Long-Horizon Vehicle Motion Planning and Control Through Serially Cascaded Model Complexity

Vincent A. Laurense, *Member, IEEE*, and J. Christian Gerdes, *Member, IEEE*

***Abstract*— The computational burden of nonlinear model predictive control (NMPC) often limits its use to short planning horizons, simple systems with slow dynamics, offline applications, or approximations of the optimal control problem. This article introduces a novel concept for NMPC, to help enable real-time integrated motion planning and control with a long planning horizon for automated vehicles. The proposed framework cascades plant models of different levels of complexity within a single planning horizon, in a single optimization problem. Leveraging the receding nature of MPC, a high-fidelity plant model in the first part of the planning horizon continuously provides a high quality of control, while the planning horizon is extended significantly at low computational cost with a lower fidelity model. Cascading the model complexity serially in a single planning horizon, rather than in different control loops, avoids infeasible reference trajectories between control loops. The concept is successfully validated with real-time motion planning and control of a full-scale automated race car, featuring combined lateral and longitudinal control and operating the vehicle near the limits of tire-road friction. The framework is deployed with open-source numerical optimization tools. In the real-world experiment, the proposed design both better approaches the optimal minimum-time solution and has a lower median solve time compared to a benchmark architecture with a single-vehicle model and a necessarily shorter planning horizon.**

***Index Terms*— Autonomous vehicles, nonlinear model predictive control (NMPC), real-time motion planning, reduced order systems.**

I. Introduction

MANY physical systems exhibit nonlinear dynamics and are subject to time-varying nonlinear constraints. Nonlinear model predictive control (NMPC) provides a means of determining inputs for such a system that optimize a certain objective function while respecting these nonlinear dynamics and constraints. The computational expense of the resulting nonlinear program (NLP) is frequently a major obstacle to applying this powerful technique.

The task of planning a dynamically feasible and collision-free trajectory of an automated road vehicle through its environment showcases both the strengths and limitations of NMPC. This integrated planning and control problem is nonlinear and involves making important tradeoffs between lateral and longitudinal motion—a combination for which NMPC is ideally suited. The computational demands, however, are daunting. Typical vehicle speeds require a relatively long planning horizon of multiple seconds, while vehicle stabilization and obstacle avoidance require a fast control loop and a sufficient level of model complexity to capture the relevant dynamics. A simulation study in this article shows that it is impossible to solve the resulting NLP with current in-vehicle computer hardware in real time.

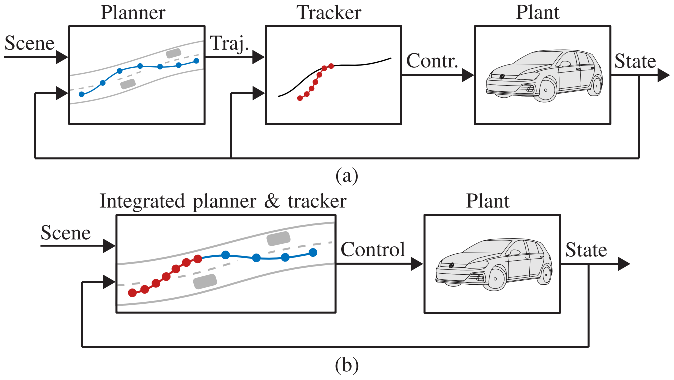

Fig. 1. Block diagrams for two different architectures for vehicle motion and planning and control. (a) A hierarchical approach with multiple control loops. (b) The proposed single-loop approach for vehicle motion planning and control with serially cascaded model complexity.

In the literature, various methods have been developed for real-time motion planning and control. A popular method is to approximate the problem and address different tasks in different control loops, as shown in Fig. 1(a). For example, Gros *et al.* [1] presented a two-time-scale scheme with MPC in a slow trajectory generation loop and a fast inner loop with time-varying linear feedback for the control of a quadcopter. Gao *et al.* [2] showed experimental results of obstacle avoidance for ground vehicles, with a point-mass model for high-level path planning and a simple four-wheel model for low-level path following. In a similar approach for a simulated race environment, Novi *et al.* [3] constructed a hierarchical approach with NMPC based on a point-mass model in an outer loop, providing a near-term terminal velocity constraint for short-horizon NMPC with a planar four-wheel vehicle model.

With simpler models in outer control loops, however, the target trajectory that a high-level planner passes to a low-level tracking controller could be infeasible to track or overly conservative—both of which are undesirable. A more accurate high-level planner (which requires more time to solve) is not able to swiftly respond to disturbances or changes in the environment, which, of course, can also be dangerous.

Gray *et al.* [4] addressed the issue of infeasibility of the reference trajectory generated by a high-level planner in a multiple-loop architecture with motion primitives. These can

Manuscript received April 18, 2020; revised October 18, 2020; accepted January 18, 2021. Date of publication March 1, 2021; date of current version December 15, 2021. Manuscript received in final form January 29, 2021. Recommended by Associate Editor M. Tanelli. *(Corresponding author: Vincent A. Laurense.)*

The authors are with the Department of Mechanical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: laurense@alumni.stanford.edu; cgerdes@stanford.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCST.2021.3056315.
Digital Object Identifier 10.1109/TCST.2021.3056315

be constructed with a high-fidelity model; however, the resulting set of possible control actions is finite and in general suboptimal.

A second general method is to approximate the NLP by a quadratic program (QP), for which fast convex optimization tools can be leveraged, for example as described by Wang and Boyd [5]. Falcone *et al.* [6] and Funke *et al.* [7] set up a time-varying QP with online model linearization for lateral control and stabilization of an automated vehicle, with convex constraints on the inputs and states that prevent the vehicle from operating in unstable conditions. This provides optimal steering control for a predetermined speed, but these examples do not consider the tradeoffs in combined lateral and longitudinal control and in general solve only an approximation of the underlying nonlinear control problem.

A different approach considers model complexity reduction with constraint augmentation. Liu *et al.* [8] demonstrated that rollover risk mitigation for a vehicle accurately modeled with 14 degrees of freedom could be achieved with a conservative model with two degrees of freedom augmented with a nonlinear bound on the maximum steering angle. In this example, the conservative bound could prevent rollover but does increase the minimum turning radius, potentially inhibiting the controller's potential to avoid an obstacle even if that is physically possible. In general, conservative bounds on a simpler vehicle model can nominally keep the system away from dangerous regions of the state space but prevent the controller from using the system's full physical capabilities in emergency situations.

Yet, another approach is to solve the optimal control problem offline, rather than online, for a finite number of polyhedral regions of the parameter vector. Thereby, the so-called explicit MPC reduces the online computation to polyhedron identification and the evaluation of a suboptimal state feedback law. Metzler *et al.* employed this technique for vehicle stability control through braking [9], and Canale *et al.* employed this technique for vehicle stability control through control of an active rear differential [10]. However, when the number of states, inputs, or linear inequalities gets large, explicit MPC is no longer practically feasible. Moreover, this method cannot handle optimization problems in which the system, cost function, or constraints are time-varying [5].

Lastly, Diehl *et al.* [11] introduced a real-time iteration scheme that provides approximations of the optimal feedback control which are iteratively refined during the runtime of the controlled process. Frasch *et al.* [12] and Siampis *et al.* [13] explored this method for real-time obstacle avoidance for ground vehicles. The approximate solution from real-time iteration is suboptimal and potentially infeasible, and the stability of the framework can only be guaranteed if disturbances from one controller tick to the next are sufficiently small.

In the control of safety-critical systems such as automated vehicles, at no level and at no time should the framework be overly conservative, suboptimal, or generating infeasible reference signals, in order to maximize the capability of controlling the system away from critical states. This article seeks to address that with a novel concept for real-time NMPC with a long planning horizon, in which vehicle models of different levels of complexity are cascaded within a *single* horizon, as shown in Fig. 1(b), in a single optimization problem. With the receding nature of MPC, a high-fidelity plant model continuously provides dynamically feasible trajectories in the near term that can control the system to its full potential, whereas a low-fidelity model allows for significant extension of the planning horizon at low computational cost.

In addition to illustrating this concept of serially cascaded model complexity through simulation, we experimentally validate the concept with real-time motion planning and control of a full-scale automated race car. The scenario of racing is purposefully challenging; the minimum-time objective constantly pushes the vehicle to its limits, important tradeoffs in longitudinal and lateral control need to be made, and the nonlinear tire dynamics are significant. Racing does not allow for conservatism, and lap time provides an absolute benchmark. By pushing an automated vehicle to its limits on the race track, we can learn how to reliably address a vehicle's full physical potential in emergency situations on the street.

The experiment compares the proposed cascaded model architecture to a benchmark architecture with a single high-fidelity model, for which the planning horizon is necessarily shorter due to the computational limits for online optimization. The results show that the cascaded model design achieves a lower lap time, while the median NLP solve time is also lower.

This article is structured as follows. First, Section II provides more details on the concept of cascaded model complexity in MPC-based optimal control. Section III introduces two vehicle models of different levels of complexity that are common in vehicle motion planning and control—the dynamic single-track model and the point-mass model. The nonlinear optimization problem with serially cascaded model complexity is detailed in Section IV, and Section V presents the experiment design. Simulation results in Section VI illustrate how serially cascaded model complexity can help enable NMPC for real-time applications, followed by successful experimental results with an automated research vehicle in Section VII. Final conclusions are presented in Section VIII.

II. Concept of Cascaded Model Complexity

In a control system design, it is evident that a plant model needs to be selected that captures all relevant dynamics to meet the control objectives. For example, in the context of vehicle motion planning and control, the objective of stabilizing the car requires a model that includes its yaw dynamics. For the objective of avoiding obstacles and road departures, one needs a model for planning the vehicle's position, direction of travel, and speed over a sufficiently long planning horizon.

The control objectives, however, do not necessarily have a constant level of relevance, or criticality, throughout the planning horizon. As an example, for vehicle stabilization, the steering inputs on the order of a second are crucial. For safely negotiating a turn that is further in the future, on the other hand, the exact steering inputs and yaw dynamics are less (or: not yet) relevant—but for an upcoming turn, it is already important to plan a safe path and start decelerating to an appropriate speed.

For model-based optimal control, the combination of the highest level of model complexity and the longest planning horizon required to meet all control objectives leads to a

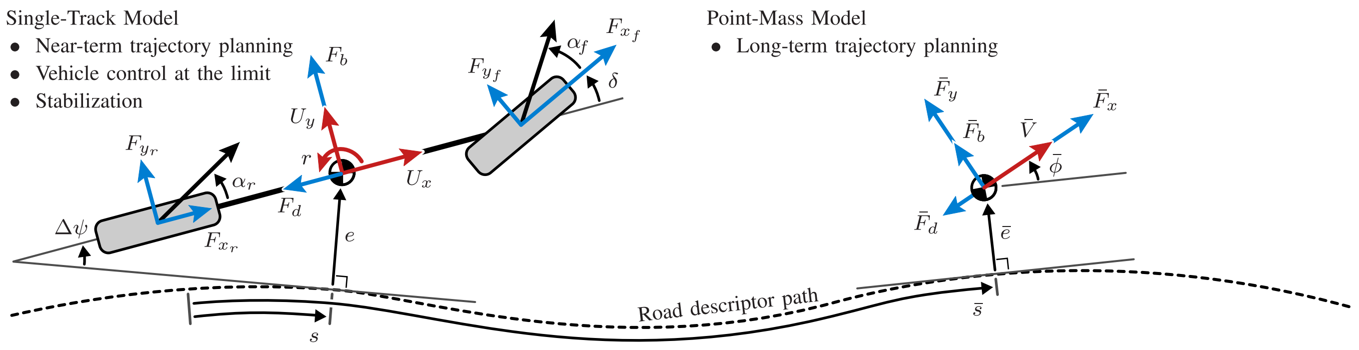

Fig. 2. Dynamic single-track vehicle model (left) and point-mass vehicle model (right) with their states relative to the road descriptor path (top view).

significant computational burden. This often prohibits real-time applications. However, different control tasks could be addressed with different plant models and different planning horizons. This is the strategy in controller hierarchies with multiple loops (shown in Fig. 1(a) and discussed in Section I), with an outer, potentially slower, planning loop with a long horizon and relatively simple plant model, and an inner tracking loop with a higher fidelity model. A simple plant model for high-level trajectory planning, however, could lead to infeasible reference signals between the control loops or conservative reference signals that do not plan to use the system to its full potential.

The novel concept introduced in this article cascades the vehicle models in a *single* planning horizon serially and computes the optimal control sequence from a single NLP. This eliminates slow control loops and infeasible reference signals between loops. By leveraging the receding nature of MPC, a high-fidelity plant model in the first part of the planning horizon continuously provides the high quality of control, using the system to its full potential without controller switching, while the planning horizon is extended significantly at low computational cost with a lower fidelity model. The resulting NLP has a low computational burden that makes it tractable for real-time control.

When serially cascading different plant models, the final state of one model must be carefully propagated to the initial state of the next model. In addition, the constraints and cost function terms should be consistent across the cascaded plant models, to avoid inconsistencies in the closed-loop behavior.

Finally, it is noted that when more computational resources become available in the future, the concept of cascaded model complexity does not necessarily become superfluous. Any plant model mismatch integrates over the open-loop planning horizon, and in addition, the system could be disturbed by external factors, making the fast and less damped states in a high-fidelity model far into the planning horizon less reliable. With the proposed framework, an efficient controller could therefore be constructed without delivering a false sense of accuracy of its planned actions.

III. Vehicle Dynamics Modeling

This section introduces two vehicle models that are common in the field of vehicle motion planning and control: the dynamic single-track model and the point-mass model. Section III-C briefly discusses the concatenation of these models in the proposed serially cascaded model architecture.

A. Dynamic Single-Track Vehicle Model

1) Overview and Equations of Motion: In the dynamic single-track vehicle model, the two tires per axle are lumped together into a single tire, which is parameterized in such a way to incorporate relevant effects such as steady-state lateral load transfer. This provides a simple planar model that includes the vehicle's yaw dynamics, which allows for stabilization. Combined with a nonlinear tire model, it describes the vehicle's dynamics up to the very limits of its physical capabilities.

The model, shown on the left in Fig. 2 with sign conventions specified by ISO 8855 [14], has three velocity states associated with its center of gravity (CG): longitudinal speed U_x, lateral speed U_y, and yaw rate r. Furthermore, there are three position states in relation to a road descriptor path: the curvilinear coordinate along this path and the lateral distance to this path, s and e, respectively, and the difference in heading between the vehicle's chassis and the path, denoted by $\Delta\psi$. The path's curvature κ defines the horizontal path geometry.

The vehicle has mass m and yaw moment of inertia I_{zz}, and the front steer angle is indicated by δ. The distances from the CG to the front and rear axle are denoted by a and b, respectively, summing up to the wheelbase L, as shown in the side-view schematic in Fig. 3. The CG is at a distance h_{cg} above the ground. The front and rear longitudinal axle forces are denoted by F_{x_f} and F_{x_r}, respectively; the lateral axle forces are denoted by F_{y_f} and F_{y_r}.

The equations of motion for this model are as follows:

$$\dot{U}_x = \frac{F_{x_f}\cos\delta - F_{y_f}\sin\delta + F_{x_r} - F_d}{m} + rU_y \tag{1a}$$

$$\dot{U}_y = \frac{F_{y_f}\cos\delta + F_{x_f}\sin\delta + F_{y_r} + F_b}{m} - rU_x \tag{1b}$$

$$\dot{r} = \frac{a\left(F_{y_f}\cos\delta + F_{x_f}\sin\delta\right) - bF_{y_r}}{I_{zz}} \tag{1c}$$

$$\dot{s} = \frac{U_x\cos\Delta\psi - U_y\sin\Delta\psi}{1 - \kappa e} \tag{1d}$$

$$\dot{e} = U_x\sin\Delta\psi + U_y\cos\Delta\psi \tag{1e}$$

$$\Delta\dot{\psi} = r - \kappa\dot{s}. \tag{1f}$$

The sum of drag forces, e.g., rolling resistance F_{rr} and aerodynamic drag with coefficient C_D, is represented by F_d

$$F_d = F_{\text{rr}} + C_D U_x^2 - mg\sin\theta. \tag{2}$$

This model also includes the effect of the road grade angle θ (downhill positive, as in Fig. 3). Similarly, the effect of the

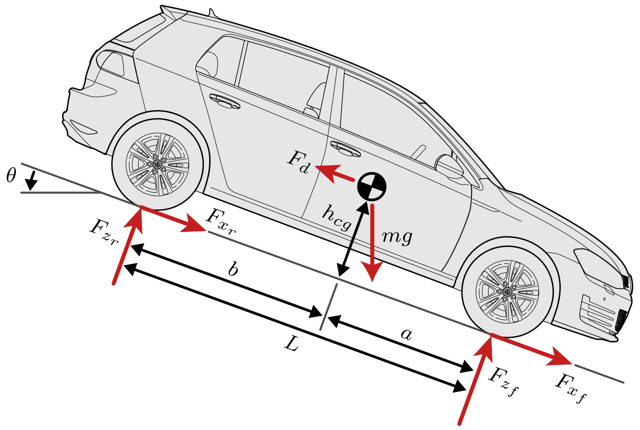

Fig. 3. Side-view schematic of the single-track vehicle model.

road bank angle φ is modeled by the lateral force F_b

$$F_b = -mg\cos\theta\sin\varphi \tag{3}$$

where positive φ denotes a road surface tilted to the right.

2) Input Model: The selected inputs for this model are the steering angle rate $\dot{\delta}$ and the total longitudinal force command F_x. The drive/brake force distribution χ provides the mapping from the total longitudinal force to the longitudinal force at each of the axles

$$F_{x_f} = \chi_f F_x \tag{4a}$$
$$F_{x_r} = \chi_r F_x \tag{4b}$$

where $\chi_f + \chi_r = 1$.

However, the front/rear proportioning of total longitudinal force is generally not equal for drive forces and brake forces. For example, for a front-wheel-drive car, the fraction of engine torque on the front axle is $\chi_{f_d} = 1$, but since a car also has rear brakes, only a fraction of the total longitudinal force command is demanded from the front axle in braking, i.e., $\chi_{f_b} < 1$. Thus, the distribution of the total longitudinal force command χ depends on the sign of F_x

$$(\chi_f, \chi_r) = \begin{cases} (\chi_{f_d}, \chi_{r_d}), & \text{if } F_x \geq 0 \\ (\chi_{f_b}, \chi_{r_b}), & \text{otherwise.} \end{cases} \tag{5}$$

Fig. 4 shows that a simple longitudinal force distribution with constant brake proportioning on a front-wheel-drive vehicle is discontinuous. For numerical optimization with an interior-point line-search filter method, such as IPOPT, however, the objective and the constraint functions of the NLP should be twice continuously differentiable [15]. Therefore, for numerical optimization, the drive/brake force distribution model is approximated as follows:

$$\tilde{\chi}_f = \frac{d_f - b_f}{2}\tanh(2(F_x + 0.5)) + \frac{d_f + b_f}{2} \tag{6a}$$
$$\tilde{\chi}_r = \frac{b_r - d_r}{2}\tanh(-2(F_x + 0.5)) + \frac{d_r + b_r}{2} \tag{6b}$$

where F_x is in kilonewtons. The approximation function is slightly offset from zero to better reflect the fact that the experimental vehicle used in this research is purely front-wheel drive and is overlaid with the true mechanical longitudinal force distribution in Fig. 4. This differentiable function approximates the front and rear longitudinal tire forces as a function of the total force command

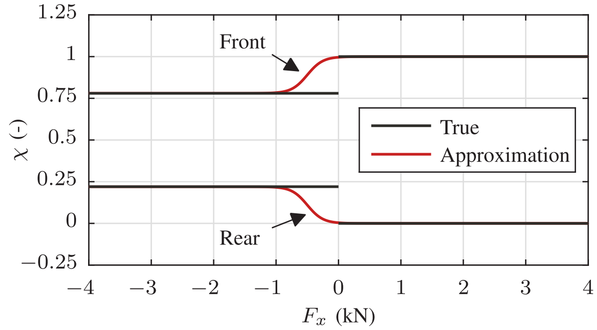

Fig. 4. Front/rear longitudinal tire force distribution as a function of the total longitudinal force; discontinuous true function and the differentiable approximation from (6), for the parameters listed in Table I.

$$\tilde{F}_{x_f} = \tilde{\chi}_f(F_x)\cdot F_x \tag{7a}$$
$$\tilde{F}_{x_r} = \tilde{\chi}_r(F_x)\cdot F_x. \tag{7b}$$

3) Tire Model: For the dynamic single-track model, a coupled nonlinear brush tire model describes the lateral axle forces, as derived by Pacejka [16] and adapted by Hindiyeh [17]. In this model, the maximum lateral tire force potential is derated with the tire's longitudinal force command

$$F_y^{\max} = \sqrt{(\mu F_z)^2 - (\tilde{F}_x)^2} \tag{8}$$

where μ denotes the tire-road friction coefficient and F_z is the tire's instantaneous normal load. The longitudinal force command cannot exceed the tire force potential, i.e., $|\tilde{F}_x| \leq \mu F_z$.

In fact, in numerical optimization, difficulties arise with (8) when $|\tilde{F}_x| = \mu F_z$, i.e., during maximum straight line acceleration or deceleration, because the partial derivative of $F_y^{\max}$ with respect to $\tilde{F}_x$ vanishes. To avoid this, $F_y^{\max}$ is approximated, and in the optimization problem, the maximum lateral force derating is implemented with only 99% of the longitudinal input force

$$\tilde{F}_y^{\max} = \sqrt{(\mu F_z)^2 - (0.99\tilde{F}_x)^2}. \tag{9}$$

The brush tire model then describes the lateral force for each of the lumped tires as a function of slip angle α

$$F_y = \begin{cases} -C_\alpha \tan\alpha \\ \quad + \dfrac{C_\alpha^2}{3\,F_y^{\max}}|\tan\alpha|\tan\alpha \\ \quad - \dfrac{C_\alpha^3}{27\left(F_y^{\max}\right)^2}\tan^3\alpha, & \text{if } |\alpha| \leq \alpha^{\text{sl}} \\ -F_y^{\max}\,\text{sgn}\,\alpha, & \text{otherwise.} \end{cases} \tag{10}$$

The lumped tire model is defined by the axle's cornering stiffness C_α and the tire-road friction coefficient μ. These parameters are fit to measurement data from ramp steer maneuvers [18]. In this way, the effect of steady-state lateral load transfer is accounted for in the single-track model.

The front and rear slip angles in the single-track model (shown in Fig. 2) are computed from

$$\alpha_f = \arctan\left(\frac{U_y + ar}{U_x}\right) - \delta \tag{11a}$$

$$\alpha_r = \arctan\left(\frac{U_y - br}{U_x}\right). \tag{11b}$$

Full sliding, also denoted as saturation, of a tire contact patch occurs at sliding slip angle α^{sl} [17]

$$\alpha^{\mathrm{sl}} = \arctan\left(\frac{3\ F_y^{\max}}{C_\alpha}\right). \tag{12}$$

Fig. 5 shows the brush model in (10) with experimental data from the ramp steer maneuvers. The figure shows that the tire model describes the measurement data well.

The figure also illustrates that any front slip angle exceeding $\pm\alpha_f^{\mathrm{sl}}$ generates a constant lateral force. This too can cause difficulties in numerical optimization, because when the local derivative vanishes, steering inputs are rendered completely ineffective. In this case, a gradient-based optimization scheme might not be able to find a steering angle command that would actually put the tires back in the nonsaturated region to realize a decrease in lateral force.

Anecdotally, when the planning horizon always contains some stages in which the tires are not fully saturated and the steering inputs are coupled through steering rate constraints or costs, the solver should have a way to pull saturated stages out of the sliding region. However, with sustained cornering in racing or planning a long evasive maneuver in a single direction, this cannot be guaranteed. Therefore, the tire model is modified to have a nonzero slope in the saturated region, making the model strictly monotonic. The tangent line modification starts at fraction ξ of the maximum lateral force

$$\alpha^{\mathrm{mod}} = \arctan\left(\frac{3\ F_y^{\max}\xi}{C_\alpha}\right) \tag{13}$$

where ξ is a user-defined parameter between 0 and 1, leaving the tire model intact up to that point. Matching slopes yields the following modified tire model:

$$\tilde{F}_y = \begin{cases} -C_\alpha \tan\alpha \\ \quad + \dfrac{C_\alpha^2}{3\tilde{F}_y^{\max}}|\tan\alpha|\tan\alpha \\ \quad - \dfrac{C_\alpha^3}{27\left(\tilde{F}_y^{\max}\right)^2}\tan^3\alpha & \text{if } |\alpha| \le \alpha^{\mathrm{mod}} \\ -C_\alpha\left(1 - 2\xi + \xi^2\right)\tan\alpha \\ \quad - \tilde{F}_y^{\max}\left(3\xi^2 - 2\xi^3\right)\operatorname{sgn}\alpha & \text{otherwise} \end{cases} \tag{14a}$$

$$= \tilde{f}_{\mathrm{tire}}(\alpha, F_z, F_x). \tag{14b}$$

Fig. 5 shows the original and the modified tire model with $\xi = 0.85$, when $F_{x_f} = 0$ kN and 7 kN. The mismatch between the measurement data and the modified model is minor.

The modified tire model implies that more lateral tire force is available with larger slip angles, but these higher forces cannot be physically realized. Therefore, in motion planning and control, the NLP needs to penalize slip angles exceeding $\pm\alpha^{\mathrm{mod}}$ to discourage the solver from exploiting this.

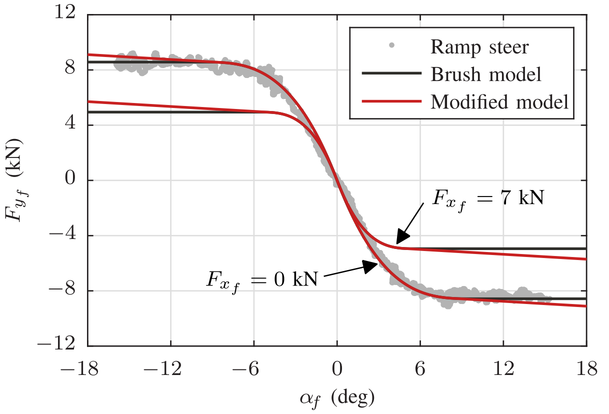

Fig. 5. Original and modified ($\xi = 0.85$) model for the front axle lateral force, for $F_{x_f} = 0$ kN and derated by 7 kN, on a flat and level road, with $\mu_f = 0.90$ (dry asphalt) and the parameters listed in Table I, superposed on measurement data.

Although we consider the planar dynamics of the single-track model, the model for front and rear axle normal loads does include the effects of vertical road topography, in addition to steady-state longitudinal load transfer

$$F_{z_f} = \frac{b}{L}m\left(g\cos\theta\cos\varphi + A_{V^2}U_x^2\right) - \frac{h_{cg}}{L}F_x \tag{15a}$$

$$F_{z_r} = \frac{a}{L}m\left(g\cos\theta\cos\varphi + A_{V^2}U_x^2\right) + \frac{h_{cg}}{L}F_x. \tag{15b}$$

In this model, F_x is the total commanded longitudinal force and (following the notation in [19]) A_{V^2} captures the speed effect of vertical curvature (cresting hills) and banked turns on the vehicle's total normal load:

$$A_{V^2} = -\frac{\partial\theta}{\partial s}\cos\varphi - \kappa\sin\varphi\cos\theta. \tag{16}$$

We assume that the vehicle stays relatively close to the road descriptor path for which the topography profile is measured and that the topography is relatively constant along the width of the track.

B. Point-Mass Model

A simpler model of a vehicle is the planar point-mass model, indicated on the right side in Fig. 2, with an overbar over its variables to clearly separate them from the variables associated with the single-track model. The point-mass model has only one velocity state, the total horizontal velocity $\bar{V}$. It too has three position states in relation to the road descriptor path: the distance along this path and the lateral distance to this path, $\bar{s}$ and $\bar{e}$, respectively, and the difference in direction of the velocity vector and the local tangent to the descriptor path, denoted by course error $\bar{\phi}$. The model is described by the following equations of motion:

$$\dot{\bar{V}} = \frac{\bar{F}_x - \bar{F}_d}{m} \tag{17a}$$

$$\dot{\bar{s}} = \frac{\bar{V}\cos\bar{\phi}}{1 - \kappa\bar{e}} \tag{17b}$$

$$\dot{\bar{e}} = \bar{V}\sin\bar{\phi} \tag{17c}$$

$$\dot{\bar{\phi}} = \frac{\bar{F}_y + \bar{F}_b}{m\bar{V}} - \kappa\dot{\bar{s}}. \tag{17d}$$

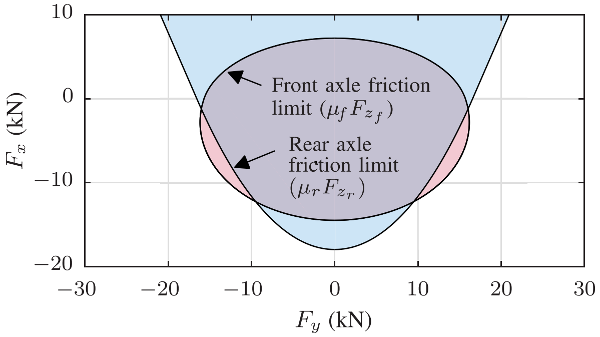

Fig. 6. Total input-force limits for the point-mass model resulting from the intersection of the friction limits at each of the axles, for $\mu_f = 0.90$ and $\mu_r = 0.95$ (dry asphalt), a flat level road, and the parameters listed in Table I.

The model for drag force $\bar{F}_d$ is the same as in (2) but with $\bar{V}$ instead of U_x, and $\bar{F}_b$ is defined by (3). There is no tire force model in this point-mass model; the inputs are the total lateral force $\bar{F}_y$ and total longitudinal force $\bar{F}_x$.

The limits on the input forces follow from the intersection of the friction limits at each of the axles. Under the assumption of steady-state cornering, these limits are

$$\left(\tfrac{b}{L}\bar{F}_y\right)^2 + \left(\tilde{\chi}_f \bar{F}_x\right)^2 \leq \left(\mu_f F_{z_f}\right)^2 \tag{18a}$$

$$\left(\tfrac{a}{L}\bar{F}_y\right)^2 + \left(\tilde{\chi}_r \bar{F}_x\right)^2 \leq \left(\mu_r F_{z_r}\right)^2. \tag{18b}$$

In these inequality constraints, the normal loads F_{z_f} and F_{z_r} are computed according to (15), with $\bar{V}$ instead of U_x. The total input force limits for the point-mass model resulting from the friction limits at each of the axles are shown in Fig. 6.

The friction limits describe ellipses. For a front-wheel-drive vehicle, the rear axle normal load increases significantly under acceleration due to longitudinal load transfer, without this axle generating any longitudinal force. Therefore, the friction limit for the rear axle is considerably larger than that for the front axle.

C. Cascaded Vehicle Models

In a cascaded model design, the translational states of the dynamic single-track model can directly be mapped to the states of the point-mass model

$$\bar{s} = s \tag{19a}$$

$$\bar{V} = \sqrt{U_x^2 + U_y^2} \tag{19b}$$

$$\bar{e} = e \tag{19c}$$

$$\bar{\phi} = \arctan\left(\frac{U_y}{U_x}\right) + \Delta\psi. \tag{19d}$$

The point-mass model does not contain the vehicle's yaw dynamics and can therefore not be used for yaw stabilization. However, the previous two sections reveal two key properties.

1) The point-mass model is simpler and requires fewer optimization variables and constraints.
2) Without the fast yaw dynamics, the point-mass model can be discretized with a larger step size.

Both these properties are crucial in extending the MPC planning horizon at low computational cost.

IV. Implementation of the NLP

A. Overview

The NLP cost function could be configured for various applications. In this case, the cost function is set up to minimize time along a finite planning horizon. The minimum-time objective constantly pushes the vehicle to its limits, forcing the controller to make important tradeoffs in lateral and longitudinal control, and motivates the controller at every time step to seek the fastest trajectory from the vehicle's current state.

For real-world experiments, we desire gradually ramping up the level of acceleration at which the vehicle is operating, for which we introduce two inequality constraints with a user-defined friction limit μ_{lim}. The constraints are relaxed with slack variables F_{e_f} and F_{e_r}: in case of plant model mismatch or external perturbations, these slack variables allow prioritization of mitigating large penalties, e.g., imminent road boundary intrusion or a slip angle violation, over the desired level of accelerations.

The NLP for the proposed controller framework with serially cascaded model complexity, that is presented in more detail in the following sections, is as follows:

$$\begin{aligned}
\min_{\substack{x_k, u_k, z_k,\\ \bar{x}_l, \bar{u}_l}} \quad & \bar{t}_M + J_{\text{term}} + J_{\text{rb}} + J_{\text{rd}} + J_{\dot{u}} + J_\alpha + J_F \\
\text{s.t.} \quad & U_{x_k} \geq U_x^{\min} && \text{(State limits)} \\
& \bar{V}_l \geq \bar{V}^{\min} \\
& \delta^{\min} \leq \delta_k \leq \delta^{\max} \\
& \dot{\delta}^{\min} \leq \dot{\delta}_k \leq \dot{\delta}^{\max} && \text{(Input limits)} \\
& F_{x_k} \leq \frac{P_{\text{eng}}}{U_{x_k}} \\
& \bar{F}_{x_l} \leq \frac{P_{\text{eng}}}{\bar{V}_l} \\
& \tilde{F}_{y_{f\,k}} = \tilde{f}_{\text{tire}}\left(\alpha_f, \tilde{F}_{x_f}, F_{z_f}\right)_k && \text{(Tire model)} \\
& \tilde{F}_{y_{r\,k}} = \tilde{f}_{\text{tire}}\left(\alpha_r, \tilde{F}_{x_r}, F_{z_r}\right)_k \\
& -\mu_f F_{z_{f\,k}} \cos\alpha_{f\,k} \leq \tilde{F}_{x_{f_k}} \leq \mu_f F_{z_{f_k}} \cos\alpha_{f\,k} \\
& -\mu_r F_{z_{r\,k}} \cos\alpha_{r_k} \leq \tilde{F}_{x_{r_k}} \leq \mu_r F_{z_{r_k}} \cos\alpha_{r_k} \\
& x_0 = x_{\text{meas}} && \text{(Dynamics)} \\
& x_{k+1} = g_{\text{st}}(x_k, u_k) \\
& \bar{x}_0 = g_{\text{tr}}(x_N) \\
& \bar{x}_{k+1} = g_{\text{pm}}(\bar{x}_k, \bar{u}_k) \\
& \tilde{F}_{x_{f_k}}^{\;2} + \tilde{F}_{y_{f_k}}^{\;2} \leq (\mu_{\text{lim}} F_{z_{f_k}})^2 + F_{e_{f_k}}^{\;2} && \text{(Friction limits)} \\
& \tilde{F}_{x_{r_k}}^{\;2} + \tilde{F}_{y_{r_k}}^{\;2} \leq (\mu_{\text{lim}} F_{z_{r_k}})^2 + F_{e_{r_k}}^{\;2} \\
& \bar{F}_{x_{f_l}}^{\;2} + \left(\frac{b}{L}\bar{F}_{y_l}\right)^2 \leq \left(\min(\mu_f, \mu_{\text{lim}})\bar{F}_{z_{f_l}}\right)^2 \\
& \bar{F}_{x_{r_l}}^{\;2} + \left(\frac{a}{L}\bar{F}_{y_l}\right)^2 \leq \left(\min(\mu_r, \mu_{\text{lim}})\bar{F}_{z_{r_l}}\right)^2.
\end{aligned} \tag{20}$$

The optimization variables for this discretized problem with multiple shooting are related to the dynamic single-track model and the point-mass model. For the single-track model, the state x, input u, and helper variable z (which helps with

NLP sparsity) at each planning stage k are defined as

$$\left.\begin{aligned} x_k &= \begin{bmatrix} U_x & U_y & r & t & e & \Delta\psi & \delta \end{bmatrix}_k^\mathsf{T} \\ u_k &= \begin{bmatrix} F_x & \dot{\delta} \end{bmatrix}_k^\mathsf{T} \\ z_k &= \begin{bmatrix} F_{y_f} & F_{y_r} & F_{e_f} & F_{e_r} \end{bmatrix}_k^\mathsf{T} \end{aligned}\right\} \forall\ k = 0, \ldots, N. \quad (21)$$

The state $\bar{x}$ and input $\bar{u}$ for the remaining stages l in the planning horizon with the point-mass model are defined as

$$\left.\begin{aligned} \bar{x}_l &= \begin{bmatrix} \bar{V} & \bar{t} & \bar{e} & \bar{\psi} \end{bmatrix}_l^\mathsf{T} \\ \bar{u}_l &= \begin{bmatrix} \bar{F}_x & \bar{F}_y \end{bmatrix}_l^\mathsf{T} \end{aligned}\right\} \forall\ l = 0, \ldots, M. \quad (22)$$

The dynamics of the vehicle's position states, presented in (1) and (17), are only marginally stable. Hence, continuous control of the vehicle is required, especially when the vehicle is to be pushed to its limits in racing or an emergency maneuver. Therefore, in this application of MPC, the control horizon is selected to be as long as the state prediction horizon.

B. NLP Feasibility

This NLP is purposefully designed with few hard inequality constraints. These include the steering system limits, engine power limit, and force limits for the point-mass model. Inputs can always be selected to respect these limits. Equality constraints are only present in the state transition models and for the tire model in (14) to relate the vehicle states and inputs to the additional optimization variables in the helper variable.

In contrast, aspects, such as deviations from the desired terminal state and road boundary intrusion, are carefully implemented with high costs, rather than constraints. In this way, the NLP is always numerically feasible, even when for example road boundary intrusion is physically unavoidable, the vehicle's current position is reported to be ever so slightly outside of the allowable road surface due to sensor noise or a minor road boundary intrusion occurred due to an external disturbance. Numerical feasibility is essential to be able to persistently solve for the control actions that the vehicle should take. With high costs on undesirable behavior, the solver will, in fact, yield the optimal action to take to avoid such behavior, using the vehicle's full potential if necessary.

C. NLP Discretization

One option for model discretization is to discretize in time, which leaves the curvilinear coordinate s as an optimization variable. Brown and Gerdes [20] took this temporal discretization approach in a control architecture with NMPC for obstacle avoidance. An advantage of this method is that the discretization time step can be selected based on the bandwidth of the plant dynamics. However, the path geometry needs to be recomputed (e.g., interpolated from a lookup table or modeled as a function of s) for every intermediate iteration of the NLP solver. This has nontrivial implications for the implementation and the performance of the optimization routine.

Alternatively, by discretizing the model spatially, the path geometry becomes constant for each stage in the planning horizon. This approach is more suitable for a racing scenario on a road with complicated topography, during which the vehicle will never plan to come to a stop. Formulating the dynamics spatially also allows for explicitly minimizing time over the fixed horizon distance. A disadvantage of spatial dynamics is that these become stiffer with lower vehicle speed, which changes during the optimization process, and hence, the spatial discretization step must be selected carefully.

In this work, the single-track model is discretized spatially along the road descriptor path based on the current vehicle speed at 30-ms intervals, for a total of N steps. For planning farther ahead, the current speed is no longer representative, as the controller will plan to accelerate and decelerate significantly over a long horizon. Therefore, the point-mass model is discretized spatially based on the speed profile that can be expected along the road descriptor path. For highway driving, this anticipated speed profile can, for example, be based on the local speed limit or the speed of the traffic ahead; for racing, Kapania *et al.* [21] computed the minimum-time speed profile for a given friction limit and path geometry. In this article, the point-mass model is spatially discretized for a total of M steps at 250-ms intervals along such a "racing" speed profile.

D. NLP Objective Function Terms

The NLP objective function in (20) is the sum of seven terms.

1) *Final Time:* The primary objective is to minimize the time at which the point-mass model reaches its last stage M of the planning horizon, denoted by $\bar{t}_M$. All other terms in the objective function are expressed as time penalties.

2) *Terminal State:* The final stage in the finite horizon must be controlled to a safe terminal state. In this case, the static road descriptor is selected for this safe state. The combination of terminal lateral position $\bar{e}_M$, course error $\bar{\phi}_M$, and excess speed $\bar{V}_{\text{exc}}$ is penalized, each term with its own weight W

$$J_{\text{term}} = W_{e_{\text{term}}} \bar{e}_M^{\,2} + W_{\phi_{\text{term}}} \bar{\phi}_M^{\,2} + W_{V_{\text{term}}} \bar{V}_{\text{exc}}^{\,2} \quad (23)$$

where

$$\bar{V}_{\text{exc}}^{\,2} = \begin{cases} \left(\bar{V}_M - \bar{V}_{\text{safe}_M}\right)^2, & \text{if } \bar{V}_M \geq \bar{V}_{\text{safe}_M} \\ 0, & \text{otherwise.} \end{cases} \quad (24)$$

3) *Road Boundary Intrusion:* To strongly encourage the vehicle to stay on the road, the problem is configured with a high conditional cost on intrusion of the road boundaries

$$\begin{aligned} J_{\text{rb}} = W_{\text{rb}} \sum_{k=1}^{N} \Delta s_k & \begin{cases} \left(e_k - e_k^{\max}\right)^2, & \text{if } e_k \geq e_k^{\max} \\ \left(e_k - e_k^{\min}\right)^2, & \text{if } e_k \leq e_k^{\min} \\ 0, & \text{otherwise.} \end{cases} \\ + W_{\text{rb}} \sum_{l=1}^{M} \Delta \bar{s}_l & \begin{cases} \left(\bar{e}_l - \bar{e}_l^{\max}\right)^2, & \text{if } \bar{e}_l \geq \bar{e}_l^{\max} \\ \left(\bar{e}_l - \bar{e}_l^{\min}\right)^2, & \text{if } \bar{e}_l \leq \bar{e}_l^{\min} \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (25)$$

For consistency of the objective function throughout the planning horizon, the intrusions are weighted by their respective discretization step Δs.

4) *Deviation from the Road Descriptor Path:* A small cost on planned lateral position yields feedback to the road descriptor

$$J_{\mathrm{rd}} = W_{\mathrm{rd}}\left(\sum_{k=1}^{N} \Delta s_k e_k^2 + \sum_{l=1}^{M} \Delta \bar{s}_l \bar{e}_l^2\right). \tag{26}$$

This provides robustness against plant model mismatch.

5) *Slew Rate:* Small costs on slew rates promote smoothness of the control inputs

$$J_{\dot{u}} = J_{\dot{\delta}} + J_{\Delta F_y} + J_{\Delta F_x} + J_{\Delta F_{\mathrm{tr}}} \tag{27}$$

where $J_{\dot{\delta}}$ penalizes steering rates in the single-track model

$$J_{\dot{\delta}} = W_{\dot{\delta}} \sum_{k=0}^{N} \dot{\delta}_k^2. \tag{28}$$

The point-mass model does not have steering as an input; there, the changes in total lateral force are penalized

$$J_{\Delta F_y} = W_{\Delta F} \sum_{l=1}^{M-1} \frac{1}{\Delta \bar{s}_l}\left(\bar{F}_{y_{l+1}} - \bar{F}_{y_l}\right)^2. \tag{29}$$

For both models, changes in the total commanded longitudinal force are included in the cost function

$$\begin{aligned} J_{\Delta F_x} = {} & W_{\Delta F} \sum_{k=0}^{N-1} \frac{1}{\Delta s_k}\left(F_{x_{k+1}} - F_{x_k}\right)^2 \\ & + W_{\Delta F} \sum_{l=1}^{M-1} \frac{1}{\Delta \bar{s}_l}\left(\bar{F}_{x_{l+1}} - \bar{F}_{x_l}\right)^2. \end{aligned} \tag{30}$$

The changes in forces at the model transition are included

$$J_{\Delta F_{\mathrm{tr}}} = \frac{W_{\Delta F}}{\Delta \bar{s}_0}\left(\left(\bar{F}_{x_1} - F_{x_N}\right)^2 + \left(\bar{F}_{y_1} - \tilde{F}_{y_{f_N}} - \tilde{F}_{y_{r_N}}\right)^2\right) \tag{31}$$

for continuity in the cascaded model planning horizon.

6) *Excessive Slip Angle*: A penalty on excessive slip angle prevents the tires from operating in the fully saturated region and stabilizes the vehicle. Furthermore, this term puts a cost on slip angles specifically beyond the point where the tire model is modified [see (14)], to discourage the solver from exploiting this unrealistic behavior of the modified tire model

$$\begin{aligned} J_\alpha = {} & W_\alpha \sum_{k=1}^{N} \begin{cases} \left(\left|\mathrm{t}\alpha_{f_k}\right| - \mathrm{t}\alpha_{f_k}^{\mathrm{mod}}\right)^2, & \text{if } \left|\mathrm{t}\alpha_{f_k}\right| \geq \mathrm{t}\alpha_{f_k}^{\mathrm{mod}} \\ 0, & \text{otherwise.} \end{cases} \\ & + W_\alpha \sum_{k=1}^{N} \begin{cases} \left(\left|\mathrm{t}\alpha_{r_k}\right| - \mathrm{t}\alpha_{r_k}^{\mathrm{mod}}\right)^2, & \text{if } \left|\mathrm{t}\alpha_{r_k}\right| \geq \mathrm{t}\alpha_{r_k}^{\mathrm{mod}} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{32}$$

where t denotes the tangent function.

7) *Excessive Force Usage Beyond the Imposed Limit:* The slack variables in the soft friction constraints are penalized

$$J_F = W_F \sum_{k=0}^{N} \left(\left(F_{e_{f\,k}}^{\,2}\right)^2 + \left(F_{e_{r\,k}}^{\,2}\right)^2\right). \tag{33}$$

E. NLP Constraints

1) *State Limits:* The tire model and point-mass model experience a singularity when $U_x = 0$ m/s and $\bar{V} = 0$ m/s, respectively, and with spatial discretization, the time between steps becomes infinite when the car comes to a stop. Therefore, in this racing application, U_x and $\bar{V}$ are lower bounded

$$U_{x_k} \geq U_x^{\min} \quad \forall\, k = 1, \ldots, N \tag{34a}$$

$$\bar{V}_l \geq \bar{V}^{\min} \quad \forall\, l = 1, \ldots, M. \tag{34b}$$

Despite these lower bounds on the longitudinal speed over the planning horizon and initial vehicle state, the research vehicle takes off reliably from a standstill. For more general applications, we successfully experimented with transitioning to kinematic vehicle models (that do not suffer from such singularities) at low speeds, but this is beyond the scope of this article.

The maximum steering angle is dictated by the vehicle's mechanical steering limits

$$\delta^{\min} \leq \delta_k \leq \delta^{\max} \quad \forall\, k = 1, \ldots, N. \tag{35}$$

2) *Input Limits:* The single-track model has a steering rate limit

$$\dot{\delta}^{\min} \leq \dot{\delta}_k \leq \dot{\delta}^{\max} \quad \forall\, k = 0, \ldots, N \tag{36}$$

and for both vehicle models, the maximum engine power P_{eng} is incorporated with inequality constraints

$$F_{x_k} \leq \frac{P_{\mathrm{eng}}}{U_{x_k}} \quad \forall\, k = 0, \ldots, N \tag{37a}$$

$$\bar{F}_{x_l} \leq \frac{P_{\mathrm{eng}}}{\bar{V}_l} \quad \forall\, l = 0, \ldots, M. \tag{37b}$$

3) *Tire Model:* The lateral axle forces are computed with the modified coupled tire model in (14)

$$\tilde{F}_{y_{f_k}} = \tilde{f}_{\mathrm{tire}}\left(\alpha_{f_k}, \tilde{F}_{x_{f_k}}, F_{z_{f_k}}\right) \quad \forall\, k = 0, \ldots, N \tag{38a}$$

$$\tilde{F}_{y_{r_k}} = \tilde{f}_{\mathrm{tire}}\left(\alpha_{r_k}, \tilde{F}_{x_{r_k}}, F_{z_{r_k}}\right). \qquad \vdots \tag{38b}$$

The longitudinal forces are constrained to the friction potential based on the instantaneous axle normal loads. Furthermore, we recognize that when an isotropic tire generates a force opposite to the direction of slip and the tires could lock but not experience excessive positive longitudinal slip, both bounds on the longitudinal tire forces must be corrected with the cosine of the slip angle

$$-\mu_f F_{z_{f_k}} \cos\alpha_{f_k} \leq \tilde{F}_{x_{f_k}} \leq \mu_f F_{z_f} \cos\alpha_{f_k} \quad \forall\, k = 0, \ldots, N \tag{39a}$$

$$-\mu_r F_{z_{r_k}} \cos\alpha_{r_k} \leq \tilde{F}_{x_{r_k}} \leq \mu_r F_{z_r} \cos\alpha_{r_k}. \qquad \vdots \tag{39b}$$

Fig. 7 illustrates this.

4) *Dynamics:* The initial state of the single-track model is equal to the measured vehicle state

$$x_0 = x_{\mathrm{meas}} \tag{40}$$

where the initial steering angle command is obtained from the previous controller time step. The initial longitudinal speed is set to at least $U_x^{\min} = \bar{V}^{\min} = 5$ m/s, even when the vehicle is actually traveling more slowly, to ensure that the initial state respects the state limits and a feasible solution to the NLP exists.

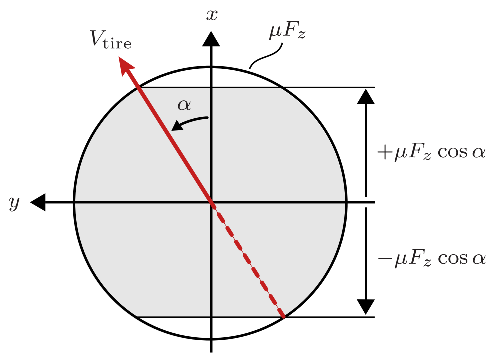

Fig. 7. An isotropic tire generates a force opposite to the direction of its velocity V_{tire} with respect to the road. The longitudinal force that the tire can generate, if longitudinal slip is limited, is bounded by the cosine of its slip angle α.

Let ν' denote the derivative of generic variable ν with respect to the curvilinear coordinate s, i.e., $\nu' = d\nu/ds$. Then, the following expressions describe the spatial dynamics g_{st} for the dynamic single-track model from (1):

$$U_x' = \frac{1}{\dot{s}}\left(\frac{\tilde{F}_{x_f}\cos\delta - \tilde{F}_{y_f}\sin\delta + \tilde{F}_{x_r} - F_d}{m} + rU_y\right) \quad (41a)$$

$$U_y' = \frac{1}{\dot{s}}\left(\frac{\tilde{F}_{y_f}\cos\delta + \tilde{F}_{x_f}\sin\delta + \tilde{F}_{y_r} + F_b}{m} - rU_x\right) \quad (41b)$$

$$r' = \frac{1}{\dot{s}}\left(\frac{a\left(\tilde{F}_{y_f}\cos\delta + \tilde{F}_{x_f}\sin\delta\right) - b\tilde{F}_{y_r}}{I_{zz}}\right) \quad (41c)$$

$$t' = \frac{1}{\dot{s}} \quad (41d)$$

$$e' = (1-\kappa e)\tan\left(\arctan\left(\frac{U_y}{U_x}\right) + \Delta\psi\right) \quad (41e)$$

$$\Delta\psi' = \frac{r}{\dot{s}} - \kappa \quad (41f)$$

$$\delta' = \frac{1}{\dot{s}}\dot{\delta}. \quad (41g)$$

These dynamics are integrated with the trapezoidal rule, with zero-order hold for the inputs ($\dot{\delta}$ and F_x) and first-order hold for the path geometry.

The spatial dynamics of the point-mass model g_{pm} are

$$\bar{V}' = \frac{1}{\dot{\bar{s}}}\frac{\left(\bar{F}_x - \bar{F}_d\right)}{m} \quad (42a)$$

$$\bar{t}' = \frac{1}{\dot{\bar{s}}} \quad (42b)$$

$$\bar{e}' = (1-\kappa\bar{e})\tan\bar{\phi} \quad (42c)$$

$$\bar{\phi}' = \frac{1}{\dot{\bar{s}}}\frac{\left(\bar{F}_y + \bar{F}_b\right)}{m\bar{V}} - \kappa. \quad (42d)$$

These dynamics are integrated with the simple forward Euler method, also with zero-order hold for the inputs ($\bar{F}_x$ and $\bar{F}_y$) and first-order hold for the path geometry. Mapping of the final state of the single-track model to the initial state of the point-mass model g_{tr} is described by (19).

5) *Friction Limits:* For conducting real-world experiments, friction limits with a user-defined friction coefficient allow for gradually ramping up the levels of acceleration at which the research vehicle is operating

$$\tilde{F}_{x_{f_k}}^{\;2} + \tilde{F}_{y_{f_k}}^{\;2} \le \left(\mu_{\text{lim}} F_{z_{f_k}}\right)^2 + F_{e_{f_k}}^{\;2} \quad \forall\, k = 0, \ldots, N \quad (43a)$$

$$\tilde{F}_{x_{r_k}}^{\;2} + \tilde{F}_{y_{r_k}}^{\;2} \le \left(\mu_{\text{lim}} F_{z_{r_k}}\right)^2 + F_{e_{r_k}}^{\;2}. \qquad \vdots \quad (43b)$$

The normal load models include the effects of steady-state longitudinal load transfer and effects of road topography as in (15), the lateral axle forces follow from (14), and the longitudinal axle forces are computed with the differentiable functions in (7). We reiterate that the soft friction constraints in (43) merely provide a means to bound the accelerations for real-world testing, if so desired. For the single-track model, the framework can never plan with forces that exceed those that can be realized with its tire models.

Similarly, the input forces to the point-mass model are constrained by the minimum of the user-defined friction limit and the tire-road friction coefficients. These constraints leverage steady-state assumptions for the lateral axle forces and are shown in Fig. 6

$$\bar{F}_{x_{f_l}}^{\;2} + \left(\tfrac{b}{L}\bar{F}_{y_l}\right)^2 \le \left(\min\left(\mu_f, \mu_{\text{lim}}\right)\bar{F}_{z_{f_l}}\right)^2 \;\forall\, l = 0, \ldots, M \quad (44a)$$

$$\bar{F}_{x_{r_l}}^{\;2} + \left(\tfrac{a}{L}\bar{F}_{y_l}\right)^2 \le \left(\min(\mu_r, \mu_{\text{lim}})\bar{F}_{z_{r_l}}\right)^2. \qquad \vdots \quad (44b)$$

F. Stability Analysis

For multivariable optimization of nonlinear systems, it is generally difficult to analytically investigate the system's stability characteristics as a function of the various weights in the objective function. In order to leverage analysis tools that are commonplace in classical control design, here the behavior of the NMPC framework is cast into a linear quadratic regulator (LQR) design. We consider the following equations of motion for the vehicle's yaw dynamics at constant longitudinal speed, linearized around equilibria, with steering rate as the input [22]:

$$\begin{bmatrix} \dot{\tilde{U}}_y \\ \dot{\tilde{r}} \\ \dot{\tilde{\delta}} \end{bmatrix} = \begin{bmatrix} \dfrac{-\tilde{C}_{\alpha_f} - \tilde{C}_{\alpha_r}}{mU_x} & \dfrac{b\tilde{C}_{\alpha_r} - a\tilde{C}_{\alpha_f}}{mU_x} - U_x & \dfrac{\tilde{C}_{\alpha_f}}{m} \\ \dfrac{b\tilde{C}_{\alpha_r} - a\tilde{C}_{\alpha_f}}{I_{zz}U_x} & \dfrac{a^2\tilde{C}_{\alpha_r} - b^2\tilde{C}_{\alpha_f}}{I_{zz}U_x} & \dfrac{a\tilde{C}_{\alpha_f}}{I_{zz}} \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} \tilde{U}_y \\ \tilde{r} \\ \tilde{\delta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\dot{\tilde{\delta}}. \quad (45)$$

The NMPC objective function in (20) is approximated by two terms that encode its balancing of competing objectives in an LQR design: the deviation of front slip angle, to reflect the incentive to maintain maximum front lateral in order to minimize time around a turn, and the steering rate, to obtain smooth control inputs. This can directly be cast into the quadratic cost function for an LQR design of the standard form

$$J = \int_0^\infty \left(x^T Q x + u^T R u\right)dt = \int_0^\infty \left(\alpha_f^2 + \tilde{W}_{\dot{\delta}}\dot{\delta}^2\right)dt \quad (46)$$

Fig. 8. Automated Volkswagen Golf GTI. Photograph by Patrick Beaudouin.

with a small angle approximation of the front slip angle

$$\alpha_f \approx \frac{U_y + ar}{U_x} - \delta. \tag{47}$$

For the quadratic LQR cost function, this yields

$$Q = \begin{bmatrix} \frac{1}{U_x^2} & \frac{a}{U_x^2} & -\frac{1}{U_x} \\ \frac{a}{U_x^2} & \frac{a^2}{U_x^2} & -\frac{a}{U_x} \\ -\frac{1}{U_x} & -\frac{a}{U_x} & 1 \end{bmatrix} \quad \text{and} \quad R = \tilde{W}_{\dot{\delta}}. \tag{48}$$

By varying the value of $\tilde{W}_{\dot{\delta}}$, the relative importance of the steering rate in the objective function is directly altered, which leads to a different optimal gain matrix in the LQR design. Fig. 9 shows the resulting eigenvalues of the closed-loop linear system for two different values of $\tilde{W}_{\dot{\delta}}$ when linearizing the system between 0% and 100% front slip angle saturation.

The analysis shows that in all driving, the closed-loop poles are stable. At 100% front slip angle saturation, the imaginary poles all end up in the same location. This is as expected since, at the sliding slip angle, changes in slip angle no longer lead to a change in lateral force (see Section III-A3). Thus, at this point, steering inputs have become ineffective in control of the yaw dynamics.[1] Below the absolute limits, however, the root loci show that the damping of these closed-loop dynamics changes as a function of the weight on the steering rate in the LQR objective function.

At first sight, the conclusion that the system's performance (damping) decreases with a lower weight on steering rate seemed counterintuitive. We expected that a lower weight would make the system more responsive in counteracting disturbances. Nonetheless, the analytical results are confirmed by experimental results: with a relatively low weight on steering rate in the NMPC objective function, the research vehicle displays oscillatory yaw behavior in sustained turns [22].

V. Experiment Design

The concept of cascaded model complexity is validated in two steps, with a simulation study and a real-world experiment.

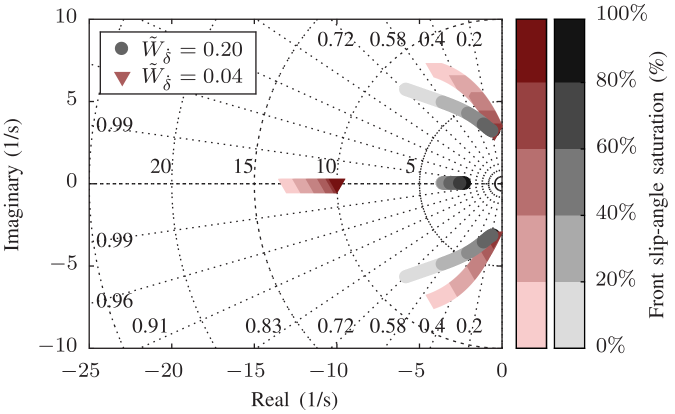

Fig. 9. Closed-loop poles for two different values of the relative weight on steering rate in an LQR framework, with the linearizations from 0% to 100% front slip angle saturation, for $U_x = 30$ m/s.

The studies feature the same optimization software and computer hardware and compare the cascaded model architecture with an architecture that features only the single-track model.

The optimization problem is implemented with CasADi 3.4.3, a free and open-source symbolic framework for automatic differentiation and optimal control [23]. The NLP is solved with IPOPT 3.12.0, a free and open-source software package for large-scale nonlinear optimization [15], using the free HSL MA27 linear solver [24].

An automated 2018 Volkswagen Golf VII GTI Performance, shown in Fig. 8, serves as the research platform for the experiments. Table I shows the main vehicle and tire parameters. The proposed control framework is deployed on an onboard computer, which has an Intel Core i7-5700EQ processor with a clock speed of 2.60 GHz. A dSpace MicroAutoBox II provides the low-level interfaces to the vehicle. A low-level steering controller running at 1 kHz tracks the desired steering angle by sending a torque command to an electronic power steering (EPS) system. Positive longitudinal force commands are converted into a throttle pedal voltage, based on an engine map that is created from measurements of the research vehicle on a dynamometer. Negative longitudinal force commands are mapped to front and rear brake pressure commands according to a static brake proportioning function. The vehicle is equipped with an antilock brake system (ABS), but no electronic stability control (ESC) or traction control (TC) is active. An Oxford Technical Solutions RT3002 provides vehicle state estimates from an integrated inertial measurement unit (IMU) and a single-antenna GPS receiver with differential corrections from an on-site GPS base station.

The simulation study and the real-world experiment are conducted on the West track at Thunderhill Raceway Park in Willows, CA, USA, for which the layout is shown in Fig. 10. This is a full-scale race track with a length of 2.6 km that exposes the vehicle to many different challenging driving scenarios. The track includes tight hairpin turns, high-speed sweeping turns, s-turns, banked turns, and a turn on top of a hill crest. The virtual road boundaries for the controller are placed at 1 m from the physical edges of the track, to create a safety margin for real-world testing near the limits of handling.

[1] This also implies that at the limits of handling, steering is no longer an effective input to control the vehicle's position states. Hence, in racing and emergency maneuvers, combined lateral and longitudinal control is crucial.

TABLE I

VEHICLE AND TIRE MODEL PARAMETERS (TWO PASSENGERS, WET ASPHALT)

Parameter	Symbol	Value	Unit
Vehicle mass	m	1,778	kg
Yaw moment of inertia	I_{zz}	3,049	kg·m^2
Distance front axle to CG	a	1.194	m
Distance rear axle to CG	b	1.436	m
Center of gravity height	h_{cg}	0.55	m
Maximum engine power	P_{eng}	172	kW
Rolling resistance	F_{rr}	218	N
Aerodynamic drag coefficient	C_D	0.4243	N/(m/s)^2
Front/rear drive force distr.	χ_{f_d}/χ_{r_d}	1 / 0	-
Front/rear brake force distr.	χ_{f_b}/χ_{r_b}	0.78 / 0.22	-
Steering angle limits	$\delta^{\text{min,max}}$	±27	deg
Steering rate limits	$\dot{\delta}^{\text{min,max}}$	±20	deg/s
Front/rear cornering stiffness	C_{α_f} / C_{α_r}	180 / 300	kN/rad
Front/rear friction coefficient	μ_f / μ_r	0.75 / 0.80	-

TABLE II

VALUES OF THE WEIGHTS IN THE NLP OBJECTIVE FUNCTION

Cost weighting term	Symbol	Value	Unit
Terminal lateral error	$W_{e_{\text{term}}}$	0.4	s/m^2
Terminal course error	$W_{\phi_{\text{term}}}$	1	s/rad^2
Terminal excessive speed	$W_{V_{\text{term}}}$	0.1	s/(m/s)^2
Road boundary intrusion	W_{rb}	1	s/m/m^2
Deviation from road descriptor path	W_{rd}	0.001	s/m/m^2
Slew rate (steering angle)	$W_{\dot{\delta}}$	0.0001	s/(deg/s)^2
Slew rate (input forces)	$W_{\Delta F}$	0.002	s·m/kN^2
Excessive slip angle	W_{α}	1,000	s
Soft friction constraint	W_F	0.0001	s/kN^4

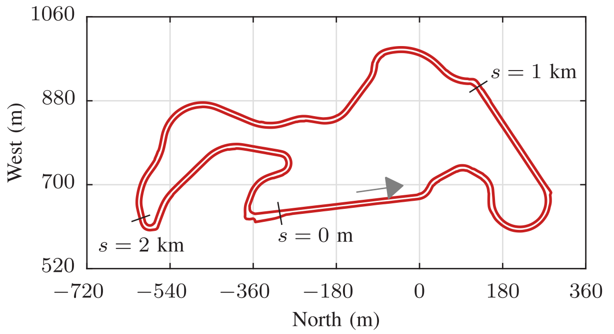

Fig. 10. Layout of the West race track at Thunderhill Raceway Park.

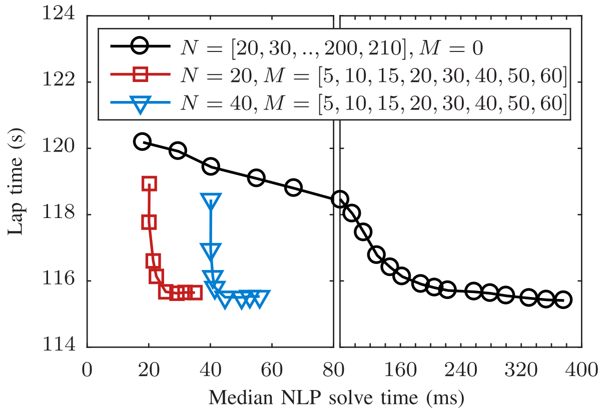

Fig. 11. Simulated lap time for $\mu_{\text{lim}} = 0.6$, plotted against the median NLP solve time, for different combinations of N steps with the single-track model followed by M steps with the point-mass model. Note the change in scaling of the horizontal axis.

Since the race track is very wide, often in excess of 8 m, the track centerline is not a realistic descriptor for the path that the vehicle will take: the controller will aim to use the full width of the track in order to minimize time. Therefore, the centerline is not representative of the topography that the vehicle will encounter, and its associated speed profile will not accurately reflect the vehicle's speed for model discretization. Instead, for this racing application, the descriptor trajectory is generated offline in a single NLP for a full lap around the track, using the point-mass model with a minimum-time objective. To reflect the fact that the point-mass model is not an accurate description of the vehicle's dynamics, the NLP is configured to leave a conservative minimum distance of 1.5 m to the road edges. The resulting trajectory, i.e., path and speed profile, provides the basis for the static map that is loaded to the vehicle. As such, computation of the road descriptor path or its speed profile is not part of the online motion planning and control architecture.

Table II lists the values for the weights in the NLP objective function. These are selected based on simulations, empirical tuning in real-world experiments with the automated research vehicle, and the stability analysis presented in Section IV-F. The tuning process focuses on obtaining good closed-loop stability and proper prioritization of control objectives. For example, the weights for excessive slip angles and road boundary intrusion are selected such that the cost on slip angle in a limit oversteer situation or an imminent road departure outweighs the cost for the steering slew rate required for recovery.

VI. SIMULATION STUDY

The simulation study serves two main purposes.

1) Demonstrate that a cascaded model architecture and an architecture with only a single-track model achieve the same lap time if their horizons are sufficiently long.
2) Illustrate that the horizon length for the single-track model to achieve the minimum lap time prohibits real-time control, but not in case of a cascaded model design.

To these ends, the simulation study evaluates different combinations of N planning stages with the single-track model followed by M stages with the point-mass model. The vehicle model in the simulation is the dynamic single-track model, as described in Section III-A.

Fig. 11 shows the time required to complete a full lap against the median NLP solve time. It illustrates that the performance of the architecture with only the single-track model improves when the number of planning stages is increased, but that the median solve time increases significantly.

In contrast, with the cascaded model design, the simulated lap time improves quickly as stages with the point-mass model are appended, while the increase in median solve time is modest. The planning horizon is still bound to the same

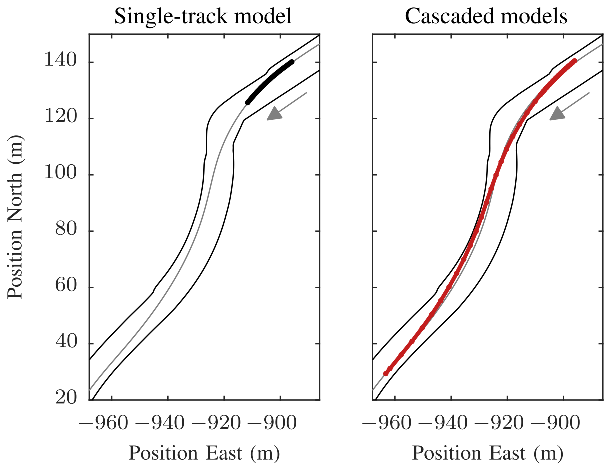

Fig. 12. Planned position states with the two different architectures, going into turn 3 with a nominal friction limit of 0.6. The road descriptor path is also included.

TABLE III

DIMENSIONS OF THE NLPS IN THE REAL-WORLD EXPERIMENT

	Single	Cascaded
Nonzeros in equality constraint Jacobian	2,932	2,373
Nonzeros in inequality constraint Jacobian	1,064	913
Nonzeros in Hessian of the Lagrangian	2,287	1,912
Number of variables	403	405
Number of equality constraints	279	277
Number of inequality constraints	310	298

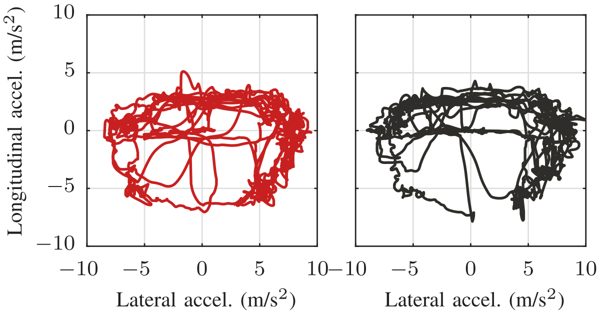

Fig. 13. Accelerations in runs where the nominal friction limit is set to 0.7, i.e., the fastest runs in the wet conditions.

(high) costs of deviations from the road descriptor trajectory at the end of the horizon, but because the planning horizon is significantly longer, there is more freedom to leverage the vehicle's potential and the width of the track more effectively in the near term.

Both architectures achieve the same minimum lap time, but the computational burden with only the single-track model is significantly larger. In fact, when targeting a replan time of 50 ms and achieving the minimum lap time, the solve time for this architecture prohibits real-time control. With the cascaded model design, on the other hand, the minimum lap time can be achieved well within a median NLP solve time of 50 ms.

Effectively, as the planning horizon is extended, the control framework transitions from predominantly trajectory tracking of the terminal state to a framework that approaches the globally optimal solution from the current state. Fig. 11 reveals that for this specific scenario, the practical limit is a planning horizon of approximately 7 s ($N = 20$ and $M = 30$ or $N = 200$); beyond that, the lap time does not decrease.

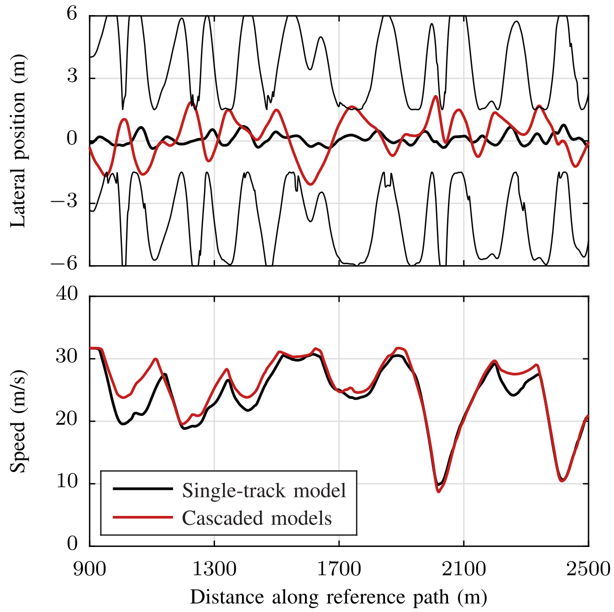

Fig. 14. Measured lateral position from the road descriptor path in between the virtual road edges (in black), and measured horizontal speed with the two different architectures, in runs where the nominal friction limit is set to 0.6.

VII. REAL-WORLD EXPERIMENT

For real-time implementation of the NMPC framework, a replan interval of 50 ms is targeted. With this requirement, the simulation study, and the notion that the solve times fluctuate, two architectures are selected for a benchmark experiment.

1) *Single-Track Model Only:* The vehicle's dynamics are propagated with only the single-track model, in $N = 32$ steps, making up a nominal horizon length of 0.96 s.
2) *Cascaded Models:* The first part of the horizon consists of $N = 22$ steps with the single-track model (nominally 0.66 s), followed by $M = 22$ steps with the point-mass model (nominally 5.5 s). The combined nominal planning horizon length is 6.16 s.

Fig. 12 shows how the planning horizon of the cascaded setup extends through two full turns of the test track, while the planning horizon with only the single-track model barely covers the first half of the first turn for this configuration of the number of stages. Table III lists the dimensions of the resulting NLPs. Despite the difference in the total number of stages between the two architectures, the number of variables and the number of constraints are comparable. The cascaded model design does have fewer nonzeros in the Hessian of the Lagrangian and the constraint Jacobian; this increase in sparsity is hypothesized to lead to a lower median solve time.

The experiments cover the first to the last turn of the Thunderhill West race track, from $s = 450–2575$ m. In the experiment, which coincidentally took place in very wet weather conditions, five levels of the nominal friction limit μ_{lim} are tested: 0.5, 0.6, 0.65, 0.68, and 0.7. In these weather

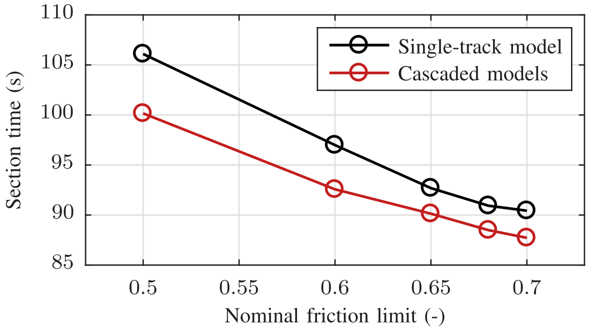

Fig. 15. Measured section time from $s = 450$–2,575 m, for all tested levels of the nominal friction limit.

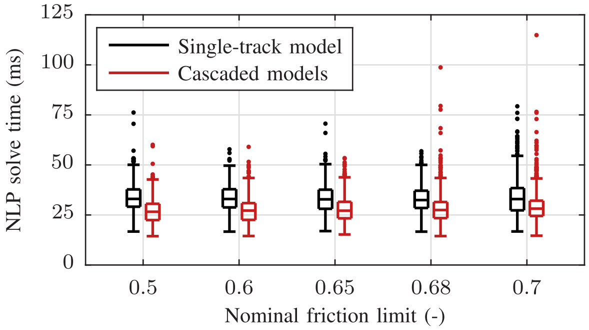

Fig. 16. Tukey boxplot of the wall-clock time of solving the NLPs with the two different NMPC architectures in all benchmark runs. The targeted replan time is 50 ms.

conditions, the vehicle is operating very close to its physical limits when the nominal friction limit is set to 0.7. Fig. 13 shows that in the fastest runs, the lateral and longitudinal accelerations are around 7 m/s^2 as expected and the minimum-time objective constantly pushes the vehicle to its limits.

Indeed, with a longer planning horizon, the controller can more effectively use the distance to the road edges, as shown in Fig. 14. It also illustrates how this allows the car to carry more speed through the turns. The ability to use the width of the track more effectively to minimize distance and increase cornering speeds has a significant effect on the closed-loop performance. Over a measurement distance of 2,125 m along the road descriptor path, the serially cascaded model design is between 1.4 and 5.9 s faster, as shown in Fig. 15.

Fig. 16 compares the computational burden of the two architectures in a Tukey boxplot and confirms the hypothesis that with the cascaded model architecture, the median NLP solve time is consistently reduced.

VIII. Conclusion

This article introduces the concept of cascaded model complexity within a single planning horizon for vehicle motion planning and control. The framework leverages the fact that lower complexity models require fewer optimization variables and constraints and that the slower dynamics of lower complexity models allow for a larger discretization step. This yields a significant extension of the horizon at low computational cost while maintaining the high quality of control with a high-fidelity model in the first part of the horizon. In simulations and real-world experiments with an automated race car, the proposed concept proves to be valuable, lowering both the lap time and the median solve time of the optimization problem.

This article considers hand-picked nonlinear vehicle models. Future work will explore *automatic* model reduction techniques for general linear and nonlinear systems and will seek other applications. For more advanced vehicle control, the framework could, for example, be extended with vehicle models for torque vectoring and rollover protection.

As an extension of the current work, it will be investigated if the horizon can be extended so far with the point-mass model, up to the real-time computation limits, that we can always plan to stop at the end of the horizon. This could eliminate the need for any offline computations of a safe speed profile for the road descriptor path.

Acknowledgment

The authors thank Volkswagen Group Research, in particular Dr. Björn Mennenga and Dr. Paul Hochrein, and the Electronics Research Laboratory, for providing the research vehicle and the members of the Dynamic Design Laboratory for their help with implementing and testing the framework.

References

[1] S. Gros, D. Buccieri, P. Mullhaupt, and D. Bonvin, “A two-time-scale control scheme for fast unconstrained systems,” in *Assessment Future Directions Nonlinear Model Predictive Control*. Berlin, Germany: Springer, 2007, pp. 551–563.

[2] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, “Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads,” in *Proc. ASME Dyn. Syst. Control Conf.*, vol. 1, Jan. 2010, pp. 265–272.

[3] T. Novi, A. Liniger, R. Capitani, and C. Annicchiarico, “Real-time control for at-limit handling driving on a predefined path,” *Vehicle Syst. Dyn.*, vol. 58, no. 7, pp. 1007–1036, Apr. 2019, doi: 10.1080/00423114.2019.1605081.

[4] A. Gray, Y. Gao, T. Lin, J. K. Hedrick, H. E. Tseng, and F. Borrelli, “Predictive control for agile semi-autonomous ground vehicles using motion primitives,” in *Proc. Amer. Control Conf. (ACC)*, Jun. 2012, pp. 4239–4244.

[5] Y. Wang and S. Boyd, “Fast model predictive control using online optimization,” *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 267–278, Mar. 2010.

[6] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, “Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation,” *Int. J. Robust Nonlinear Control*, vol. 18, no. 8, pp. 862–875, 2008.

[7] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes, “Collision avoidance and stabilization for autonomous vehicles in emergency scenarios,” *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 4, pp. 1204–1216, Jul. 2017.

[8] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, “A study on model fidelity for model predictive control-based obstacle avoidance in high-speed autonomous ground vehicles,” *Vehicle Syst. Dyn.*, vol. 54, no. 11, pp. 1629–1650, Nov. 2016.

[9] M. Metzler, D. Tavernini, A. Sorniotti, and P. Gruber, “An explicit nonlinear MPC approach to vehicle stability control,” in *Proc. 14th Int. Symp. Adv. Vehicle Control*, 2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9033968/references#references

[10] M. Canale, L. Fagiano, and V. Razza, “Approximate NMPC for vehicle stability: Design, implementation and SIL testing,” *Control Eng. Pract.*, vol. 18, no. 6, pp. 630–639, Jun. 2010.

[11] M. Diehl, H. G. Bock, and J. P. Schlöder, “A real-time iteration scheme for nonlinear optimization in optimal feedback control,” *SIAM J. Control Optim.*, vol. 43, no. 5, pp. 1714–1736, Jan. 2005.

[12] J. V. Frasch *et al.*, “An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles,” in *Proc. Eur. Control Conf. (ECC)*, Jul. 2013, pp. 4136–4141.

[13] 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 Trans. Control Syst. Technol.*, vol. 26, no. 6, pp. 1982–1994, Nov. 2018.

[14] *Road Vehicles—Vehicle Dynamics and Road-Holding Abilit—Vocabulary*, Standard ISO 8855, 2011.

[15] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, Mar. 2006.

[16] H. B. Pacejka, *Tyre Vehicle Dynamics*, 3rd ed. Oxford, U.K.: Butterworth-Heinemann, 2012.

[17] R. Y. Hindiyeh, "Dynamics and control of drifting in automobiles," Ph.D. dissertation, Dept. Mech. Eng., Stanford Univ., Stanford, CA, USA, 2013.

[18] C. Voser, R. Y. Hindiyeh, and J. C. Gerdes, "Analysis and control of high sideslip manoeuvres," *Vehicle Syst. Dyn.*, vol. 48, no. sup1, pp. 317–336, Dec. 2010.

[19] J. Subosits and J. C. Gerdes, "Autonomous vehicle control for emergency maneuvers: The effect of topography," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2015, pp. 1405–1410.

[20] M. Brown and J. C. Gerdes, "Coordinating tire forces to avoid obstacles using nonlinear model predictive control," *IEEE Trans. Intell. Vehicles*, vol. 5, no. 1, pp. 21–31, Mar. 2020.

[21] N. R. Kapania, J. Subosits, and J. Christian Gerdes, "A sequential two-step algorithm for fast generation of vehicle racing trajectories," *J. Dyn. Syst., Meas., Control*, vol. 138, no. 9, Sep. 2016, Art. no. 091005.

[22] V. A. Laurense, "Integrated motion planning and control for automated vehicles up to the limits of handling," Ph.D. dissertation, Dept. Mech. Eng., Stanford Univ., Stanford, CA, USA, 2019.

[23] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, Mar. 2019.

[24] HSL. (2014). *A Collection of Fortran Codes for Large Scale Scientific Computation*. [Online]. Available: http://www.hsl.rl.ac.uk/

Vincent A. Laurense (Member, IEEE) received the M.Sc. degree in aerospace engineering *(cum laude)* from the Delft University of Technology, Delft, The Netherlands, in October 2012, with a focus on modeling manual feedforward control. He is currently pursuing the Ph.D. degree with Stanford University, Stanford, CA, USA, with a focus on the topic of robust vehicle motion planning and control up to the limits of tire-road friction.

With an internship at the Nissan Advanced Technology Center, Atsugi, Japan, and the Honors Program at the Delft University of Technology, he expanded his curriculum with automotive dynamics and safety, providing an academic context to his role as a data engineer in Dutch motorsports teams. Subsequently, he worked on automotive human–machine interaction with Dr. Boer at Entropy Control Inc., La Jolla, CA, USA, and connected vehicles with TNO Automotive, Helmond, The Netherlands. His research interests include vehicle dynamics and control, advanced driver support systems, autonomous vehicles, and optimal control.

J. Christian Gerdes (Member, IEEE) received the Ph.D. degree from the University of California at Berkeley, Berkeley, CA, USA, in 1996.

He is currently a Professor in mechanical engineering with Stanford University, Stanford, CA, USA, where he is also the Director of the Center for Automotive Research. He is also a Co-Founder of Peloton Technology, Mountain View, CA, USA. His laboratory studies how cars move, how humans drive cars, and how to design future cars that work cooperatively with the driver or drive themselves. When not teaching on campus, he can often be found at the race track with students, instrumenting historic race cars, or trying out their latest prototypes for the future.

Dr. Gerdes and his team have been recognized with several awards, including the Presidential Early Career Award for Scientists and Engineers, the Ralph Teetor Award from SAE International, and the Rudolf Kalman Award from the American Society of Mechanical Engineers.