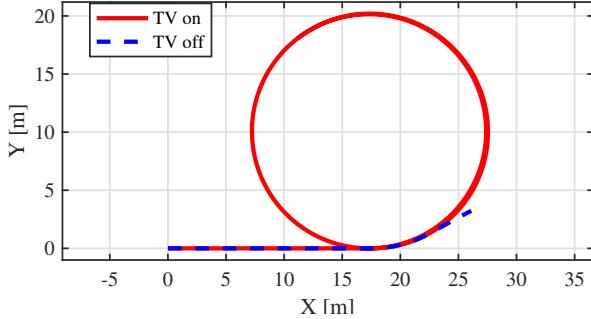


Fig. 9. Trajectory for the steering step response at  $17.5 \text{ ms}^{-1}$ .

limits of handling. Note, that even though the simulations presented in this paper show that LTV-MPC apparently outperforms NMPC but a refinement of the model and the NMPC framework shall enable us to clearly surpass the performance achieved by LTV-MPC only.

Future work will optimize the controller algorithm to increase sampling period and decrease computational effort in order to enable a real-world deployment on embedded computing hardware with limited performance. Furthermore, lap-time simulation using an industry-standard vehicle dy-

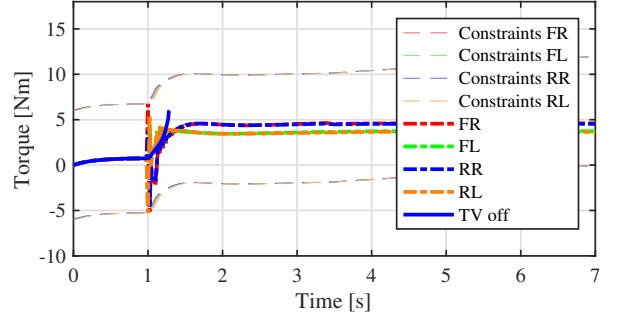


Fig. 10. Torque distribution for the steering step response at  $17.5 \text{ ms}^{-1}$ .

namics simulation tool such IPG CarMaker or ADAMS/Car would help to tune the controllers and show the overall benefit of the proposed torque vectoring systems.

## REFERENCES

- [1] K. Sawase and Y. Sano, "Application of active yaw control to vehicle dynamics by utilizing driving/breaking force," *JSAE Review*, pp. 289 – 295, 1999.
- [2] B. Jäger, P. Neugebauer, R. Kriesten, N. Parspour, and C. Gutenkunst, "Torque-vectoring stability control of a four wheel drive electric vehicle," in *26th Intelligent Vehicles Symposium*, 2015, pp. 1018–1023.

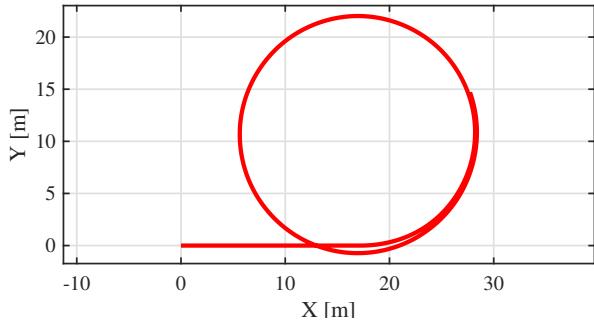


Fig. 14. Trajectory for the steering step response with NMPC at  $17.5 \text{ ms}^{-1}$ .

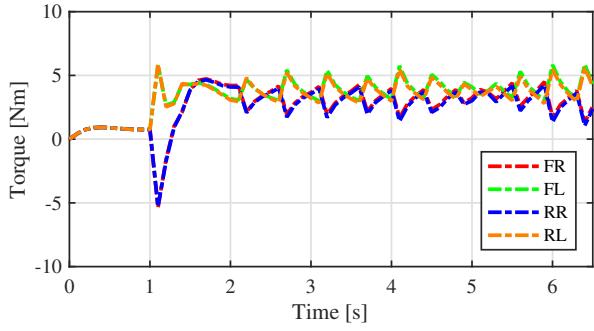


Fig. 11. Torque distribution for the steering step response with NMPC at  $17.5 \text{ ms}^{-1}$ .

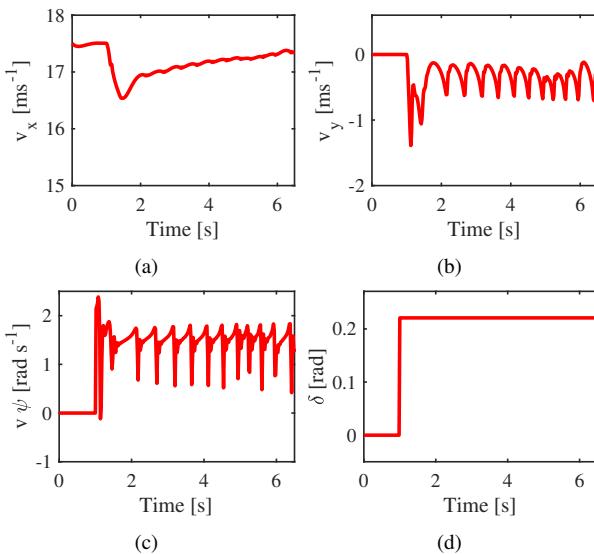


Fig. 12. States for the steering step response with NMPC at  $17.5 \text{ ms}^{-1}$  (a) longitudinal velocity, (b) lateral velocity, (c) yaw rate, (d) steering input.

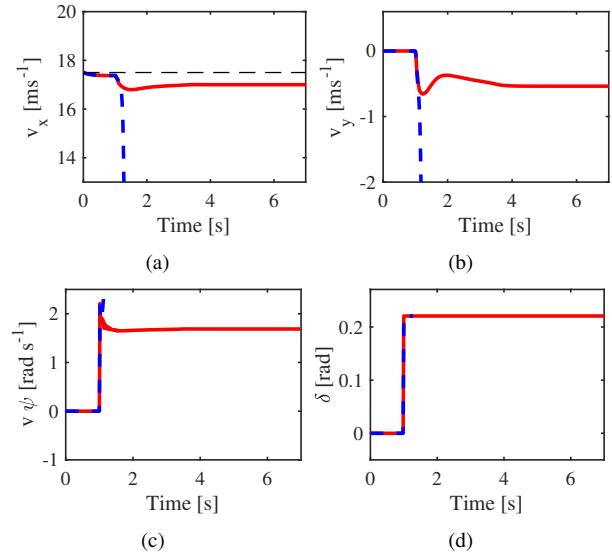


Fig. 13. States for the steering step response at  $17.5 \text{ ms}^{-1}$  (a) longitudinal velocity, (b) lateral velocity, (c) yaw rate, (d) steering input.

- [3] E. Siampis, M. Massaro, and E. Velenis, "Electric rear axle torque vectoring for combined yaw stability and velocity control near the limit of handling," in *52th Conference on Decision and Control*, 2013, pp. 1552–1557.
- [4] E. Siampis, E. Velenis and S. Longo, "Predictive rear wheel torque vectoring control with terminal understeer mitigation using nonlinear estimation," in *54th Conference on Decision and Control*, 2015, pp. 4302–4307.
- [5] E. Siampis, E. Velenis, and S. Longo, "Model predictive torque vectoring control for electric vehicles near the limits of handling," in *14th European Control Conference*, 2015, pp. 2553–2558.
- [6] G. Vasiljevic and S. Bogdan, "Model predictive control based torque vectoring algorithm for electric car with independent drives," in *24th Mediterranean Conference Control and Automation*, 2016, pp. 316–321.
- [7] E. Siampis, E. Velenis, S. Gariuolo, and S. Longo, "A real-time nonlinear model predictive control strategy for stabilization of an electric vehicle at the limits of handling," *IEEE Transactions on Control Systems Technology*, pp. 1–13, 2017.
- [8] J. Park, H. Jeong, I. G. Jang, and S.-H. Hwang, "Torque distribution algorithm for an independently driven electric vehicle using a fuzzy control method," *Energies*, pp. 8537–8561, 2015.
- [9] S. L. J. Feustel and M. Hand, "The electric super sports car SLS AMG electric drive," *ATZ worldwide*, pp. 4–10, 2013.
- [10] S. Antonov, A. Fehn, and A. Kugi, "Unscented Kalman filter for vehicle state estimation," *Vehicle System Dynamics*, pp. 1497–1520, 2011.
- [11] H. B. Pacejka, *Tyre and Vehicle Dynamics*. Butterworth Heinemann, 2006.
- [12] E. A. Wan and R. V. D. Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 2000, pp. 153–158.
- [13] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.
- [14] G. Batista, G. Takács, and B. Rohal-Ilkiv, "Application aspects of active-set quadratic programming in real-time embedded model predictive vibration control," in *20th World Congress of IFAC*, Jun 2017, pp. —.
- [15] B. Houska, H. Ferreau, and M. Diehl, "ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.
- [16] E. Mikuláš, "Model predictive torque vectoring control for the SGT-FE17 racing car," Master's thesis, Slovak University of Technology in Bratislava, Bratislava, Slovakia, 2017.