Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Near Time Optimal Trajectory Generation for Over-Actuated Vehicles using Nonlinear Model Predictive Controller

Hannes Wind, * **Frieder Gottmann,** * **Oliver Sawodny** *

* *Institute for System Dynamics, University of Stuttgart, 70563 Stuttgart, Germany (e-mail: {wind,gottmann,sawodny}@isys.uni-stuttgart.de).*

Abstract: The generation of time dependent paths is crucial for autonomous driving. While relatively simple models are sufficient for normal driving situations, more complex models are required the closer the trajectory is planned towards the handling limits. This paper considers a near time optimal trajectory generation for combined longitudinal and lateral dynamics for an over actuated vehicle. A nonlinear model predictive controller which accounts for actuator constraints is used to generate these trajectories. The predictive model is a reduced dynamic double track model with three degrees of freedom. The resulting trajectories show the ability of following a reference path with the use of all available actuators up to the handling limits.

Keywords: Nonlinear predictive control, Trajectory and Path Planning, Autonomous Vehicles

1. INTRODUCTION

Nowadays, vehicles are equipped with many active safety systems to improve driving safety (Heißing (2013)). Some of these systems need additional actuators which lead to over-actuated vehicles. Autonomous driving can be a future step in developing these active safety systems. Therefore, it is fundamental to plan time dependent paths called trajectories as mentioned in Werling (2011). This paper considers a near time optimal trajectory generation for over actuated vehicles. The near time optimal trajectories imply that the vehicle is operated up to the dynamic limits. The resulting trajectories can be examined to optimally use the over actuation for an obstacle avoidance. Furthermore, a system for automated racing or a racing trainer can be realized based on the resulting trajectories.

Gutjahr et al. (2016) presents a trajectory generation algorithm using a linear time-varying model predictive controller. The lateral dynamics are considered by introducing additional constraints.

Attia et al. (2014) considers a nonlinear model predictive controller for the lateral dynamics of the vehicle which is based on a nonlinear bicycle model. Inputs are the front-wheel-steering as well as the braking and driving torque. The longitudinal dynamics, which is controlled using a Lyapunov approach, is combined with the lateral dynamics via a geometric and velocity reference.

Katriniok and Abel (2011) and Katriniok et al. (2013) use a nonlinear bicycle model combined with a nonlinear tire model. Inputs are either only the front-wheel-steering or an additional braking torque while actuator dynamics are considered. The trajectory generation is performed using a linear time-varying model predictive controller.

Falcone et al. (2007) considers a double track model assuming constant vertical tire forces combined with a nonlinear tire model. The front-wheel-steering and a braking torque are assumed as control inputs. The trajectories are generated using a linear time-varying model predictive controller. It aims to keep the distance to a reference path small and follow a velocity reference.

Gottmann et al. (2018) compares different models for a racing line generation. The models reach from lumped mass model with acceleration constraints to a nonlinear bicycle model. The benefits of having a rear-wheel-steering are pointed out.

The novelty of this paper is a model predictive trajectory planner for an over actuated vehicle to generate near time optimal trajectories up to its handling limits. The trajectories are generated using a validated nonlinear dynamic double track vehicle model in a nonlinear model predictive control (NMPC) framework. To the best knowledge of the authors, such a detailed model has not been used to generate a near time optimal trajectory. This paper, however, does not consider the controller to track the reference trajectory.

The remainder of this paper is structured as follows. Section 2 presents the considered model and shows some adaptions. In Section 3, the trajectory generation using an NMPC is introduced and the optimization problem is presented. Simulation results are pointed out in Section 4 and a conclusion is given in 5.

2. MODELING

In this section, the nonlinear vehicle model which is used in the NMPC framework to generate the trajectories is introduced. Firstly, the pose model in Frenet coordinates is given, which is based on Gottmann et al. (2018). Secondly,

Copyright lies with the authors

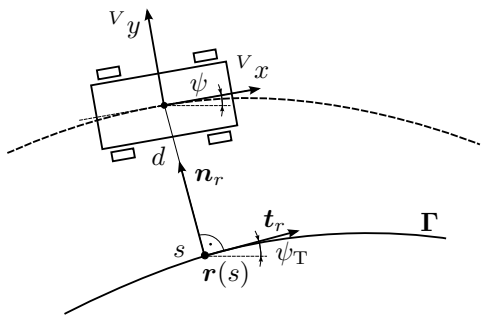

Fig. 1. Pose model of a vehicle based on Werling et al. (2010).

a reduced dynamic model for the vehicle with mainly five states is introduced. This model is derived and validated in Henning and Sawodny (2016). Finally, the model is slightly adapted for the usage in an NMPC.

2.1 Pose Model in Frenet Coordinates

A Frenet frame moves along a differentiable path. For the vehicle model, this is depicted in Fig. 1 which is based on Werling et al. (2010). The tangential vector $\boldsymbol{t}_r$ and the normal vector $\boldsymbol{n}_r$ form the orthonormal basis of the Frenet frame at the base point of the path $\boldsymbol{r}(s)$. The tangential vector has the orientation ψ_T and s represents the total distance travelled. The position and orientation of the vehicle with the frame $[^V x, ^V y]$ can be described with respect to the Frenet frame with the distance d and the orientation ψ. This formulation is chosen since the distance from the vehicle to the path and the orientation error e_ψ is explicitly available which can be considered in the cost function of the optimization problem for the trajectory generation. The kinematic model of the vehicle in a Frenet frame is given by Gottmann et al. (2018)

$$\begin{aligned} \dot{s} &= \frac{v_x}{1-\kappa_\mathrm{T} d}\cos(e_\psi) - \frac{v_y}{1-\kappa_\mathrm{T} d}\sin(e_\psi) \\ \dot{d} &= v_x \sin(e_\psi) + v_y \cos(e_\psi) \\ \dot{e}_\psi &= \dot{\psi} - \kappa_\mathrm{T} \cdot \dot{s} \\ s(0) &= s_0,\ d(0) = d_0,\ e_\psi(0) = e_{\psi,0} \end{aligned} \tag{1}$$

with the velocity in the x- and y direction v_x and v_y, the error between the path and the vehicle given by $e_\psi = \psi - \psi_\mathrm{T}$ and the initial conditions. The curvature κ_T is given by Burg (2012)

$$\kappa_\mathrm{T} = \frac{\mathrm{d}\psi_\mathrm{T}}{\mathrm{d}s}. \tag{2}$$

A further advantage by describing the pose of the vehicle in a Frenet frame is that the vehicle dynamics are described in a vehicle fixed coordinate system. The velocities in x- and y-direction and the yaw rate are calculated by the dynamic model and then entered in the pose model in (1).

2.2 Vehicles Dynamics

The vehicle dynamics are described in the body fixed coordinate system V depicted in Fig. 2. The superscript V indicating this coordinate system is mostly neglected in the following model. The actuators of the considered vehicle are the front-wheel-steering with the steering angle δ_f, the rear-wheel-steering δ_r, the driving torque M_D and

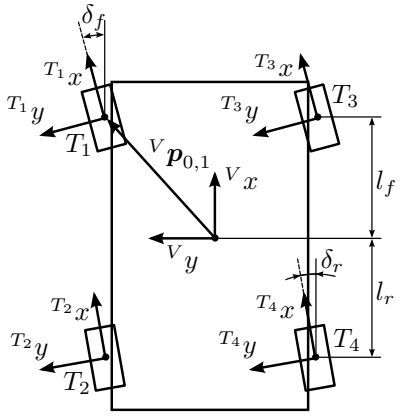

Fig. 2. Coordinate systems of a double track vehicle model.

the braking torque M_B which are applied on each wheel equally as well as the wheel individual brakes $\boldsymbol{M}_{\mathrm{B,add}}$. The states and the inputs of the vehicle considering the dynamics and the pose model are given by

$$\begin{aligned} \boldsymbol{x} &= [s, d, e_\psi, \dot{\psi}, v_x, v_y, a_x, a_y, \delta_f, \delta_r, M_\mathrm{D}]^T, \boldsymbol{x} \in \mathbb{R}^{11}, \\ \boldsymbol{u} &= [\dot{\delta}_\mathrm{V}, \dot{\delta}_\mathrm{H}, \dot{M}_\mathrm{D}, M_\mathrm{B}, M_{\mathrm{B,add},i}]^T, \boldsymbol{u} \in \mathbb{R}^8, i = 1,2,3,4. \end{aligned} \tag{3}$$

The first three states represent the pose model in Section 2.1. The next five states $[\dot{\psi}, v_x, v_y, a_x, a_y]$ are used to describe the vehicle dynamics. The dynamics of the steering angles and the driving torque are approximated with the last three states which are the integral from the first three inputs. This model of the vehicle is over actuated since it has more inputs than degrees of freedom.

The derivation of the differential equation for the five states representing the vehicle dynamics is presented in Henning and Sawodny (2016). For the velocity and the yaw rate, these are given by

$$\begin{aligned} \dot{v}_x &= \dot{\psi} \cdot v_y + \frac{^V F_{x,\mathrm{Chassis}}}{m} \\ \dot{v}_y &= -\dot{\psi} \cdot v_x + \frac{^V F_{y,\mathrm{Chassis}}}{m} \\ \ddot{\psi} &= \frac{1}{J_z}\left(\sum_{i=1}^{4} \begin{pmatrix} ^V p_{x,0,i} \\ ^V p_{y,0,i} \\ 0 \end{pmatrix} \times \begin{pmatrix} ^V F_{x,i} \\ ^V F_{y,i} \\ 0 \end{pmatrix}\right) \\ v_x(0) &= v_{x,0},\ v_y(0) = v_{y,0},\ \dot{\psi}(0) = \dot{\psi}_0, \end{aligned} \tag{4}$$

with the mass of the vehicle m and the vector $^V \boldsymbol{p}_{0,i}$ from the frame V to the tire i as depicted in Fig. 2. The forces $^V F_{x,i}$ and $^V F_{y,i}$ are the transformed forces from the tire coordinate system T_i into the body fixed coordinate system V generated by the tire i in x- and y direction. The forces with the additional subscript *Chassis* include furthermore the drag forces. The initial conditions are labeled with the subscript 0. The model requires the accelerations in x- and y-direction for the calculation of the vertical tire force, which are approximated by

$$\begin{aligned} \dot{a}_x &= \frac{1}{T_{a_x}}\left(\frac{F_{x,\mathrm{Chassis}}}{m} - a_x\right) \\ \dot{a}_y &= \frac{1}{T_{a_y}}\left(\frac{F_{y,\mathrm{Chassis}}}{m} - a_y\right) \\ a_x(0) &= a_{x,0},\ a_y(0) = a_{y,0}, \end{aligned} \tag{5}$$

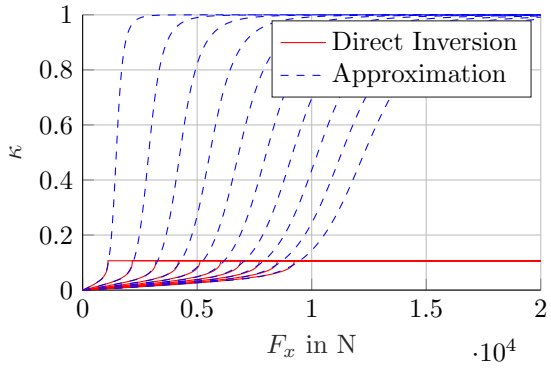

Fig. 3. The approximated slip $\kappa_{i,\text{approx}}$ and the slip calculated by directly inverting the simplified MFT. The different representations of this function have different vertical tire forces as a basis.

with the time constants T_a according to Henning and Sawodny (2016).

2.3 Adaption for the Usage in the NMPC Framework

Some equations for the calculation of (4) contain discontinuities. This behavior is undesirable for the NMPC since the chosen optimizer requires a valid gradient of the system dynamics to find the optimal solution. Therefore, these equations are adapted for this usage.

One equation concerns the calculation of the generated force by the tires. Using an approximated Magic Formula Tire (MFT) model as presented in Orend (2005) and Pacejka (2012), the tire force in x- and y-direction resolved in the tire frame is given by

$$\begin{aligned} {}^{T_i}F_{x,i} &= k_{\mu_x} F_{\max,i} \sin\left(C \arctan\left(\frac{B}{\mu} s_{\text{ges},i}\right)\right) \frac{s_x}{s_{\text{ges},i}} \\ {}^{T_i}F_{y,i} &= -F_{\max,i} \sin\left(C \arctan\left(\frac{B}{\mu} s_{\text{ges},i}\right)\right) \frac{s_y}{s_{\text{ges},i}} \end{aligned} \tag{6}$$

with the friction coefficient μ, the tire parameters C, B and k_{μ_x}, the maximal force of the tire $F_{\max}$ depending on the vertical tire force and the scaled slip in x-direction $s_x = k_\kappa \kappa$ and in y-direction $s_y = \alpha$ according to Henning and Sawodny (2016). The calculation of the slip angle α is given by Pacejka (2006, 2012). The combined slip results in

$$s_{\text{ges},i} = \sqrt{s_{x,i}^2 + s_{y,i}^2} + s_{\text{add}}, \tag{7}$$

with a small added slip s_{add} to avoid singularities (Henning and Sawodny (2016)). Since the wheel speeds are not modeled as an explicit state, the slip cannot directly be calculated. In Henning and Sawodny (2016), a workaround using the inverse of (6) is suggested. This results in

$$\kappa_i = \frac{\mu}{k_\kappa \cdot B} \tan\left(\frac{1}{C} \arcsin\left(\frac{F_{\text{ges},i}}{F_{\max,i}}\right)\right) \frac{1}{F_{\text{ges},i}} \frac{F_{x,i}}{k_{\mu_x}} \tag{8}$$

with

$$F_{\text{ges},i} = \sqrt{\left(\frac{F_{x,i}}{k_{\mu_x}}\right)^2 + F_{y,i}^2}. \tag{9}$$

The required force in x-direction is approximated using the torque at each wheel in combination with the radius as suggested in Henning and Sawodny (2016).

The function calculating the slip in (8) is saturated as depicted in Fig. 3 and therefore shows a discontinuity. To avoid this, the function is approximated by

$$\kappa_{i,\text{approx}} = \frac{2}{\pi} \arctan\left(\mu \cdot \left(a_1 \xi_i + a_3 \xi_i^3 + a_5 \xi_i^5 + a_7 \xi_i^7\right)\right), \tag{10}$$

with $\xi_i = \frac{F_{\text{ges},i}}{F_{\max,i}}$ according to Gottmann et al. (2018). The parameters a_j $j = 1, 3, 5, 7$ are identified by minimizing the squared error between (8) and (10). Fig. 3 shows the direct inversion and the approximated function dependent on different vertical tire forces.

Furthermore, the vertical tire forces needed to calculate the horizontal tire forces in (6) can become negative under certain situations. Since this is not possible in real world, the function is saturated in Henning and Sawodny (2016). To avoid this discontinuity, the vertical tire force is approximated by

$$F_{z,\text{blend},i} = \left(\left(\frac{1}{\pi} \arctan(F_{z,i}) + \frac{1}{2}\right) \cdot F_{z,i} + 0.35\right). \tag{11}$$

For positive values of F_z, the function is linear apart from close to zero. For negative values, the function is continuously decreasing and not reaching zero.

3. TRAJECTORY GENERATION

In this section, the trajectory generation via the nonlinear model predictive control is presented. The aim is to generate a near time optimal trajectory considering constraints of the system. Firstly, the general idea is introduced. Secondly, the inputs and states are scaled for the usage within the NMPC framework. Finally, the optimization problem is shown.

3.1 Background

A MPC solves an open-loop optimal control problem with a finite horizon in every sample time and applies the first control to the system. One advantage of this method is that constraints on inputs and states can be considered (Allgöwer et al. (1999); Mayne et al. (2000)). For the purpose of generating a trajectory, the simulated state is used as an initial condition instead of the measured state as presented in Van den Broeck et al. (2009) and successfully applied in industry in Richter et al. (2014) and Schaper et al. (2013).

Fig. 4 shows the used structure for generating the trajectories using an NMPC. The NMPC uses the same model for the optimization and for the prediction of the states. This has the advantage that the full state is available without any disturbances. The optimal input trajectory can be used as an feedforward part for the system and the output trajectory can be used with a subsequent controller, as suggested in Gottmann et al. (2017), to regard for disturbances. Since this paper deals with generating the trajectory, the controller is not considered here.

3.2 Scaling of Inputs and States

The maximum values of the states and inputs vary greatly among each other. For a better numerical conditioning, the

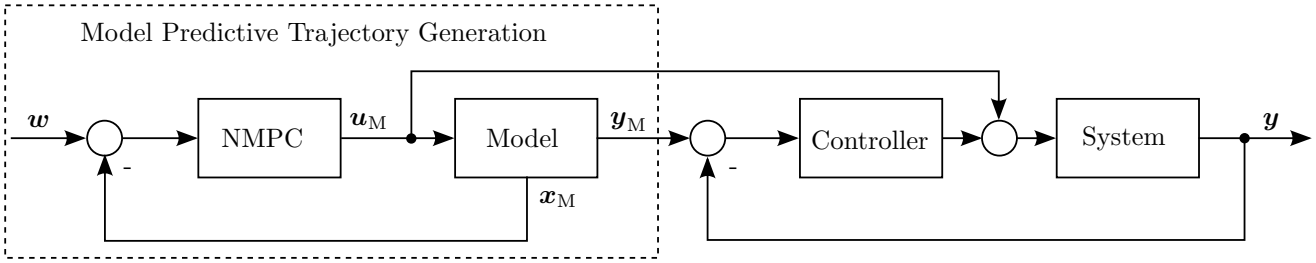

Fig. 4. Trajectory generation using an NMPC. The NMPC uses the same model for the optimization and the prediction and therefore has the full state available. Reference values can be the inputs and the outputs of the model.

inputs and states are scaled. This transformation is given by

$$\begin{aligned} \tilde{\boldsymbol{u}} &= \boldsymbol{N}_u \cdot \boldsymbol{u} \quad & \boldsymbol{u} &= \boldsymbol{N}_u^{-1} \cdot \tilde{\boldsymbol{u}}, \\ \tilde{\boldsymbol{x}} &= \boldsymbol{N}_x \cdot \boldsymbol{x} \quad & \boldsymbol{x} &= \boldsymbol{N}_x^{-1} \cdot \tilde{\boldsymbol{x}}, \end{aligned} \tag{12}$$

where the tilde operator indicates the scaled input and state. The transformation matrix results in

$$\begin{aligned} \boldsymbol{N}_x &= \operatorname{diag}\left(\frac{1}{x_{\max,1}}, \frac{1}{x_{\max,2}}, \ldots, \frac{1}{x_{\max,11}}\right), \\ \boldsymbol{N}_u &= \operatorname{diag}\left(\frac{1}{u_{\max,1}}, \frac{1}{u_{\max,2}}, \ldots, \frac{1}{u_{\max,8}}\right), \end{aligned} \tag{13}$$

with the maximum values for the states $x_{\max}$ and for the input $u_{\max}$. Considering the transformation in (12), the scaled differential equation results in

$$\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{N}_x \cdot \boldsymbol{f}(\boldsymbol{N}_x^{-1} \cdot \tilde{\boldsymbol{x}}, \boldsymbol{N}_u^{-1} \cdot \tilde{\boldsymbol{u}}) = \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}}). \tag{14}$$

3.3 Optimization Problem

For the generation of a near time optimal trajectory, the pose model in (1) is considered. If the distance traveled s is maximized, the time is implicitly minimized and thus near time optimal trajectories are generated. Since the problem cannot be solved analytically, a nonlinear optimization which can consider constraints is used to maximize the distance traveled. An optimization algorithm usually minimizes a cost function Therefore, the maximization of the scaled distance traveled is reformulated to

$$-\min\{-\tilde{s}\} = \max\{\tilde{s}\}. \tag{15}$$

The optimization problem is given by

$$\begin{aligned} \min_{\{\tilde{\boldsymbol{x}}_{k+1}, \tilde{\boldsymbol{u}}_k\}_{k=0}^{N-1}} & \sum_{k=0}^{N-1} \left(\tilde{\boldsymbol{x}}_k^T \hat{\boldsymbol{Q}}_k \tilde{\boldsymbol{x}}_k - q_{1,k} \cdot \tilde{s}_k + \tilde{\boldsymbol{u}}_k^T \boldsymbol{R}_k \tilde{\boldsymbol{u}}_k \right) \\ & + \tilde{\boldsymbol{x}}_N^T \hat{\boldsymbol{Q}}_N \tilde{\boldsymbol{x}}_N - q_{1,N} \cdot \tilde{s}_N \\ \text{s.t.} \quad & \tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{f}_k(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{u}}_k) \quad k = 0, (1), N-1 \\ & \tilde{\boldsymbol{x}}_0 = \tilde{\boldsymbol{x}}_I \\ & \begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix} \leq \begin{bmatrix} \tilde{\delta}_{\mathrm{V},k} \\ \tilde{\delta}_{\mathrm{H},k} \\ \tilde{M}_{\mathrm{A},k} \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad k = 0, (1), N \\ & \begin{bmatrix} -1 \\ -1 \\ -1 \\ 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} \dot{\tilde{\delta}}_{\mathrm{V},k} \\ \dot{\tilde{\delta}}_{\mathrm{H},k} \\ \dot{\tilde{M}}_{\mathrm{A},k} \\ \dot{\tilde{M}}_{\mathrm{B},k} \\ \dot{\tilde{M}}_{\mathrm{B,add},k,i} \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{array}{l} k = 0, (1), N-1 \\ i = 1, 2, 3, 4. \end{array} \end{aligned} \tag{16}$$

with the initial condition $\tilde{\boldsymbol{x}}_I$ and the weighting matrices for the states and the inputs

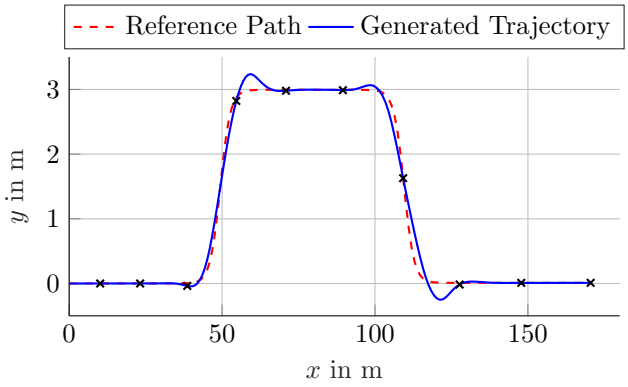

Fig. 5. The reference path and the generated trajectory for the double lane change. The time distance between two crosses is one second.

$$\begin{aligned} \hat{\boldsymbol{Q}}_k &= \hat{\boldsymbol{Q}}_N = \operatorname{diag}(0, q_2, q_3, \ldots, q_{11}) \\ \hat{\boldsymbol{R}}_k &= \hat{\boldsymbol{Q}}_N = \operatorname{diag}(r_1, r_2, r_3, \ldots, r_8). \end{aligned} \tag{17}$$

The weighting coefficient q_1 is not included in the matrix $\hat{\boldsymbol{Q}}$ which penalizes the states quadratically. It weights the negative scaled distance traveled in a linear way to maximize $\tilde{s}$. With the coefficients q_2 and q_3 the distance d and the orientation error e_ψ are penalized. The resulting trajectory greatly depends on these coefficients. The other coefficients are optimized to achieve a better convergence time of the optimization problem. The constraints in (16) consider actuator constraints. For the steering angles and the driving torque, also the gradient is limited.

The discretization time is chosen to 50 ms and the horizon length N is 30 which results in a time horizon of the optimization problem to 1.5 s. The optimization problem is solved using a multi-stage quadratic and nonlinear program (MSQNLP) as presented in Sonntag (2014).

4. SIMULATION RESULTS

This section shows the simulation results of the near time optimal trajectory generation using an NMPC. The test scenario is a double lane change. Firstly, the resulting trajectory for the position of the vehicle and for the states and inputs are shown. Secondly, the weighting coefficients of the cost function are changed and the affects are discussed.

4.1 Double Lane Change

The double lane change is a risky driving task as discussed in Arefnezhad et al. (2018) and therefore used as a refer-

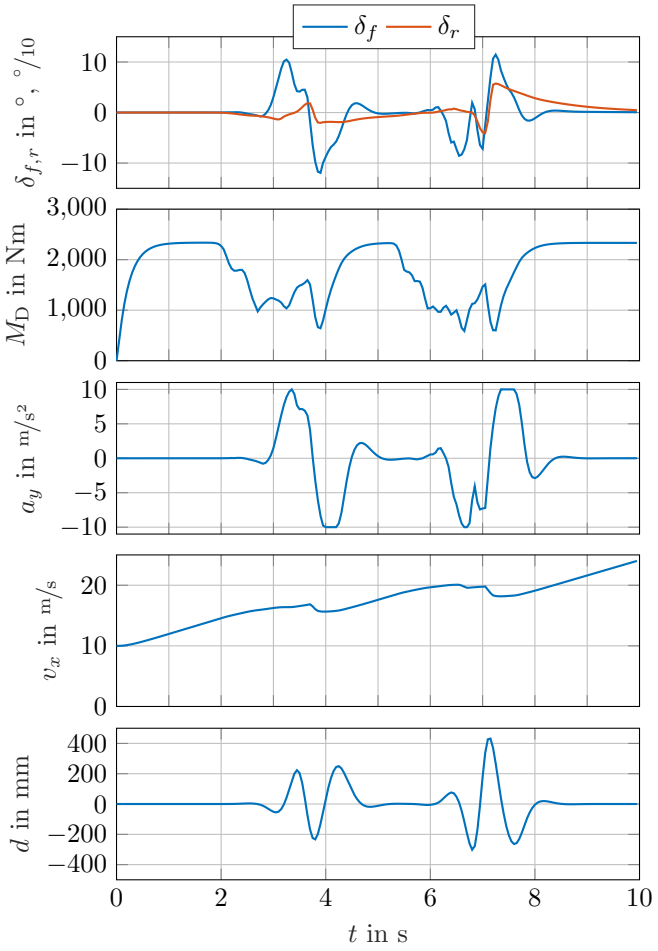

Fig. 6. Simulation results of the steering angles, the driving torque, the lateral acceleration, the longitudinal velocity and the distance from the vehicle to the reference path.

ence path for this simulation case. Fig. 5 shows the reference path for the double lane change and the generated trajectory while the time distance between two crosses is one second. Fig. 6 shows some states and Fig. 7 shows the braking torque of this simulation case. Considering the driving torque, it can be seen that it mostly accelerates at the limits apart from the transition to the other lane. This results in a mostly increasing longitudinal velocity v_x. Only when the braking torque M_B is acting, the velocity is reduced. Considering the lateral acceleration a_y, the physical limits of $10\,\mathrm{m/s^2}$ are reached while changing the lane. The steering angle of the rear axis and front axis are mostly in the same direction which contribute to stabilize the vehicle. The wheel individual braking torque are used to increase the yawing moment and therefore to reduce the error in the orientation. The distance d shows lower values for changing to the left lane than changing back to the right lane. Fig. 5 illustrates this behavior. The higher distance is necessary to reduce the path curvature due to the increased velocity.

4.2 Variation of the Weighting Coefficients

To show the ability of the trajectory generation using the NMPC, the weighting coefficients in the cost function are changed. The simulation labeled with V1 is the one

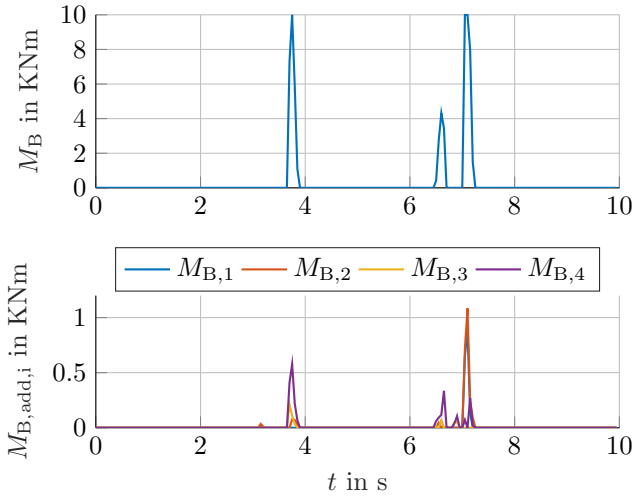

Fig. 7. Simulation results for the braking torque and the individual braking torques.

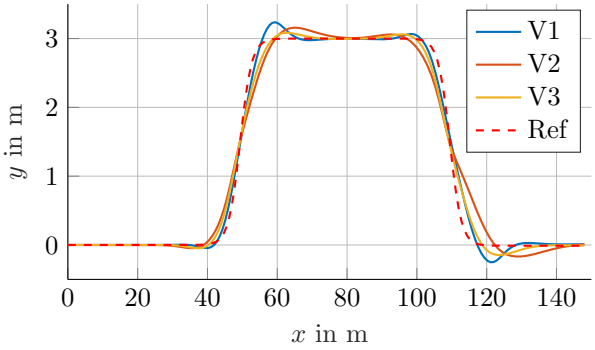

Fig. 8. Generated and reference path for the three different simulation cases. V1 is presented in Section 4.1, V2 uses $\frac{q_2}{10}$ and V3 uses $\frac{q_2}{10}$ and $\frac{q_1}{2}$.

Table 1. Distance traveled at $t = 10\,\mathrm{s}$ for the three simulation cases.

	V1	V2	V3
$s(t = 10\ \mathrm{s})$ in m	172,3828	208,9751	151,8332

presented in Section 4.1. The second case reduces the weighing coefficient q_2 for the distance d by a factor of 1/10. The third case also reduces the coefficient q_2 by 1/10 and further reduces the weighing coefficient q_1 by 1/2. Fig. 8 shows the generated path for the three cases and the reference path. Fig. 9 shows the lateral acceleration and the distance d over the distance traveled s. Considering V2, decreasing q_2 results in higher deviation from the reference path and therefore higher distances d. For V3, decreasing q_2 and q_1 results in slightly higher deviations from the path but reduces the lateral accelerations greatly. Also V2 shows lower lateral accelerations since it can further reduce the path curvature by increasing the distance d.

Table 1 shows the distance traveled at $t = 10\,\mathrm{s}$. Considering the distance traveled at $t = 10\,\mathrm{s}$, the case V1 shows lower values than V2 since the higher weighting of the distance d results in higher path curvature and therefore lower velocity v_x. The distance traveled is the lowest for case V3 since the weighting q_1 is reduced.

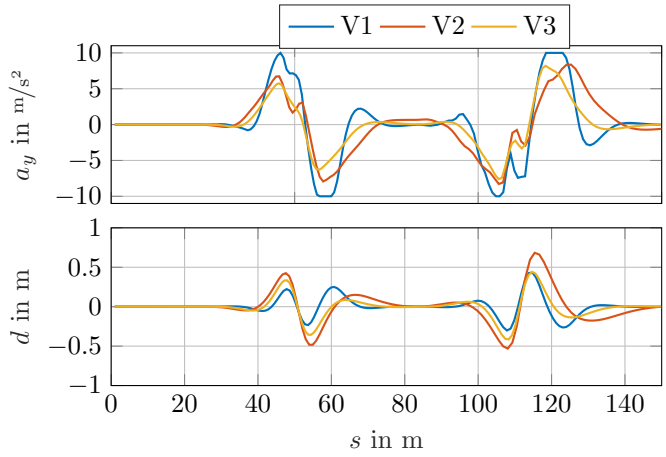

Fig. 9. Lateral acceleration and distance for the three different simulation cases. V1 is presented in Section 4.1, V2 uses $\frac{q_2}{10}$ and V3 uses $\frac{q_2}{10}$ and $\frac{q_1}{2}$.

5. CONCLUSION

In this paper, a nonlinear model predictive controller is used to generate near time optimal trajectories for an over actuated vehicle. The pose of the vehicle is modeled in a Frenet frame. For the vehicle dynamics, a validated nonlinear double track model with three degrees of freedom is used. Since some equations show discontinuity, adaptions for the usage withing the NMPC framework are carried out. To increase the numerical conditioning, the states and inputs are scaled. The simulation results for a double lane change show trajectories which follow the reference path with some deviations. The actuator constraints are considered and the trajectories are generated up to the dynamic limits of the vehicle. A variation of the weighting coefficients show the ability of generating trajectories with different focus.

REFERENCES

Allgöwer, F., Badgwell, T.A., Qin, J.S., Rawlings, J.B., and Wright, S.J. (1999). Nonlinear predictive control and moving horizon estimation—an introductory overview. In *Advances in control*, 391–449. Springer.

Arefnezhad, S., Ghaffari, A., Khodayari, A., and Nosoudi, S. (2018). Modeling of double lane change maneuver of vehicles. *International Journal of Automotive Technology*, 19(2), 271–279.

Attia, R., Orjuela, R., and Basset, M. (2014). Combined longitudinal and lateral control for automated vehicle guidance. *Vehicle System Dynamics*, 52(2), 261–279.

Burg, K. (2012). *Vektoranalysis*. Höhere Mathematik für Ingenieure, Naturwissenschaftler und Mathematiker. Vieweg+Teubner Verlag, Wiesbaden.

Falcone, P., Borrelli, F., Asgari, J., Tseng, H., Eric, and Hrovat, D. (2007). A model predictive control approach for combined braking and steering in autonomous vehicles. In *Control & Automation, 2007. MED '07. Mediterranean Conference on*, 1–6.

Gottmann, F., Böhm, M., and Sawodny, O. (2017). Integrated trajectory and path tracking of under-actuated and over-actuated cars up to the handling limits. In *2017 American Control Conference (ACC)*, 954–960. IEEE.

Gottmann, F., Wind, H., and Sawodny, O. (2018). On the influence of rear axle steering and modeling depth on a model based racing line generation for autonomous racing. In *2018 IEEE Conference on Control Technology and Applications (CCTA)*, 846–852. IEEE.

Gutjahr, B., Gröll, L., and Werling, M. (2016). Lateral vehicle trajectory optimization using constrained linear time-varying mpc. *IEEE Transactions on Intelligent Transportation Systems*, 18(6), 1586–1595.

Heißing, B. (2013). *Fahrwerkhandbuch*. Grundlagen, Fahrdynamik, Komponenten, Systeme, Mechatronik, Perspektiven. Springer Vieweg, Wiesbaden.

Henning, K.U. and Sawodny, O. (2016). Vehicle dynamics modelling and validation for online applications and controller synthesis. *Mechatronics*, 39, 113–126.

Katriniok, A. and Abel, D. (2011). 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*, 6828–6833.

Katriniok, A., Maschuw, J.P., Christen, F., Eckstein, L., and Abel, D. (2013). Optimal vehicle dynamics control for combined longitudinal and lateral autonomous vehicle guidance. In *Control Conference (ECC), 2013 European*, 974–979.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.

Orend, R. (2005). Steuerung der ebenen fahrzeugbewegung mit optimaler nutzung der kraftschlusspotentiale aller vier reifen. *at–Automatisierungstechnik/Methoden und Anwendungen der Steuerungs-, Regelungs-und Informationstechnik*, 53(1/2005), 20–27.

Pacejka, H.B. (2006). *Tyre and vehicle dynamics*. Elsevier Butterworth-Heinemann, Amsterdam and Heidelberg [u.a.].

Pacejka, H.B. (2012). *Tire and vehicle dynamics*. Elsevier, Amsterdam and Heidelberg [u.a.].

Richter, M., Arnold, E., Schneider, K., Eberharter, J.K., and Sawodny, O. (2014). Model predictive trajectory planning with fallback-strategy for an active heave compensation system. In *2014 American Control Conference*, 1919–1924. IEEE.

Schaper, U., Arnold, E., Sawodny, O., and Schneider, K. (2013). Constrained real-time model-predictive reference trajectory planning for rotary cranes. In *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 680–685. IEEE.

Sonntag, M. (2014). Multi-stage quadratic and nonlinear programming (msqnlp). `https://www.isys.uni-stuttgart.de/lehre/lehrveranstaltungen/nmopt/msqnlp/index.html`. Accessed: 2019-10-29.

Van den Broeck, L., Diehl, M., and Swevers, J. (2009). Performant design of an input shaping prefilter via embedded optimization. In *2009 American Control Conference*, 166–171. IEEE.

Werling, M. (2011). *Ein neues Konzept für die Trajektoriengenerierung und -stabilisierung in zeitkritischen Verkehrsszenarien*. KIT Scientific Publishing, Karlsruhe.

Werling, M., Groll, L., and Bretthauer, G. (2010). Invariant trajectory tracking with a full-size autonomous road vehicle. *IEEE Transactions on Robotics*, 26(4), 758–765.