2021 European Control Conference (ECC)
June 29 - July 2, 2021. Rotterdam, Netherlands

Parameterization Approach of the Frenet Transformation for Model Predictive Control of Autonomous Vehicles

Rudolf Reiter[1] and Moritz Diehl[2]

***Abstract*— Model predictive control (MPC) and nonlinear optimization-based planning for autonomous vehicles are often formulated in a transformed coordinate frame, namely the curvilinear Frenet frame. Mostly the center line of the road is used as a transformation curve, but the choice of the transformation curve might have properties which make the optimization problem hard or even infeasible to solve in the whole search space. This paper proposes an optimization-based parameterization approach to establish an alternative transformation curve which yields favorable numerical properties for the consecutive use of numerical optimization approaches such as MPC. The optimization objective minimizes the change of curvature and pushes the evolute (i.e. singular region) of the transformation curve outside the feasible region. The convergence improvement of the proposed parameterization approach in terms of integrator precision, optimization time and iteration counts is compared in simulation examples, using a time-optimal nonlinear optimization formulation.**

I. INTRODUCTION

In the last decade nonlinear optimization-based approaches for both, trajectory planning and control of autonomous vehicles have been investigated actively in scientific research and real-world applications ([1], [2], [3]). Nonlinear optimization helps to either perform control tasks which respect nonlinearities and constraints or plan optimal trajectories or paths for motion control systems. Many of the existing approaches use a particular state transformation, which maps the Cartesian states to a road aligned path, formulated as a curve in the Cartesian frame. The transformed frame is referred to as "Frenet frame" and used for example in [4], [5], [6], [7], [8]. This leads to favorable properties of the resulting system model regarding optimization objectives and the structure of the resulting nonlinear program (NLP). The formulation of maximum path progress with a constant time horizon, which corresponds to time-optimal planning is straight-forward by maximizing over the system state of path progress and the tracking of the transformation curve can be achieved easily by minimizing over the lateral distance state. Also the motion of other traffic participants is typically aligned with the road, which can be integrated easily into the constraints. The choice of the road center line as a transformation curve comes naturally since often the reference curve and the road boundaries are parallel to the center line. Other traffic participants also often move along center lane aligned trajectories. This usual choice of the transformation curve is shown in Fig. 1. Due to these

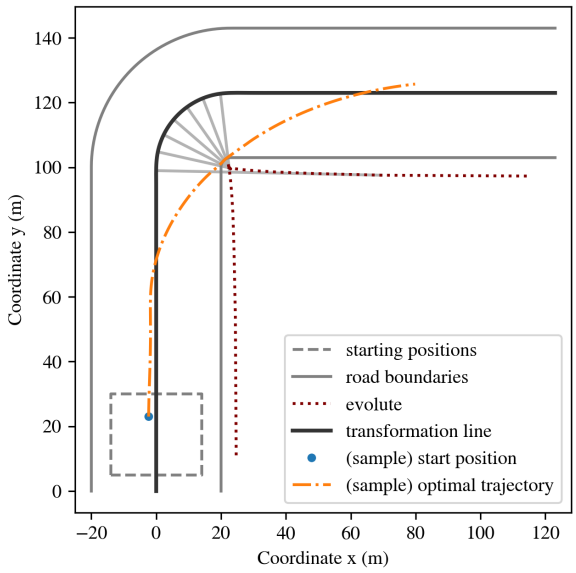

Fig. 1. Center line transformation.

advantages, the center line is widely taken as the transformation to the road geometry and restrictions are put on the road geometry itself to make this transformation unique and free of singularities [4], [8]. In contrast to academic cases, real world scenarios come with several differences, which make the choice of the transformation curve a design parameter or even require a necessary adaption, so that the whole road space is feasible for the system states. So far no work has considered this transformation as a design parameter, it was rather taken as given input. Given a point on a curve, an osculating circle describes a circle that has the same tangent as the curve in this point and which has the same curvature (Fig. 7 shows the osculating circle in point p_i). The curve, describing the evolution of the center of the osculating circles is referred to as evolute. As a hard constraint for the choice of the transformation curve, the road boundary perpendicular to the transformation curve must be closer than its oriented radius of all osculating circles of the planar transformation curve. Since an NLP is an approximation of the optimal control problem (OCP) and approximated with a discretization in time, the distance of the road boundary to the evolute must even be raised by a certain factor to achieve numerical robustness. By transforming the vehicle model into

[1]Rudolf Reiter is with the Virtual Vehicle Research Center, Inffeldgasse 21a, 8010 Graz, Austria and the University Freiburg, 79110 Freiburg, Germany. rudolf.reiter@imtek.uni-freiburg.de

[2]Moritz Diehl is with the Department of Microsystems Engineering (IMTEK) and the Department of Mathematics, University Freiburg, 79110 Freiburg, Germany moritz.diehl@imtek.uni-freiburg.de

the Frenet coordinates, the curvature becomes part of the dynamic system. Consequently, also the nonlinearity of the curvature directly becomes part of the vehicle model. Besides of the hard constraint regarding the singularity emerging from the curvature, this work also addresses further favorable properties of the transformation choice. The dynamic state equations (including the curvature) enter the NLP by equality constraints and at least one derivative is used by the solver. Good convergence properties are achieved if higher order derivatives of the NLP constraints can be lowered and this is performed by the presented parameterization of the transformation curve. An example of the parameterized curve is shown in Fig. 2. For known tracks (e.g. race tracks, known road networks) the transformation can be computed offline in advance.

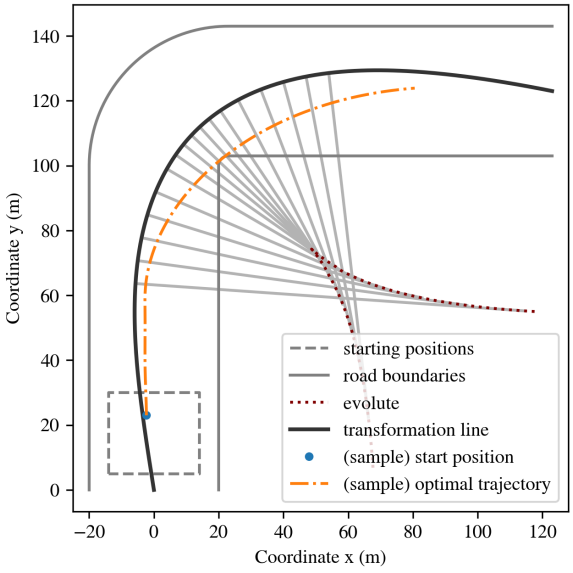

Fig. 2. Suggested transformation with objective as in (13).

II. SYSTEM MODEL

A. Single Track Model

A kinematic model in a curvilinear reference frame is used, which was presented first in [9] and leads to a slip-free tire model. The direction of the movement of the center of gravity (CG) with the vehicle mass m is given by the angle $\psi + \beta$, where ψ is the vehicle orientation. The side-slip angle β is defined as depicted in Fig. 3 and gives the relative angle of motion related to the vehicle coordinate system. The side-slip angle is given by

$$\beta = \arctan\left(\frac{l_\mathrm{r}}{l_\mathrm{r}+l_\mathrm{f}}\tan\delta\right). \tag{1}$$

The system model can then be described by the following Equations (2) in the Cartesian coordinate frame. The velocity vector v denotes the velocity related to the CG.

$$\dot{p}_\mathrm{X} = v\cos(\psi+\beta) \tag{2a}$$
$$\dot{p}_\mathrm{Y} = v\sin(\psi+\beta) \tag{2b}$$
$$\dot{\psi} = \frac{v}{l_\mathrm{r}}\sin\beta \tag{2c}$$
$$\dot{v} = \frac{F_\mathrm{x}^\mathrm{d}}{m}\cos\beta \tag{2d}$$

The geometry of the vehicle is simply described by the longitudinal position of the CG with the front distance l_f and the rear distance l_r. The input force F_x^d only acts on the rear wheel, where as the steering angle δ only deflects the front wheel, which is an arbitrary choice and does not influence the proposed algorithm.

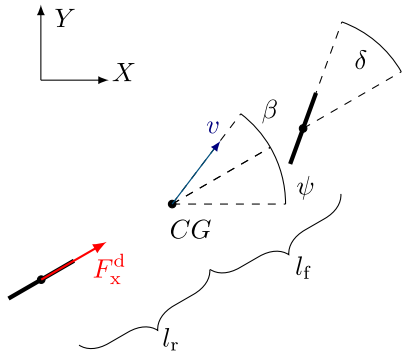

Fig. 3. Kinematic single-track model.

B. Curvilinear Transformation

So far the system model is independent of any road geometry, but as pointed out, a useful transformation leads to the vehicle system equations in the Frenet frame. It leads to the dynamic system (3) with the states $x = \begin{bmatrix} s, & n, & \alpha, & v \end{bmatrix}^T$ and controls $u = \begin{bmatrix} F_\mathrm{x}^\mathrm{d}, & \delta \end{bmatrix}^T$, which now depend on the curvature. Here, path aligned states are used which describe the progress on the transformation path $s(t)$, the normal distance to the transformation path $n(t)$ and the heading angle mismatch $\alpha(s,t) = \psi(t) - \psi^\mathrm{c}(s)$. In many works (e.g. [8], [4]) this transformation is performed along the center line, which is generally not the case here.

$$\dot{s} = \frac{v\cos(\alpha+\beta)}{1-n\kappa(s)} := f_s(x) \tag{3a}$$
$$\dot{n} = v\sin(\alpha+\beta) \tag{3b}$$
$$\dot{\alpha} = \dot{\psi} - \kappa(s)\dot{s} := f_\alpha(x) \tag{3c}$$
$$\dot{v} = \frac{F_\mathrm{x}^\mathrm{d}}{m}\cos(\beta) \tag{3d}$$

The equations can be summarized by the nonlinear dynamic system equations of first order.

$$\dot{x} = f(x,u) \tag{4}$$

Note that the curvature $\kappa(s)$ together with bounds on the normal distance state n now fully describe the road geometry. Fig. 4 shows the transformation of a point to the curve $\gamma(s)$,

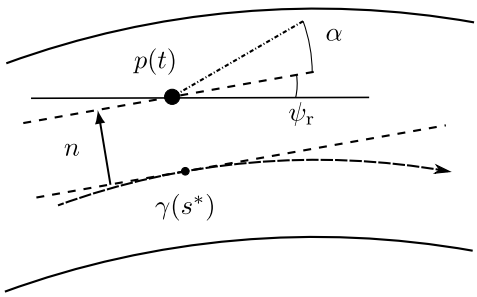

Fig. 4. Path-parametric model as in [4].

normal distance n, the error angle α and the heading angle of the reference ψ_r.

III. NEWTON-TYPE OPTIMIZATION

As described in [4], the time-optimal racing problem can be described very generally by the following multiple shooting NLP and allows the usage of a Gauß-Newton Hessian approximation but might also be solved with an exact Hessian ([10]). It is important to emphasize that the presented approach shifts the reference line, which might not be desired for certain applications. For those applications, a reference offset could be used but is out of the scope of this work. The time dependence of the states x and controls u is discretized with fixed time intervals Δt and an integration scheme $F(x_k, u_k, \Delta t)$. Equation (5e) puts constraints on the states, which can depend on the path variable s as well as on the time index k, to account for time varying constraints like moving obstacles.

$$\min_{\substack{x_0,\dots,x_N,\\ u_0,\dots,u_{N-1}}} \quad \sum_{k=0}^{N-1} \|x_k - x_{k,\mathrm{ref}}\|_Q^2 + \|u_k\|_R^2 + \|x_N - x_{N,\mathrm{ref}}\|_{Q_\mathrm{N}}^2 \tag{5a}$$

$$\text{s.t.} \qquad x_0 = x_\mathrm{c}, \tag{5b}$$

$$x_{k+1} = F(x_k, u_k, \Delta t), \quad k = 0, \dots, N-1, \tag{5c}$$

$$\underline{u} \le u_k \le \overline{u}, \quad k = 0, \dots, N-1, \tag{5d}$$

$$\underline{x}_k(s_k) \le x_k \le \overline{x}_k(s_k), \quad k = 0, \dots, N-1, \tag{5e}$$

Newton-type algorithms use first order (Gauß-Newton Hessian) or second order derivatives (exact Hessian) of the constraints, which are particularly interesting for (5c) and (5e), since both equations are influenced by the chosen representation of the road geometry $\kappa(s)$. Since (5e) is in general driving scenarios not exactly known a priori, it can just be assumed that obstacles move and their geometry is aligned along the center line. Nevertheless the model integration (5c) is of fixed structure which can be exploited in order to parameterize the transformation of the road geometry, resulting in $\kappa(s)$. By taking a closer look at the system dynamics (3a), the function value is heading towards infinity if the normal distance of the transformation line n is close to the reciprocal value of $\kappa(s)$, which corresponds to the radius of curvature. The curvature ratio $\rho(s, n)$ can be defined as the ratio of the lateral distance n to the radius of the osculating circle $R(s) = \frac{1}{\kappa(s)}$, which yields $\rho(s, n) = \frac{n}{R(s)} = n\kappa(s)$.

IV. SINGULARITY AND SMOOTHNESS PROBLEM

By symbolically computing the first (6) and second (7) partial derivatives of the state s, it is further obvious that the denominators have a quadratic dependence on the non-linearity described above, which make the resulting Jacobian of the constraints arbitrarily ill-conditioned if the curvature ratio $\rho(s, n)$ is sufficiently close to 1. Equation (6) shows that higher order derivatives for the differential equations of the integrator in (5c) can be lowered by reducing the maximum possible value for the denominator, which is the main subject of the presented parameterization approach. According to [11], it can generally be assumed that lowering higher order derivatives increases convergence properties of usually applied numerical solvers.

$$\frac{\partial f_s(x)}{\partial s} = \frac{v\cos(\alpha+\beta)n\kappa(s)'}{(1-n\kappa(s))^2} \tag{6a}$$

$$\frac{\partial f_\alpha(x)}{\partial s} = -(\kappa(s)' f_s(x) + \kappa(s)\frac{\partial f_s(x)}{\partial s}) \tag{6b}$$

$$\frac{\partial^2 f_s(x)}{\partial s^2} = \frac{v\cos(\alpha+\beta)n((1-n\kappa(s))\kappa(s)'' + 2n(\kappa(s)')^2)}{(1-n\kappa(s))^3} \tag{7a}$$

$$\frac{\partial^2 f_\alpha(x)}{\partial s^2} = -\kappa(s)'' f_s(x) - 2\kappa(s)'\frac{\partial f_s(x)}{\partial s} - \kappa(s)\frac{\partial^2 f_s(x)}{\partial s^2} \tag{7b}$$

Also the partial derivative $\kappa(s)'$ should stay small to reduce higher order derivatives.

V. OPTIMAL CURVILINEAR PARAMETERIZATION

Using the previously defined weaknesses of the center line transformation approach (i.e. mainly the singularities close or inside the feasible region) as costs, a novel parameterization approach is presented, which is formulated as an NLP. By means of discrete geometric considerations the singularity is pushed outside the feasible region and the curvature along the path length variable is smoothed while the distance to the center line is kept as close as possible. The solution transformation not only guarantees that no singularities are located inside the feasible region but also pushes these very nonlinear regions outside the feasible region as far as possible. Here, the waypoints $o_i \in \mathbb{R}^2$ describe N discrete points of a given piece-wise linear reference curve $\Gamma_o(s)$ in the Cartesian frame, depending on the path length s. The vector $v_i \in \mathbb{R}^2$ represents the tangent norm vector, with $\|v_i\| = 1$ pointing "left" facing the positive road direction, as shown in Fig. 5. The scalar value $t_i \in \mathbb{R}$ is used as the i-th element of the optimization variables $t = [t_0, ..., t_N]^T \in \mathbb{R}^N$ and shifts the reference point towards the road boundaries, resulting in the shifted point p_i (Fig. 6). The road borders are given as maximum or minimum deflection t_i of point p_i into direction v_i, which are denoted by $t_i^{\max}$ or $t_i^{\min}$. At the discrete total path length s_i of the piece-wise linear reference

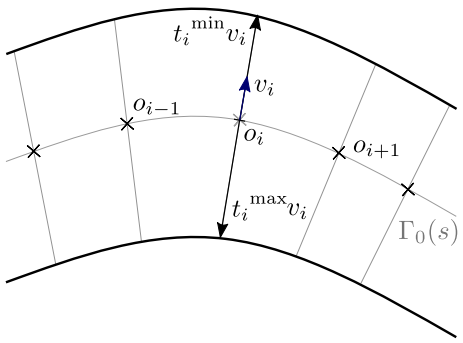

Fig. 5. Geometry of waypoint borders.

path it holds that $\Gamma(s_i) = p_i$. The distance vector between point p_i and p_{i+1} is given by h_i, where $\Delta s_i = s_{i+1} - s_i = \|h_i\|$.

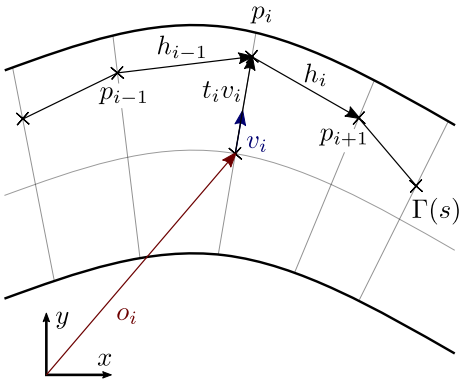

Fig. 6. Geometry of waypoint displacement.

$$p_i = \begin{bmatrix} p_{\mathrm{x},i} \\ p_{\mathrm{y},i} \end{bmatrix} = o_i + t_i v_i \tag{8}$$

$$h_i = \begin{bmatrix} h_{\mathrm{x},i} \\ h_{\mathrm{y},i} \end{bmatrix} = p_{i+1} - p_i \tag{9}$$

As shown in [12], the discrete curvature κ_i for unequally spaced points can be computed by means of the first central and second finite differences, with the notation of D_i^1 for first, D_i^2 for second discrete derivatives and the composition of $D_i^n = \begin{bmatrix} D_{x,i}^n, & D_{y,i}^n \end{bmatrix}^T$. Another method to compute the discrete curvature would be the method of osculating circles as shown in [13].

$$D_i^1 := -\frac{\Delta s_i}{\Delta s_{i-1}(\Delta s_i + \Delta s_{i-1})} p_{i-1} - \frac{\Delta s_i + \Delta s_{i-1}}{\Delta s_i \Delta s_{i-1}} p_i + \frac{\Delta s_{i-1}}{\Delta s_i(\Delta s_i + \Delta s_{i-1})} p_{i+1} \tag{10a}$$

$$D_i^2 := 2\frac{\Delta s_i p_{i-1} - (\Delta s_i + \Delta s_{i-1}) p_i + s_{i+1} p_{i+1}}{\Delta s_{i-1} \Delta s_i (\Delta s_i + \Delta s_{i-1})} \tag{10b}$$

$$\kappa_i(t_{i-1}, t_i, t_{i+1}) = \frac{D_{\mathrm{x},i}^1 D_{\mathrm{y},i}^2 - D_{\mathrm{y},i}^1 D_{\mathrm{x},i}^2}{((D_{\mathrm{x},i}^1)^2 + (D_{\mathrm{y},i}^1)^2)^{\frac{3}{2}}} \tag{11}$$

The discrete curvature is now used to formulate an overall objective, which is the sum of several objectives. First, the maximum value of the ratio of the "inner" (the side where the curve bends) boundary distance ($t_i^{\min} - t_i$ if $\kappa_i < 0$ or $t_i^{\max} - t_i$ if $\kappa_i > 0$) to the signed radius of curvature $R_i = 1/\kappa_i$ is minimized by means of the slack variable $\bar{\rho}$. The osculating circle to point p_i is sketched in Fig. 7, which also shows the maximum deflection $t_i^{\min} - t_i$ of the reference curve $\Gamma(s)$ at point $s = s_i$. Note that the vector to the center of the osculating circle which is a multiple of the new tangent vector $\bar{v}_i$ to $\Gamma(s_i)$ generally does not need to be parallel to the tangent vector v_i of the initial curve $\Gamma_0(s_i)$. Nevertheless it is assumed that the vectors v_i and $\bar{v}_i$ are equal, since the mismatch is generally small, if the curve Γ_0 is initialized properly, i.e. close to the center line. The slack variable $\bar{\rho}$ bounds the maximum allowed value of the curvature ratio $\rho = n\kappa$ and is also bounded by a value $\bar{\rho}_{max}$, which should be between 0.5 and 0.95 and penalized by a cost $T_\rho(\bar{\rho})$. The optimization figuratively pushes the evolute shown in Fig. 2 off the boundaries and guarantees that they do not intersect by the constraint on $\bar{\rho}$. The parameter w_ρ contributes to the shape of the penalty function for relevant values $0 < \bar{\rho}_i < \bar{\rho}_{max}$ and might also be set to zero, if $\bar{\rho}_{max}$ is used conservatively (i.e. low values). It is noteworthy that this

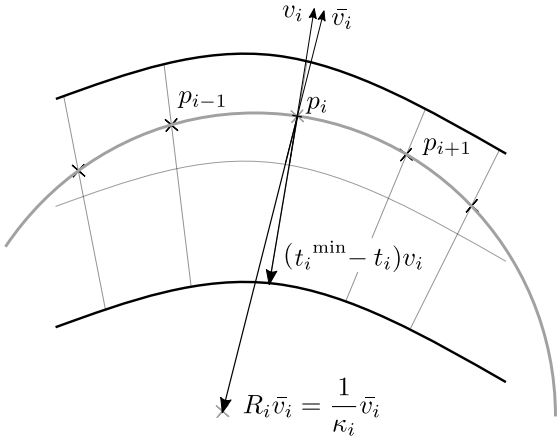

Fig. 7. Geometry of osculating circle.

objective stated alone would lead to a flat minimum, which would result in nonsingular solutions. Secondly, the discrete

derivative of the curvature is minimized and weighted by $w_{\mathrm{d}\kappa}$ in objective $T_{\mathrm{d}\kappa}$. Thirdly, a term $T_{\mathrm{dc}}(t)$ is added, which can be used to force the resulting transformation line towards the road center. This might be useful if obstacles parallel to the center-line are added.

$$T_{\mathrm{d}\kappa}(t) = \sum_{i=1}^{N-2} \left(\frac{\kappa_{i+1}(t_i, t_{i+1}, t_{i+2}) - \kappa_i(t_{i-1}, t_i, t_{i+1})}{\|h_i(t_i, t_{i+1})\|} \right)^2 \tag{12a}$$

$$T_{\mathrm{dc}}(t) = \sum_{i=0}^{N} \left(\frac{t_i^{\min} + t_i^{\max}}{2} - t_i \right)^2 \tag{12b}$$

$$T_{\rho}(\bar{\rho}) = \sum_{i=1}^{N-1} \frac{\bar{\rho}_i}{1 - \bar{\rho}_i} \tag{12c}$$

$$\begin{aligned} \min_{\bar{\rho} \in \mathbb{R}^{N-1}, t \in \mathbb{R}^N} \quad & w_\rho T_\rho(\bar{\rho}) + w_{\mathrm{d}\kappa} T_{\mathrm{d}\kappa}(t) + w_{\mathrm{dc}} T_{\mathrm{dc}}(t) \\ \text{s.t.} \quad & t_i^{\min} \le t_i \le t_i^{\max} \quad i = 0, \dots, N, \\ & (t_i^{\min} - t_i)\kappa_i \le \bar{\rho}_i \quad i = 1, \dots, N-1, \\ & (t_i^{\max} - t_i)\kappa_i \le \bar{\rho}_i \quad i = 1, \dots, N-1, \\ & \bar{\rho}_i \le \bar{\rho}_{\max} \quad i = 1, \dots, N-1 \end{aligned} \tag{13}$$

VI. SIMULATION RESULTS

A. Integration error

Since the integration scheme which is used in order to obtain an NLP out of the OCP by direct multiple shooting is one of the most important contributors to the overall accuracy of the OCP solution, the improvement of the integration performance is shown in a simplified setting (Fig. 8). The vehicle model ($l_{\mathrm{r}} = 2\,\mathrm{m}$ and $l_{\mathrm{f}} = 1\,\mathrm{m}$) in the Cartesian frame (3) as well as the vehicle model in the Frenet frame (3a) are forward simulated with a constant speed of $1\,\frac{\mathrm{m}}{\mathrm{s}}$ and a constant steering angle of $\delta = -0.15\,\mathrm{rad}$. The "Frenet model" is transformed into the Frenet frame with respect to a transformation curve, given as a circle. This circular transformation curve leads to a constant curvature and the center point of the circle represents the whole singularity. In subsequent experiments this circle is displaced along a line which crosses the motion path of the vehicle. The displacement is characterized by the maximum curvature ratio $\rho_{\max}$, which is the maximum value of the curvature ratio $\rho(s, n)$ along the exactly simulated trajectory. This value would be equal to zero if the integrated trajectory lies exactly on the transformation curve (circle perimeter) and equal to one if the trajectory crosses the radius center (singularity). Two different integration schemes are used (Euler and Runge-Kutta-4 (RK4)) in order to demonstrate the superior integration result in terms of the end-point error to exactly simulate a quarter-circle path. Additionally, each integration scheme is performed with different step lengths. As a step size $\Delta t_0 = 2.6\,\mathrm{s}$ is used. In Fig. 9 the integration schemes are compared, dependent on the curvature ratio ρ for a certain step size. The Euler integration was performed with 4 times the number of steps than the

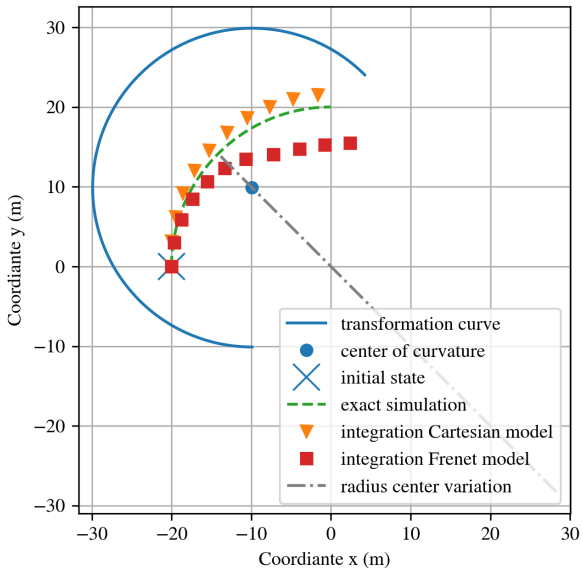

Fig. 8. Integration setting.

RK4 integration to compare them related to their number of function evaluations. It can easily be verified that at some value of the curvature ratio ρ each integration scheme shows high errors and at some ratio ρ the end-point simulation error is rather randomly close to the exact solution. Based on this observations, for a given integration scheme it is obvious to set an upper border for the curvature ratio ρ in order to define the transformation curve.

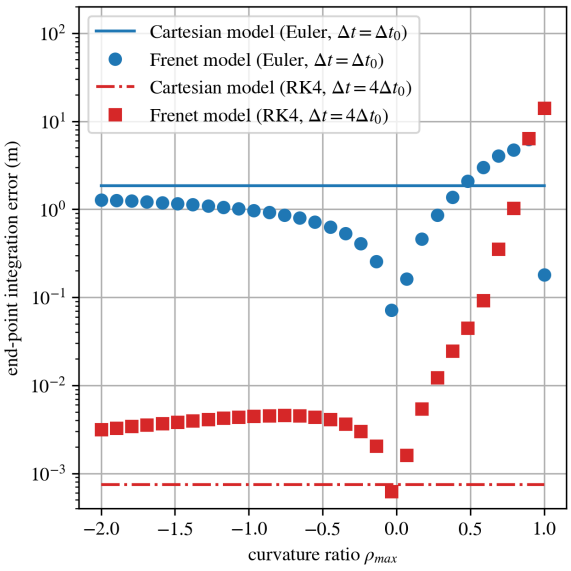

Fig. 9. Integration errors.

The integration performs best if the simulated path is located exactly at the transformation curve (curvature ratio $\rho = 0$) and performs poorly, if the path is located close to the

singularity (curvature ratio $\rho = 1$). For negative curvature ratios ρ Euler integration performs always better with the Frenet model. Although, if the Runge-Kutta-4 integration scheme is used, the Cartesian model is superior, despite for a curvature ratio of $\rho = 0$.

B. Model predictive control

To point out the superior properties of the presented approach for parameterizing the transformation curve, an exemplary task for time-optimal MPC is solved. The time-optimal trajectory for the curve in Fig.2 is obtained by solving an NLP of the form (5). Two different transformation curves are compared as shown in Fig. 1 for the center line and Fig. 2 for the numerically better behaving parameterized optimal transformation curve, which was obtained by solving (13) ($w_{\mathrm{dc}} = 10$, $w_\rho = 10$, $\bar{\rho}_{\max} = 0.7$ and $w_{\mathrm{d}\kappa} = 10^8$). The parameters of the vehicle model were taken from a real race car model with wheel bases $l_{\mathrm{r}} = 1.4$m, $l_{\mathrm{f}} = 1.6$m, a maximum lateral and longitudinal acceleration of $5\frac{\mathrm{m}}{\mathrm{s}^2}$ for both. The OCP was discretized with a horizon $T = 9$s and a discretization time of $\Delta T = 0.05$s. Other values are equivalent to [4]. For solving the NLP, acados [14] was used with a RK4 integrator performing one step per multiple shooting interval. The problem was solved with 40 different starting positions as marked in Fig. 1 and the resulting numerical differences are shown in Table I. The presented parameterization results in superior numerical properties for all relevant performance measures. Note that the so called "center curve" was also re-parameterized slightly in order to obtain smooth system differential equations ($w_{\mathrm{dc}} = 10^3$, $w_\rho = 10$, $\bar{\rho}_{\max} = 0.9$ and $w_{\mathrm{d}\kappa} = 10^6$). Without the presented parameterization the solver failed in a significant amount of simulation runs, which is another indicator that some form of parameterization like the presented approach is even necessary for most applications of Frenet model MPC.

TABLE I
COMPARISON OF TRANSFORMATION CURVES FOR MPC.

Statistical NLP value	center line	optimal trans. curve
solution time (mean)	611.1ms	470.9ms
SQP iterations (mean)	11.0	8.2
QP iterations (mean)	22.3	11.1
QP solver fails	40%	0%

VII. CONCLUSIONS

The paper presents a parameterization approach that is suggested to be used with Frenet transformations related to numerical optimization approaches for autonomous driving. It formalizes the problem of finding the transformation curve, which must not have singularities in the feasible driving region. A parameterization, stated as an optimization problem is presented, which ensures feasibility as well as superior numerical properties for Newton-type optimization. The performance increase is shown by means of a plain integration and a small but relevant test example. Further considerations and future work might focus on the related nonconvexity of the constraints, the implementation as a real time capable NLP or the consecutively needed transformation of the borders.

ACKNOWLEDGMENT

The publication was partially funded within the COMET K2 Competence Centers for Excellent Technologies from the Austrian Federal Ministry for Climate Action (BMK), the Austrian Federal Ministry for Digital and Economic Affairs (BMDW), the Province of Styria and the Styrian Business Promotion Agency (SFG). The Austrian Research Promotion Agency (FFG) has been authorised for the programme management. This research was also supported by the German Federal Ministry for Economic Affairs and Energy (BMWi) via DyConPV (0324166B). The authors thank Prof. Dr. Anton Gfrerrer (Institute of Geometry, Technical University of Graz), DI Daniel Klöser (Christian-Albrechts-University Kiel), Dr. Johannes Rumetshofer (VVRC), DI Jonathan Frey (ALUF) and Dr. Michael Stolz (VVRC) for their comments and valuable input to this work.

REFERENCES

[1] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous uav guidance," *Journal of Intelligent and Robotic Systems*, vol. 57, pp. 65–100, 11 2010.

[2] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[3] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[4] D. Kloeser, T. Schoels, T. Sartor, A. Zanelli, G. Frison, and M. Diehl, "NMPC for racing using a singularity-free path-parametric model with obstacle avoidance," in *Proceedings of the IFAC World Congress*, 2020.

[5] T. Novi, A. Liniger, R. Capitani, and C. Annicchiarico, "Real-time control for at-limit handling driving on a predefined path," *Vehicle System Dynamics*, vol. 58, no. 7, pp. 1007–1036, 2020.

[6] S. Chen and H. Chen, "MPC-based path tracking with PID speed control for autonomous vehicles," *IOP Conference Series: Materials Science and Engineering*, vol. 892, no. 1, 2020.

[7] J. L. Vázquez, M. Brühlmeier, A. Liniger, A. Rupenyan, and J. Lygeros, "Optimization-based hierarchical motion planning for autonomous racing," *ArXiv*, vol. abs/2003.04882, 2020.

[8] R. Verschueren, M. Zanon, R. Quirynen, and M. Diehl, "Time-optimal race car driving using an online exact hessian based nonlinear MPC algorithm," *2016 European Control Conference, ECC 2016*, pp. 141–147, 2017.

[9] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," 06 2010, pp. 987 – 993.

[10] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Nob Hill, 2017.

[11] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[12] A. K. Singh and B. S. Bhadauria, "Finite difference formulae for unequal sub-intervals using Lagrange's interpolation formula," *International Journal of Mathematical Analysis*, vol. 3, no. 17-20, pp. 815–827, 2009.

[13] J. Hoschek and D. Lasser, *Fundamentals of Computer-aided Geometric Design*. Jones and Bartlett, 1993. [Online]. Available: https://books.google.at/books?id=IJxnIAAACAAJ

[14] R. Verschueren, G. Frison, D. Kouzoupis, N. van Duijkeren, A. Zanelli, B. Novoselnik, J. Frey, T. Albin, R. Quirynen, and M. Diehl, "acados: a modular open-source framework for fast embedded optimal control," 2019.