Collision Avoidance and Stabilization for Autonomous Vehicles in Emergency Scenarios

Joseph Funke, Matthew Brown, Stephen M. Erlien, and J. Christian Gerdes

Abstract—Emergency scenarios may necessitate autonomous vehicle maneuvers up to their handling limits in order to avoid collisions. In these scenarios, vehicle stabilization becomes important to ensure that the vehicle does not lose control. However, stabilization actions may conflict with those necessary for collision avoidance, potentially leading to a collision. This paper presents a new control structure that integrates path tracking, vehicle stabilization, and collision avoidance and mediates among these sometimes conflicting objectives by prioritizing collision avoidance. It can even temporarily violate vehicle stabilization criteria if needed to avoid a collision. The framework is implemented using model predictive and feedback controllers. Incorporating tire nonlinearities into the model allows the controller to use all of the vehicle's performance capability to meet the objectives. A prediction horizon comprised of variable length time steps integrates the different time scales associated with stabilization and collision avoidance. Experimental data from an autonomous vehicle demonstrate the controller safely driving at the vehicle's handling limits and avoiding an obstacle suddenly introduced in the middle of a turn.

Index Terms—Autonomous vehicles, collision avoidance, model predictive control, vehicle control, vehicle stability.

I. Introduction

AS AUTONOMOUS vehicles leave the research laboratory and enter public roads, they must react to emergency scenarios, some of which may necessitate maneuvering up to the vehicle's handling limits in order to avoid a collision. Vehicle stabilization becomes increasingly important when operating near these handling limits, as it modifies inputs to the vehicle to ensure that it does not lose control. Yet, enforcing stabilization criteria does not necessarily assist collision avoidance and may conflict with the demands of the desired trajectory.

Autonomous vehicle control is commonly divided into trajectory generation and trajectory following. Trajectory generation algorithms, such as those used in the top DARPA Urban Challenge finishers Boss and Junior, often assumed vehicle operation below the handling limits, which facilitates using simple trajectory following algorithms [1], [2]. Similar assumptions continue to be incorporated in more recent research with Junior and other vehicles such as Bertha [3], [4]. In an emergency situation, these approaches rely on built-in production stabilization systems such as Electronic Stability Control (ESC) [5], which enforce stabilization criteria regardless of an impending collision. Thus, while these approaches may generate a collision free trajectory, it becomes unclear whether the control actions modified by the underlying ESC systems lead to a collision-free trajectory for the actual vehicle.

Alternative approaches have explicitly incorporated stabilization into trajectory tracking using model predictive control (MPC) techniques. MPC provides a framework that can replace a simple path tracking control law with a continuously resolved optimization seeking to minimize path tracking errors within constraints such as stabilization criteria. Falcone *et al.* [6] tracked a lane change trajectory on an icy road by limiting vehicle states to a linear regime of tire force generation. Katriniok *et al.* [7] similarly tracked a lane change trajectory, but extended the stabilization criteria to incorporate nonlinear vehicle dynamics as well, increasing allowable control actions to track a trajectory. Such MPC trajectory tracking algorithms can be paired with trajectory generation algorithms to offer a complete guidance system. Falcone *et al.* [8] and Gao *et al.* [9] suggest hierarchical frameworks, in which a high-level MPC algorithm communicates obstacle avoidance trajectories to a low-level MPC algorithm focused on trajectory tracking.

While these approaches directly incorporate stabilization rather than relying on production ESC systems, they still suffer from the underlying challenge that stabilization constraints may lead to tracking errors that could result in a collision. The impact of stabilization control actions is thus relevant to both the trajectory following and trajectory generation processes. To better capture these impacts of stabilization, trajectory generation and tracking can be combined into one control problem. Adding collision constraints to the MPC algorithm allows the controller to modify control inputs in order to avoid a collision. Liniger *et al.* [10] demonstrated time optimal trajectory generation and tracking around a race track with other vehicles, without explicit stabilization constraints, for remote control vehicle racing. Gao *et al.* [11] and Turri *et al.* [12] demonstrated obstacle avoidance on a straight icy road using full-sized vehicles and stability bounds designed to limit vehicle states to the linear regime of tire force generation. However, such bounds may be overly restrictive in emergencies that require all of the tires' force generation capability to avoid a collision.

We similarly introduce an integrated trajectory generation and tracking approach, but with two improvements. First,

stabilization limits are extended into the nonlinear operating regime of the vehicle, allowing vehicle operation up to the handling limits. Incorporation of nonlinear dynamics is possible with a careful linearization, regulation, and trust region modeling approach of rear tire force generation. Second, varied length time steps in the prediction horizon of the MPC optimization enable better integration of stabilization and collision avoidance in a single problem. Short time steps model immediate vehicle stabilization and tracking decisions, while long time steps later in the horizon incorporate the future implications of those actions for collision avoidance. These improvements allow a fundamental shift in how stabilization is incorporated into autonomous vehicle control. Instead of prioritizing stabilization highest, either implicitly with ESC systems or explicitly with stabilization constraints, this approach prioritizes collision avoidance above stabilization. This allows the system to temporarily violate stabilization criteria, if necessary, in order to avoid a collision. These temporary violations are planned along with subsequent control inputs throughout the prediction horizon that can return vehicle states to within the desired bounds.

The resulting approach builds off of Erlien *et al.*'s [13] human-shared controller and Brown *et al.*'s [14] subsequent path tracking controller with an autonomous trajectory tracking controller that follows desired speeds while deviating from desired paths to best meet collision avoidance and stabilization criteria. This paper extends initial results [15] with a closer look at model linearization, the variable time step prediction horizon, and prioritization of objectives and with a more illustrative set of experiments. Three experiments demonstrate controller performance up to the handling limits of a real vehicle. Unlike the previous work considering transient lane change maneuvers on straight roads, these experiments test on curved paths with steady-state operation near the handling limits, the first for MPC control to the best of our knowledge. The first test is obstacle free to demonstrate the controller driving safely and smoothly near the limits. The second test introduces a known obstacle in the middle of the turn, showcasing the controller's ability to account for this future information. In the third test, an obstacle suddenly appears mid-turn, triggering the controller's immediate response and temporary stability constraint violation to avoid a collision. This last example demonstrates that collision avoidance can sometimes be aided by temporarily violating stability criteria, which is a fundamental departure from the traditional approaches. Section II introduces the overall structure of the controller and how Sections III–VII relate to that structure, and Section VIII provides these experimental results.

II. OVERVIEW OF OPERATION

The controller determines steering commands that best mediate among sometimes conflicting demands to track a nominal path, ensure vehicle stability, and avoid collisions. To effectively tradeoff among these objectives, current steering decisions must be informed by the upcoming environment and estimates of future vehicle dynamics. A prediction horizon incorporates this future information at discrete times into the future. Fig. 1 shows an example of a vehicle's current position,
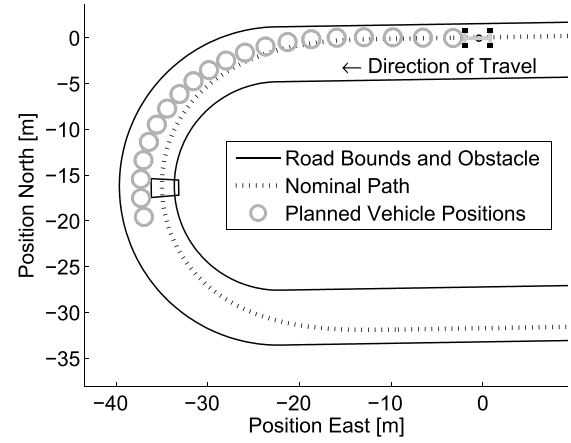

Fig. 1. Vehicle positions over a prediction horizon. The vehicle is currently positioned at (0, 0), and the controller determines planned vehicle positions for each future discrete step in the prediction horizon, here deviating from the path to avoid the upcoming obstacle.

an obstacle in its path, and its plan around the obstacle through this prediction horizon.

A model of the vehicle estimates how steering inputs translate to future vehicle states pertaining to path tracking, stabilization, and collision avoidance at each point in the prediction horizon. With this model, an MPC approach calculates vehicle inputs throughout the prediction horizon that optimize objectives subject to this model and constraints, such as stabilization and avoidance. To achieve real-time responses, formulating the problem with a quadratic objective function and time-varying affine constraints enables the use of fast quadratic program solvers. The resulting generic MPC problem is as follows:

$$\min_u \sum_{k=0}^{n-1}(x^{k^T}Q^k x^k + u^{k^T}R^k u^k) + x^{n^T}Q^n x^n \qquad (1)$$

subject to

$$x^{k+1} = A^k x^k + B^k u^k + C^k \quad k = 0\ldots n-1 \qquad (2)$$

$$W^k u^k \leq Z^k \quad k = 0\ldots n-1 \qquad (3)$$

$$H^k x^k \leq G^k \quad k = 1\ldots n \qquad (4)$$

where vehicle states x and vehicle input u are defined at each discrete time step k in the prediction horizon. The solution to this optimization is defined as u^*, which by (2) defines associated vehicle states x^*. Every execution step, which occurs at 10-ms intervals, the controller resolves this optimization and applies u^{*0} to the vehicle, each time with a horizon shifted 10 ms to reflect the time that has passed since the last execution step. This approach is known as receding horizon control.

Sections III–VII describe the formulation of the above problem in detail. Section III introduces the structure of a desired trajectory, which is subsequently incorporated into the vehicle state x to enable path tracking. Section IV discusses the linearization of nonlinear vehicle dynamics into an affine time-varying model such as (2). Linearizations rely on the estimates of expected states, $\bar{x}$, which derive from either longitudinal control or the previous execution step's solution u^*_{prev}

and resulting states x^*_{prev}. Using the previous solution to aid in linearization of the next leads to a successive linearization approach to modeling nonlinear dynamics, as previously incorporated in both online [6] and offline [16] optimizations. Trust regions on the linearizations prevent the linearizations from becoming excessively inaccurate, leading to constraints on allowable states (4). Collision avoidance and stabilization criteria impose further constraints on x through (4), as discussed in Section V. Section VI introduces variable length time steps, which enable unequal sample times for the n points in the horizon to best include information necessary for both stabilization and collision avoidance. Section VII defines the final MPC problem, discusses how prioritizations are established, and introduces a simple longitudinal controller.

III. TRAJECTORY

Trajectories define the path and speed for the autonomous vehicle to follow in the absence of stability and environment constraints. This controller incorporates trajectories defined in terms of a path and speed profile.

A. Path

The path provides a nominal route for the vehicle to follow and a reference for the vehicle model and environmental bounds. Paths are parameterized by curvature K as a function of distance s along the path. Given a starting East/North position (E_0, N_0) and heading orientation ψ_0, the position and tangent direction along the path can be calculated as a function of curvature and path distance

$$\psi(s) = \int_0^s K(x)dx + \psi_0 \tag{5}$$

$$E(s) = \int_0^s \cos \psi(x)dx + E_0 \tag{6}$$

$$N(s) = \int_0^s \sin \psi(x)dx + N_0. \tag{7}$$

B. Speed Profile

The speed profile defines the desired vehicle longitudinal velocity $U_{x,\text{des}}$ at each point s along the path. Tradeoffs between collision avoidance and stabilization become most apparent when tracking a path near the handling limits of the vehicle; a speed profile can be defined in terms of these limits.

Handling limits are derived from the available road-tire friction μ, which limits a vehicle's lateral acceleration a_y and longitudinal acceleration a_x to

$$\sqrt{a_x^2 + a_y^2} \leq \mu g \tag{8}$$

where g is the gravitational constant. Lateral acceleration relates to path curvature K using a steady-state assumption

$$a_y \approx U_x^2 K \tag{9}$$

limiting the speed along a path. Given a desired friction usage μ_{des}, the previous work presents an algorithm based on (8) and (9) to determine speeds along the path that yield the desired vehicle accelerations of $\mu_{\text{des}}g$ [17].
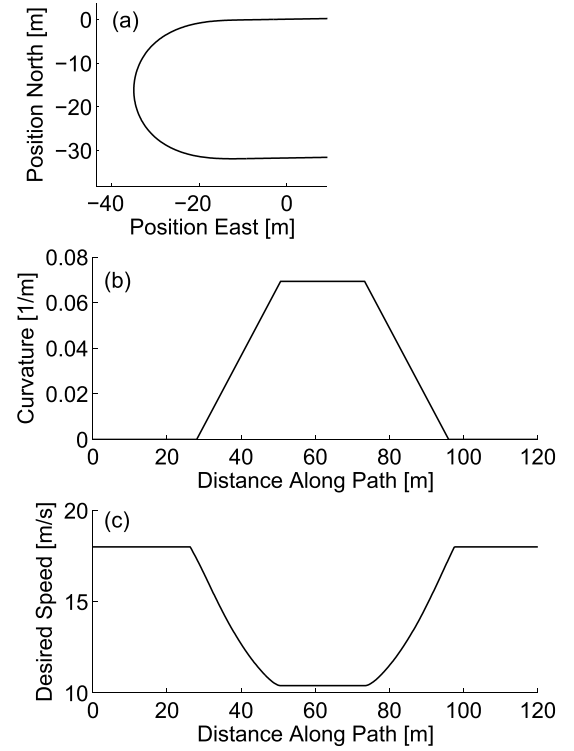

Fig. 2. Trajectory (a) path, (b) curvature, and (c) speed profile. Path curvature dictates path shape and bounds possible speeds.

C. Usage

The experiments use a single path, shown in Fig. 2(a), which is parameterized by the curvature shown in Fig. 2(b). Setting $\mu_{\text{des}} = 0.9\mu \approx 0.765$ creates a speed profile near the handling limits of the vehicle, shown in Fig. 2(c), with an 18-m/s max speed. Experiments use this static trajectory example to demonstrate the capabilities of the controller, but trajectories could just as easily derive from an online trajectory generator.

IV. VEHICLE MODEL

The MPC optimization solves for optimal front tire forces as the input u in (1) that achieve the desired tradeoff among collision avoidance, stabilization, and path tracking. This optimization is based on the predictions from a five-state bicycle model representation of the vehicle, introduced as (2). The first two vehicle states and associated tire models relate to vehicle stabilization, while the last three path states relate to path tracking and collision avoidance. Linearizing the governing equations about predicted operating points produces an affine time-varying vehicle model amenable to real-time implementation.

A. Vehicle States

The planar bicycle model used to represent the vehicle, shown in Fig. 3, utilizes small angle assumptions and the approximation that the tires on each axle can be lumped together. Assuming a given longitudinal velocity U_x, the vehicle's velocity states are lateral velocity U_y and yaw rate r, which are described by the governing equations

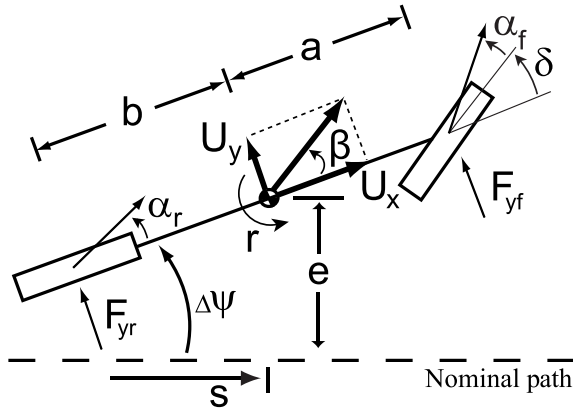$$\dot{U}_y = \frac{F_{yf} + F_{yr}}{m} - rU_x \quad \dot{r} = \frac{aF_{yf} - bF_{yr}}{I_{zz}} \tag{10}$$

Fig. 3.   Bicycle model schematic. U_y and r represent the vehicle velocity states, and $\Delta\psi$, e, and s relate the vehicle to the path.

Fig. 4.   Experimental data from standard passenger vehicle tires on asphalt ($\mu \approx 0.85$), and front tire model fit.

where F_{yf} and F_{yr} are the front and rear tire forces, m is the vehicle's mass, I_{zz} is the vehicle's moment of inertia, and a and b are the distances from the vehicle's center of gravity to the front and rear axles, respectively.

Vehicle side slip angle, $\beta \approx U_y/U_x$, is sometimes used instead of lateral velocity U_y, because it offers a more intuitive vehicle state. This is a reasonable choice when assuming constant speed, as is done in [13] and [14]; however, when incorporating the changes in speed, the derivative of β introduces a nonconvex $\beta \dot{U}_x/U_x$ term in the governing equation that is easily avoided by instead using U_y.

B. Tire Model

Tire forces F_{yf} and F_{yr} in (10) are defined by the brush tire model originally presented by Fiala [18] and modified by Pacejka [19]

$$F_y = \begin{cases} -C_\alpha \tan\alpha + \dfrac{C_\alpha^2}{3\eta\mu F_z}|\tan\alpha|\tan\alpha \dots \\ \qquad\qquad -\dfrac{C_\alpha^3}{27\eta\mu^2 F_z^2}\tan^3\alpha, & |\alpha| < \alpha_{\text{sat}} \\ -\eta\mu F_z \text{sgn}\,\alpha, & \text{otherwise} \end{cases}$$
$$= f_{\text{tire}}(\alpha) \tag{11}$$

where C_α and μ are experimentally determined tire cornering stiffness and friction coefficients, F_z is the normal load, α is the tire slip angle, and α_{sat} is the saturating slip angle. The tire slip angle is the angle between the tire's orientation and direction of travel

$$\alpha_f = \tan^{-1}\left(\frac{U_y + ar}{U_x}\right) - \delta \approx \frac{U_y + ar}{U_x} - \delta \tag{12}$$

$$\alpha_r = \tan^{-1}\left(\frac{U_y - br}{U_x}\right) \approx \frac{U_y - br}{U_x} \tag{13}$$

where δ is the steering angle. The saturating tire slip angle is the slip angle at which additional slip angle produces no additional force; from (11), this is defined as

$$\alpha_{\text{sat}} = \tan^{-1}\left(\frac{3\eta\mu F_z}{C_\alpha}\right). \tag{14}$$

Lateral tire forces as defined by (11) depend on a presumed longitudinal force F_x. This longitudinal force is accounted
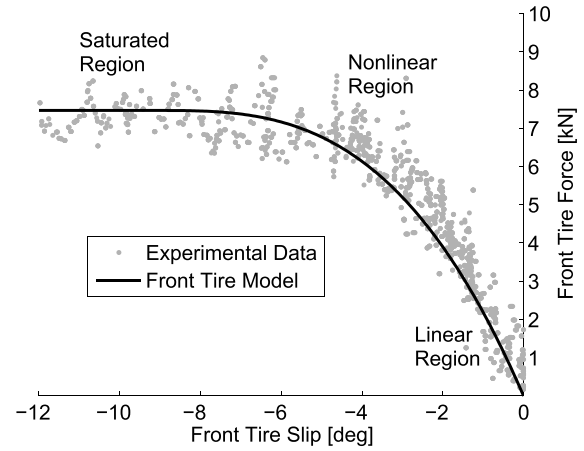
for in two ways. First, a derating factor η has been added to the model to capture the reduced lateral force capability due to F_x. Friction limits (8) reduce the maximum achievable lateral force F_y for a given longitudinal force F_x. This effect is modeled using a derating factor defined as follows:

$$\eta = \frac{\sqrt{\mu^2 F_z^2 - F_x^2}}{\mu F_z}. \tag{15}$$

Second, the normal load F_z on a tire varies according to a static longitudinal weight transfer model

$$F_{zf} = \frac{1}{L}(mbg - hF_x)$$
$$F_{zr} = \frac{1}{L}(mag + hF_x) \tag{16}$$

where L is the length of the vehicle and h is the height of the vehicle's center of gravity. The first term in these equations is the static weight distribution and the second term is the effect of accelerating or braking.

The parameters for tire cornering stiffness and road-tire friction were experimentally determined by driving the vehicle at a constant speed with a slowly increasing steering angle. The resulting quasi-steady-state maneuver provides lateral accelerations and vehicle states that can be mapped to expected front and rear tire forces and slip angles. Fitting C_α and μ to this data provides empirical values for these quantities that account for otherwise unmodeled effects, such as lateral weight transfer and suspension geometry. Fig. 4 provides an example of the resulting data and fit. In the future, these parameters could be estimated online using techniques such as that suggested by Hsu *et al.* [20].

The affine vehicle model incorporates nonlinear front and rear tire models (11) differently. The goal with the rear tire model is to accurately capture the nonlinear relationship between vehicle states and rear tire force generation, which becomes especially critical to stabilization. A combination of regularization, linearization, and trust regions achieves this goal.

Regularization in conjunction with a successive linearization approach, in which predicted states are based on the previous
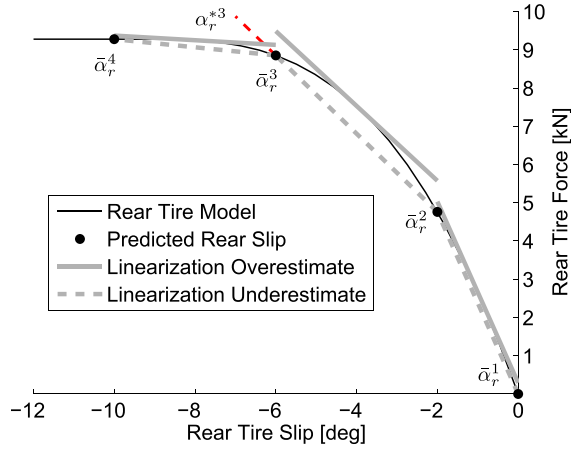
Fig. 5. Rear tire model, predicted operating points at four time steps, and potential linearizations of the rear tire model for these points. Linearizations can consistently overestimate or underestimate rear tire force between points.

execution solution ($\bar{x} = f(x_{\text{prev}}^*)$), provides predicted states $\bar{x}$ at each time step k in the prediction horizon. These states provide slip angle predictions $\bar{\alpha}_r$ through (13) that can be used as nominal points at which to perform linearizations of the nonlinear tire curve (11). However, using only the previous solution can lead to jitter across execution steps, as the fast dynamics lead to solutions and resulting linearizations that can oscillate about the nonlinear solution. Instead, an average of the previous prediction $\bar{\alpha}_{r,\text{prev}}$ and resulting solution $\alpha_{r,\text{prev}}^*$ offers a form of regularization of predicted slip angles $\bar{\alpha}_r$ that smooths oscillation across executions

$$\bar{\alpha}_r^k = (1 - r_\alpha)\bar{\alpha}_{r,\text{prev}}^k + (r_\alpha)\alpha_{r,\text{prev}}^{*k} \quad k = 1 \ldots n. \quad (17)$$

A value of $r_\alpha = 1/2$ significantly reduces jitter while still relying on the previous solution, which is important in allowing the tire model to adjust over control executions to a new plan in an emergency scenario.

Once given predicted slip angles $\bar{\alpha}_r$, linearizations of the nonlinear tire curve can be performed. The tire curve at step k in the prediction horizon could be linearized about either $\bar{\alpha}_r^k$ or $\bar{\alpha}_r^{k+1}$, but the resulting approximation becomes excessively inaccurate near $\bar{\alpha}_r^{k+1}$ or $\bar{\alpha}_r^k$, respectively. The linearization could alternatively be performed at the average of $\bar{\alpha}_r^k$ and $\bar{\alpha}_r^{k+1}$. Assuming the tire model defined in (11), this linearization always overestimates the expected rear tire force for the time between k and $k + 1$, since the magnitude of the second derivative of (11) decreases away from $\alpha_r = 0$. An example is pictured in Fig. 5 as the linearization overestimate. Instead, the linearization is formed as an interpolation between $\bar{\alpha}_r^k$ and $\bar{\alpha}_r^{k+1}$, as pictured in Fig. 5 as the linearization underestimate. This approach ensures accurate predictions about $\bar{\alpha}_r^k$ and $\bar{\alpha}_r^{k+1}$ up to the force generation capabilities of the tires but underestimates intermediate force generation for the time between k and $k + 1$. Some linearization error is inevitable and becomes increasingly significant with larger differences in $\bar{\alpha}_r^k$ and $\bar{\alpha}_r^{k+1}$ associated with longer time steps. This approach leverages such error by tending to underestimate intermediate forces, which in turn causes the controller to react earlier to potential threats and stability violations and improves the overall safety of the system.

While the chosen linearization will underestimate tire force between k and $k + 1$ if slip angle predictions do not change between control executions (i.e., $\alpha_r^* = \bar{\alpha}_r$), the linearization will overestimate force for solutions and corresponding slip angles that shift away from the prediction points. This can be seen in Fig. 5: if $\alpha_r^{*3} < \bar{\alpha}_r^3$ for example, then the linearization from $k = 2$ to $k = 3$ will overestimate F_{yr}^3. For this reason, trust regions bound the vehicle states to allowable deviations from the predictions to ensure a level of model accuracy. Trust regions, as generally represented in (4), are defined in terms of $\alpha_{r,\text{sat}}$ and n_{tr}, so that n_{tr}/r_α control executions are necessary to traverse the entire range of rear tire force

$$\bar{\alpha}_r^k - \frac{2}{n_{\text{tr}}}\alpha_{r,\text{sat}} \le \alpha_r^{*k} \le \bar{\alpha}_r^k + \frac{2}{n_{\text{tr}}}\alpha_{r,\text{sat}} \quad k = 1 \ldots n. \quad (18)$$

When an emergency occurs, the controller will require up to n_{tr}/r_α executions to reach the final solution. The key, however, is that during each of those steps, an accurate model still allows the controller to determine an appropriate input, enabling an immediate response even as the model converges. A value of $n_{\text{tr}} = 8$ limits rear slip angle variation to $(1/4)\alpha_{r,\text{sat}}$ while ensuring that the tire curve can be fully traversed within $n_{\text{tr}}/r_\alpha = 8 \cdot 2 = 16$ executions of the controller. This totals 160 ms at the 10-ms sample rate.

The resulting rear tire model extends the linear slip angle bounds (around $3°$ for the tire model in Fig. 5) often used [6], [11], [12], to all slip angles that can generate additional force. Katriniok et al. [7] similarly extended slip angle bounds beyond the linear region with a local linearization approach. However, their focus on only trajectory tracking enabled short rather than long prediction steps and in turn linearizations of the tire curve at individual operating points with less concern for model propagation between steps.

The vehicle model also incorporates the front tires to enable full tire force utilization; however, unlike the rear tire slip angle that evolves based on vehicle states, the controller directly controls front tire slip through the steering input. This allows F_{yf} to serve directly as the input to the vehicle model. Using F_{yf} as the input, rather than steering angle, allows the optimization to choose inputs that explicitly account for the limited capabilities of the front tires. Once calculated, a desired F_{yf} can be converted into a steering angle according to (11) and (12)

$$\delta = \frac{U_y + ar}{U_x} - f_{\text{tire}}^{-1}(F_{yf}) \quad (19)$$

where f_{tire}^{-1} represents the inverse tire curve.

C. Path States

The last three vehicle states relate the vehicle to the nominal path. Distance along the path s, heading error between the vehicle and path $\Delta\psi$, and lateral error from the path e are described by the following governing equations of motion:

$$\dot{\Delta\psi} = r - \dot{s}K(s) \quad (20)$$

$$\dot{e} = U_x \sin\Delta\psi + U_y \cos\Delta\psi \approx U_x\Delta\psi + U_y \quad (21)$$

$$\dot{s} = \frac{U_x \cos\Delta\psi - U_y \sin\Delta\psi}{1 - K(s)e} \approx \frac{U_x}{1 - K(s)\bar{e}}. \quad (22)$$

Approximations for $\dot{e}$ and $\dot{s}$ assume small $\Delta\psi$ and U_y. The governing equation for $\dot{s}$ also uses expected lateral error, $\bar{e}$, rather than the actual lateral error e, to preserve linearity of the equation. Expected lateral error derives from x^*_{prev}, similar to the slip angles used for rear tire linearization. Lateral error is an integral of other vehicle states and does not rapidly change between control executions such as rear slip angle, and so a regularization approach such as (17) is not necessary.

D. Model Linearization

The resulting vehicle model can be expressed in continuous form with the affine expression

$$\dot{x} = Ax + BF_{yf} + C \quad (23)$$

where $x = [U_y \ r \ \Delta\psi \ s \ e]^T$. Matrices A, B, and C depend on operating points F_x, U_x, $\bar{\alpha}_r$, $\bar{e}$, and $K(s)$. The model can be discretized at individual time steps along the prediction horizon, each with associated operating points, to yield an affine time-varying model similar to (2). Section VI discusses the discretization of this model, since discretization choice depends on the time steps used.

V. STABILITY AND ENVIRONMENTAL ENVELOPES

Vehicle stabilization and collision avoidance objectives are defined in terms of envelopes, or regions in the state space where the vehicle state should reside.

A. Stability Envelope

This controller uses the envelope suggested by Beal and Gerdes [21], which bounds the vehicle's velocity states U_y and r by the maximum steady-state force generation of the tires. Rear tire saturation defines the lateral velocity U_y limit. Saturation occurs at the rear tire saturation slip angle (14) and converts into a bound on U_y with (13)

$$U_{y,\max} = U_x \alpha_{r,\text{sat}} + \text{br}. \quad (24)$$

This bound incorporates a rear tire slip angle limit similar to the previous work [6], [11], [12], but extends beyond the linear regime and into the fully saturated region, allowing full utilization of rear tire force.

An additional bound on yaw rate creates a closed set for vehicle velocity states and is based on the maximum steady-state condition of (10). This yields a maximum sustained yaw rate

$$r_{\max} = \min\left(\frac{F_{yf,\max}(1+a/b)}{mU_x}, \frac{F_{yr,\max}(1+b/a)}{mU_x}\right) \quad (25)$$

where a and b are the distances from the vehicle's center of gravity to the front and rear axles, respectively. The two quantities in (25) encode front and rear tire saturation.

The bounds (24) and (25) define an invariant set for vehicle velocity states U_y and r, pictured in Fig. 6. Vehicle stability is guaranteed for all states in the envelope. Exceeding these bounds does not necessarily result in instability, but for states outside the envelope, there is no guarantee that a control input exists to move the system closer to the boundary in the next time step.
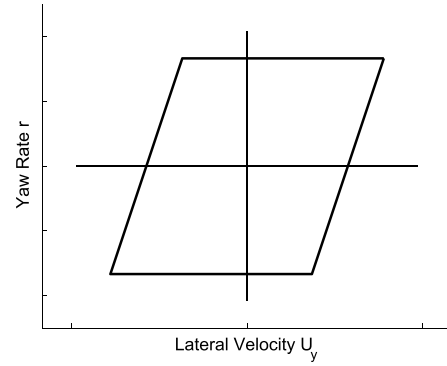

Fig. 6. Stability envelope for vehicle stabilization. Bounds on the vehicle's velocity states U_y and r provide an invariant set that the controller seeks to remain inside.

The stability bounds derive from expected maximum tire force, which is subject to a degree of uncertainty. Underestimated tire force results in an overly restrictive set, preventing the vehicle from achieving otherwise acceptable vehicle states. Overestimated tire force leads to constraints that are not restrictive enough. In that case, vehicle states may reach their real limit and begin quickly growing before reaching the constraint boundary, requiring additional control authority to return them to within the set. With enough mismatch, this could result in a spin. The experiments in Section VIII demonstrate that even roughly estimated tire parameters result in a sufficient stability envelope.

B. Environmental Envelope

Collision avoidance is modeled using an environmental envelope as originally proposed by Erlien *et al.* [13]. The environmental envelope is a set of lateral error bounds $e_{\max}$ and $e_{\min}$ as a function of distance s along a path that avoids obstacles and remains on the road. Enforcing these bounds on the vehicle's lateral error as follows:

$$e^k + f_{\text{width}}(\Delta\psi^k) \leq e^k_{\max}$$
$$e^k - f_{\text{width}}(\Delta\psi^k) \geq e^k_{\min} \quad (26)$$

provides a convex method of enforcing environmental constraints on the vehicle's position. The function $f_{\text{width}}(\Delta\psi)$ linearly approximates the vehicle's effective width as its orientation from the path changes. For example, at $\Delta\psi = 0$, the vehicle's effective width is half its actual width, while at $\Delta\psi = 90°$, the vehicle's effective width is at least a, the distance from the center of gravity to the front axle.

Experiments presented here rely on a virtual, predefined set of road bounds and obstacles. Virtual obstacles offer consistent and repeatable tests in order to validate the control approach itself. To be functional in a real situation, the virtual environment must be replaced by the real environment as perceived by sensors, such as radar and camera, which adds complexity. First, a real environment may offer multiple paths of traversal, such as passing on the left or right of an obstacle, which must be chosen before bounds (26) can be determined. This choice can be made prior to the problem formulation via a higher level decision planner, or each choice can be encoded as a separate MPC problem to be chosen

among after evaluation. Second, perceived sensor data will be less spatially accurate and temporally consistent than a virtual environment. Spatial inaccuracy can be accounted for with added buffer distance between the actual and enforced environmental bounds in (26), while temporal inconsistency can be reduced with object tracking algorithms. Perceiving and understanding the environment is a major field of research itself, but once such information has been classified in terms of road edges and obstacle locations along with a desired path of traversal, the remainder of the control structure does not change.

VI. VARIABLE TIME STEP DISCRETIZATION

To effectively prioritize among tracking, stabilization, and collision avoidance, the MPC controller must optimize lateral force inputs accounting for near-term vehicle dynamics and future collision avoidance. However, the number of discretization points that can be evaluated in real time is limited by computational power. Therefore, short time steps that are necessary to model vehicle dynamics result in a prediction horizon too short to effectively incorporate future environmental information. Instead, the prediction horizon is discretized with different length time steps. Short time steps in the near term provide accurate vehicle dynamics modeling for path tracking and stabilization. Longer time steps further along the prediction horizon extend the predictions to incorporate future obstacles and path information while offering more approximate vehicle dynamics modeling for tracking and stabilization.

Discretizing the prediction horizon with variable time steps was originally suggested by Erlien *et al.* [13] for a shared control framework; short time steps mapped to tracking driver commands and long time steps encoded environment constraints. The same time steps are used here, with two underlying modifications for fully autonomous control. Both changes improve the match over the prediction horizon between the MPC plan and the resulting closed-loop response. Unlike shared control, which incorporates unknown future driver commands, the closed-loop response of a fully autonomous system, in the absence of disturbances and model uncertainty, should match the optimized plan. Otherwise, the closed-loop performance is suboptimal. Matching the closed-loop response also improves model linearizations such as rear tire force, which are based on the predictions of future vehicle states.

A. Selected Time Steps

The prediction horizon consists of three components: a near term comprised of ten $T_{s,\text{short}} = 10$-ms time steps, followed by one time-varying "correction" time step $T_{s,\text{corr}}$, and finally a far term horizon of 19 $T_{s,\text{long}} = 200$-ms time steps. Computation time to optimize the resulting $n = 30$ problem on an i7 processor approaches the 10 ms available for calculation.

The near-term horizon provides high fidelity vehicle dynamics modeling necessary for path tracking and vehicle stabilization. The controller executes at 10-ms intervals, so these 10 ms time steps accurately discretize the continuous system in the near term. The far term horizon extends the predictions further into the future, incorporating the next 3.8 s of environmental information.
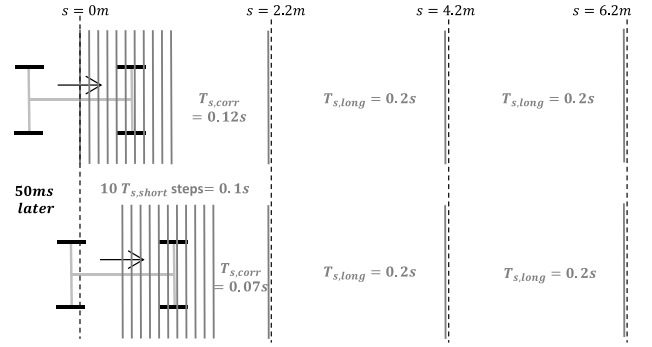

Fig. 7. Prediction horizon discretization assuming a speed of 10 m/s, with the time steps in gray and the spatial discretization in black. The correction time step $T_{s,\text{corr}}$ varies to maintain consistent spatial locations of the long time steps as the vehicle moves forward.

The correction time step exists between the two horizons, varying length between $T_{s,\text{short}}$ and $T_{s,\text{long}}$ to maintain the spatial location of the long time steps. Without the correction time step, the estimated distance along the path of the long time steps shifts with each short time step, resulting in a different spatial discretization of the environment with each controller execution. With the correction time step, the estimated positions at each long time step remain fixed, maintaining a consistent environmental discretization throughout the receding horizon. An example of the effect of the correction time step is shown in Fig. 7.

B. Vehicle Model Discretization

Continuous dynamical models are often discretized at each time step with a zero-order hold (ZOH) approximation. A ZOH assumes constant continuous inputs over the duration of each discrete time step and yields a causal difference equation that can be included as a constraint in the MPC problem. A ZOH assumption works well for the short time steps, since the assumed constant continuous inputs accurately reflect the actual input to the vehicle.

A ZOH less accurately represents time steps in the far term horizon. As the receding horizon shifts forward, the long time steps are eventually replaced by the short time steps, and the controller will calculate inputs at the shorter time intervals. To address this far-term modeling inaccuracy, the long time steps are discretized with a first-order hold (FOH), as suggested by Brown *et al.* [14]. An FOH assumes linear variation between time steps. This approximation provides a more accurate interpolation of inputs and results in a better prediction of future state propagation. Furthermore, the FOH discretization offers a discrete representation of linearly varying curvature paths such as the trajectory in Fig. 2.

The continuous vehicle model (23) is discretized using a matrix exponential approach [22] with a ZOH for the first ten short time steps and an FOH for the next 20 long time steps. The resulting discretized vehicle model is

$$x^{k+1} = \begin{cases} A^k x^k + B_1^k F_{yf}^k + C^k & k = 0 \ldots 9 \\ A^k x^k + B_1^k F_{yf}^k + B_2^k F_{yf}^{k+1} + C^k & k = 10 \ldots 29 \end{cases}$$
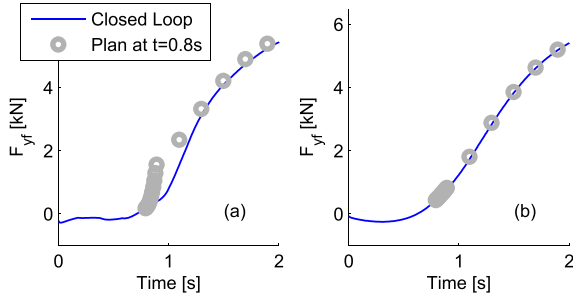
$$(27)$$

Fig. 8. Simulation results of the MPC plan compared with the closed-loop response with (a) unscaled weights and ZOH discretization and (b) scaled weights and combined ZOH and FOH discretization. The plan matches better in (b) and results in a different closed-loop response from (a).
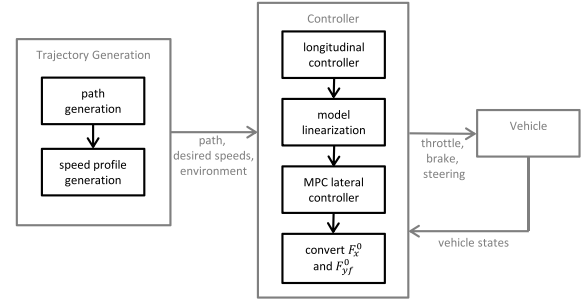

Fig. 9. Control structure. The controller calculates longitudinal and then lateral inputs, which are converted into steering, throttle, and brake commands for the vehicle.

where A^k, B_1^k, B_2^k, and C^k are the time-varying linearizations along the prediction horizon.

C. Weight Scaling

In addition to discretization choice, optimization weights should scale according to time step length to remain consistent along the receding horizon. Consider constant vehicle states x_c for a period of time T evenly divisible by both short and long time steps. The cost associated with these vehicle states, as given by an objective function (1), is

$$J_1 = \sum_{k=1}^{T/T_{s,\text{short}}} x_c^T Q_{\text{short}}^k x_c, \quad J_2 = \sum_{k=1}^{T/T_{s,\text{long}}} x_c^T Q_{\text{long}}^k x_c. \quad (28)$$

For these costs to be equivalent, Q_{short}^k must be related to Q_{long}^k by

$$Q_{\text{short}}^k = \frac{T_{s,\text{short}}}{T_{s,\text{long}}} Q_{\text{long}}^k. \quad (29)$$

That is, the weights must be scaled along the horizon according to the length of the associated time step. Similar logic can be applied to yield a different scaling for weights that incorporate time.

Fig. 8 provides a simulation example of planned and closed-loop control inputs before and after the discretization and weight scaling modifications. Fig. 8(a) shows the results from the controller using a ZOH discretization throughout the horizon and unscaled cost weights. The resulting closed-loop response matches poorly with the plan. Fig. 8(b) shows the simulation results using the combined ZOH and FOH discretization and scaled cost weights. Here, the plan matches the closed-loop response. Fig. 8(b) shows that, in the absence of disturbances and plant-model mismatch, using appropriate discretization and weight scaling significantly reduces the discrepancy between the plan and closed-loop response.

VII. Controller Framework

The trajectory, stability envelope, and environmental envelope define the control objectives among which the controller must mediate. An MPC controller determines lateral inputs that best meet these objectives given a set of longitudinal commands provided from a simple longitudinal controller. The overall control structure is pictured in Fig. 9.

A. Longitudinal Controller

The longitudinal controller determines the longitudinal forces along the prediction horizon to track speeds defined by the speed profile. The controller assumes that the vehicle's longitudinal dynamics can be represented as a point mass. At each point k in the MPC prediction horizon, the following feedforward plus feedback control law is used:

$$F_x, \text{cmd}^k = m a_{x,\text{des}}(s^k) + k_p \left(U_{x,\text{des}}(s^k) - U_x^k \right) + F_{x,\text{drag}}^k \quad (30)$$

where $U_{x,\text{des}}(s^k)$ is the speed determined by the speed profile at s^k, $a_{x,\text{des}}(s^k)$ is the necessary longitudinal acceleration to track the rate of change in the speed profile at s^k, k_p is a speed tracking gain, and $F_{x,\text{drag}}^k$ offsets dissipative effects, such as rolling resistance and aerodynamic drag. The commanded forces $F_{x,\text{cmd}}^k$ are calculated at each point in the prediction horizon and integrated to estimate vehicle speed throughout the horizon. Lookup tables convert $F_{x,\text{cmd}}^0$ into brake pressures and motor torques that are applied to the vehicle.

B. Lateral Controller

The MPC optimization calculates front tire forces based on the affine vehicle model (27). Nonzero diagonal entries of Q corresponding to lateral error e and heading error $\Delta\psi$ weights path tracking. The trust region on rear tire slip angle (18) can be represented as

$$H_{\alpha_r}^k x^k \leq G_{\alpha_r}^k \quad k = 1 \ldots n. \quad (31)$$

Representing the stability constraints (24) and (25) in the problem as

$$H_{\text{veh}}^k x^k \leq G_{\text{veh}}^k \quad k = 1 \ldots n \quad (32)$$

enforces vehicle stabilization. Encoding the environmental envelope (26) as

$$H_{\text{env}}^k x^k \leq G_{\text{env}}^k \quad k = 1 \ldots n \quad (33)$$

incorporates collision avoidance.

The final optimization problem takes the form

$$\min \sum_{k=1}^{n} x^{k^T} Q^k x^k + v^{k^T} R^k v^k + \left(\sigma_{\text{veh}}^k \right)_+ W_{\text{veh}}^k$$
$$+ \ldots \left(\sigma_{\text{env}}^k \right)_+ W_{\text{env}}^k \quad (34)$$

TABLE I
CONTROLLER PARAMETERS AND WEIGHTS

Description	Symbol	Value	Units
Controller Parameters			
Number of time steps	n	30	(none)
Controller execution time step	$T_{s,MPC}$	0.01	s
Short time step size	$T_{s,short}$	0.01	s
Long time step size	$T_{s,long}$	0.2	s
Correction time step size	$T_{s,corr}$	[0.01 0.2]	s
Rear tire slip regularization	r_α	0.5	(none)
Rear tire trust region bound	n_{tr}	8	(none)
Weight Normalizations			
Max expected input change	$dF_{yf,max}$	10	kN/s
Max expected heading error	$\Delta\psi_{max}$	0.15	rad
Max expected lateral error	e_{max}	3	m
Max expected yaw rate	r_{max}	1	rad/s
Max expected lateral velocity	$U_{y,max}$	$0.15U_x^k$	m/s
Weight Prioritizations			
Environment collision slack cost	W_e	500	(none)
Yaw rate stability slack cost	W_r	50	(none)
Lateral velocity stability slack cost	W_{U_y}	50	(none)
Heading error cost	$Q_{\Delta\psi}$	1	(none)
Lateral error cost	Q_e	1	(none)
Max change in input cost	R_{max}	5	(none)
Nominal change in input cost	R_{base}	0.1	(none)

subject to the vehicle model (27) and

$$H_{\alpha_r}^k x^k \leq G_{\alpha_r}^k \quad k = 1 \ldots n \tag{35}$$

$$H_{veh}^k x^k \leq G_{veh}^k + \sigma_{veh}^k \quad k = 1 \ldots n \tag{36}$$

$$H_{env}^k x^k \leq G_{env}^k + \sigma_{env}^k \quad k = 1 \ldots n \tag{37}$$

$$\left| F_{yf}^k \right| \leq F_y f, \quad \max^k \quad k = 0 \ldots n \tag{38}$$

$$|v^k| \leq v_{max}^k \quad k = 1 \ldots n \tag{39}$$

where σ_{veh} and σ_{env} are the slack variables on the constraints enforcing the stability and environmental envelopes, $v^k = F_{yf}^k - F_{yf}^{k-1}$, and v_{max} is a maximum allowable change in force based on the physical capabilities of the steering system.

This optimization incorporates both collision avoidance and stabilization, enabling the prioritization of these two sometimes conflicting objectives. W_{env} and W_{veh} weight collision and stability constraint violations. Ideally, both constraints are met, and slack variables σ_{veh} and σ_{env} are zero in the optimal solution. However, these softened constraints ensure problem feasibility if constraints must be violated, and their associated weights provide a means of encoding these priorities. Setting $W_{env} \gg W_{veh} \gg ||Q||_\infty$ encodes a prioritization of collision avoidance, then stability, and finally path tracking. This prioritization allows the controller to selectively violate stability criteria if necessary to avoid a collision. Clear priorities are achieved by scaling all of the weights by the maximum expected value of the corresponding variables. Table I outlines the experimental values for normalization and prioritization.

Weighting functions are based on desired behavior of the associated variables. Quadratic weights on x and v allow small deviations of these quantities but strongly penalize large deviations, while linear weights on σ_{veh} and σ_{env} immediately penalize small constraint violations [23]. Such weights encode the permissibility of small tracking errors while penalizing even small violations of either envelope. The initial design introduced in [15] penalized σ_{env} with a quadratic weight but led to small, avoidable envelope violations in some scenarios.

TABLE II
WEIGHTING FUNCTIONS

Weight = priority × normalization × time step scaling

$$R^k = \min\left(\frac{C_f R_{base}}{\bar{C}_f(F_{yf,prev}^k)}, R_{max}\right) \frac{1}{(dF_{yf,max} T_{s,long})^2} \frac{T_{s,long}}{T_s^k}$$

$$Q^k = \text{diag}\left(0, \; 0, \; Q_{\Delta\psi}\frac{1}{\Delta\psi_{max}^2}\frac{T_s^k}{T_{s,long}}, \; 0, \; Q_e\frac{1}{e_{max}^2}\frac{T_s^k}{T_{s,long}}\right)$$

$$W_{veh}^k = \left[W_r\frac{1}{r_{max}}\frac{T_s^k}{T_{s,long}} \qquad W_{U_y}\frac{1}{U_{y,max}}\frac{T_s^k}{T_{s,long}}\right]$$

$$W_{env}^k = \left[W_e\frac{1}{e_{max}}\frac{T_s^k}{T_{s,long}} \qquad W_e\frac{1}{e_{max}}\frac{T_s^k}{T_{s,long}}\right]$$

In addition to the objectives of collision avoidance, stabilization, and path tracking, the optimization also weights input changes through R and v. Weighting changes in F_{yf} promotes earlier and smoother response to constraints later in the horizon. Furthermore, quickly changing the steering angle to achieve a desired F_{yf} can excite unmodeled vehicle dynamics and introduce steering tracking errors. However, F_{yf} relates nonlinearly to steering angle through (19); small changes in F_{yf} near the saturated region of the tire curve result in large steering changes. This nonlinear relationship can be approximated in the optimization by scaling R by the effective cornering stiffness, or the slope of the tire curve, around the operating region defined by the solution to the previous execution step. The resulting scaling is

$$R^k = \min\left(\frac{C_f}{\bar{C}_f(F_{yf,prev}^{*k})}R_{base}, R_{max}\right) \tag{40}$$

neglecting the normalization and time step scalings. $\bar{C}_f(F_{yf,prev}^{*k})$ is a local cornering stiffness, or the slope of a linearization in Fig. 5, based on the previous execution solution $F_{yf,prev}^*$, C_f is the nominal cornering stiffness, R_{base} is the base R weighting, and R_{max} is the maximum allowable R weighting. The minimization is necessary, because the cornering stiffness at the peak of the tire curve approaches zero, resulting in an otherwise infinite weight. Table II outlines the complete weighting functions.

The optimization resolves each time step, as is standard with MPC, and F_{yf}^0 converts using (19) into a steering angle command. The optimization solves in real time at 100 Hz with code generated from CVXGEN [24].

VIII. EXPERIMENTAL RESULTS

Experiments are conducted with X1, a rear-wheel drive electric vehicle, shown in Fig. 10. A GNSS-aided inertial navigation system provides real-time measurements of vehicle states. An i7 computer calculates control inputs at 100 Hz, which are executed by the vehicle's drive-by-wire system.

The vehicle follows the path and speed profile shown in Fig. 2 under three conditions: no obstacle, a known obstacle, and an obstacle that appears midtest. Fig. 11 shows the location of the obstacle, when the obstacle is detected by the controller, and the resulting vehicle paths for the three cases. Videos of the experimental setup and MPC plan while driving
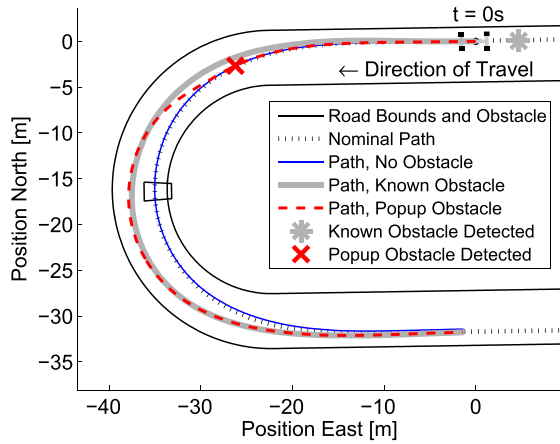
Fig. 10. Drive-by-wire test vehicle X1.

Fig. 11. Environment and vehicle paths for the three test cases. When the obstacle is present, the vehicle deviates from the desired path to avoid it. In the known obstacle case, the vehicle enters the turn outside of the desired path as it prepares to avoid the obstacle.

are available at https://stanford.box.com/s/vfdzcwot4t6gj6mx b9yej87h6e4xxwcc.

A. No Obstacle

In the first scenario, the vehicle drives around the turn near its handling limits. Results demonstrate the controller's ability to safely and smoothly operate the vehicle near its limits. The vehicle tracks the nominal path within 40 cm while operating at 90% of its maximum friction capability of $\mu \approx 0.85$. Fig. 12 shows the resulting vehicle accelerations, speed tracking, and lateral error during the maneuver. Vehicle accelerations remain near the desired accelerations, which are modified from (8) to incorporate longitudinal weight transfer according to (16). The vehicle's acceleration tracks the positive longitudinal acceleration only up to the acceleration capability of the motor. The vehicle similarly tracks desired speeds well, at least within its forward acceleration limits.

Fig. 13 shows the steering commands, yaw rate, and lateral error. The solid dots highlight the periods of time where the vehicle's yaw rate exceeds the yaw stability bound (25), and the open circles show the MPC plan over the prediction horizon when $t = 2$s. Of the two vehicle velocity states r and U_y, only yaw rate r and its associated stability bound (25) are shown in Fig. 13, because the U_y stability constraint (24) is never active. The vehicle's yaw rate exceeds (25) twice. Due to model uncertainty, the MPC plan at $t = 2$s, in
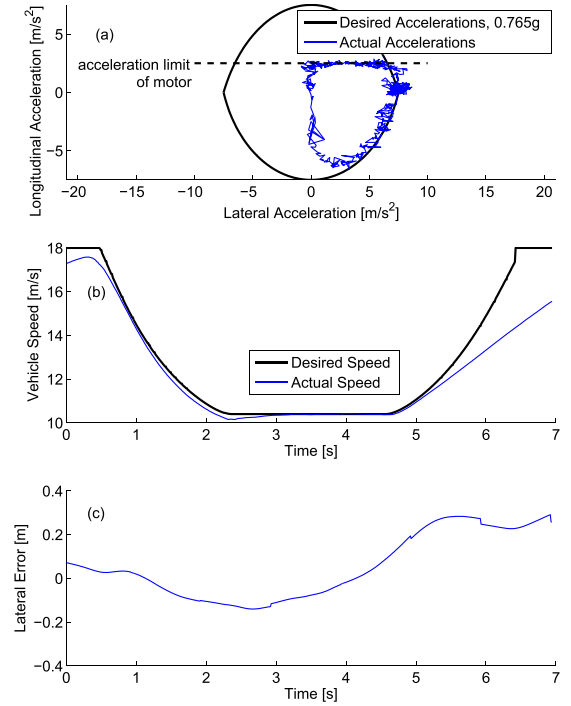

Fig. 12. No obstacle test case. (a) Acceleration. (b) Speed. (c) Lateral error. Full longitudinal deceleration is not required due to the 18-m/s speed limit, and longitudinal acceleration is limited by the motor. Otherwise, accelerations and speeds track the desired values. Lateral error remains below 40 cm.
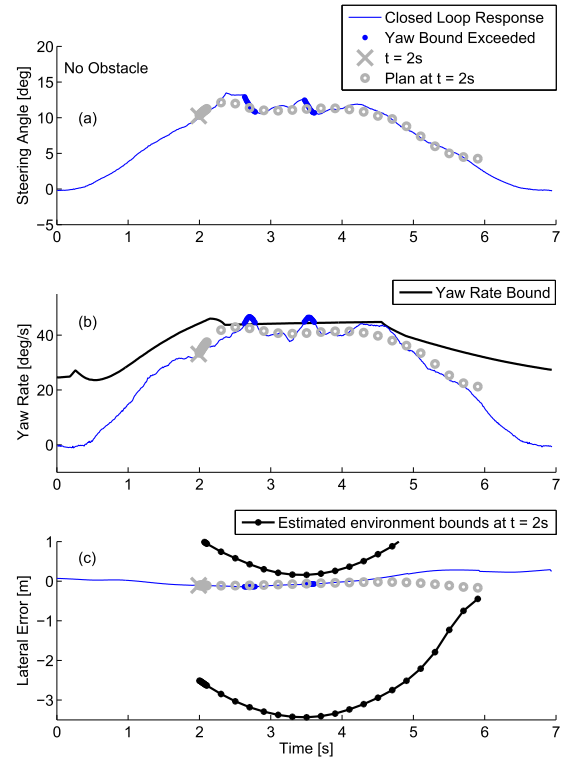

Fig. 13. No obstacle test case. (a) Steering input. (b) Yaw rate. (c) Lateral error. The controller smoothly drives the vehicle around the corner and responds to small violations of the stability criteria.

open circles, does not predict exceeding the stability envelope, but the controller reacts with small countersteers, as seen by decreasing steering angles associated with each violation,
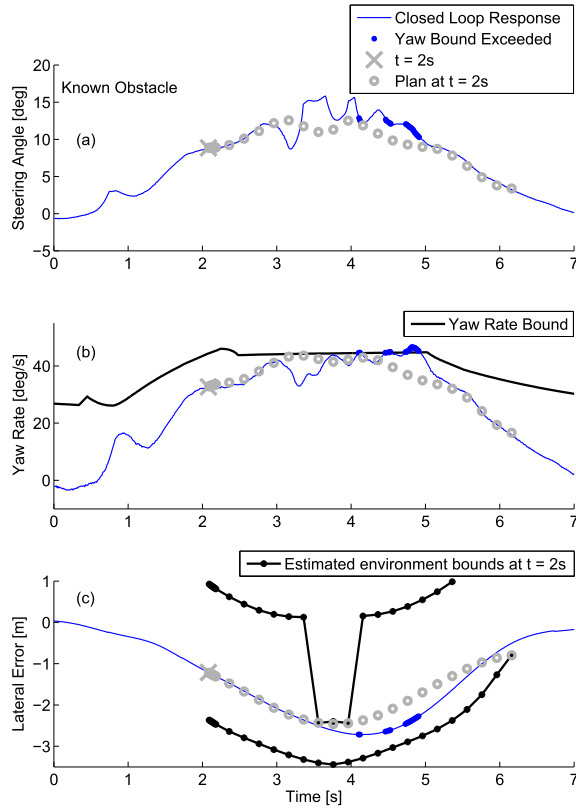
Fig. 14. Known obstacle test case. (a) Steering input. (b) Yaw rate. (c) Lateral error. Because the controller is aware of the obstacle early, the controller can solve for a collision free path without violating stability criteria.
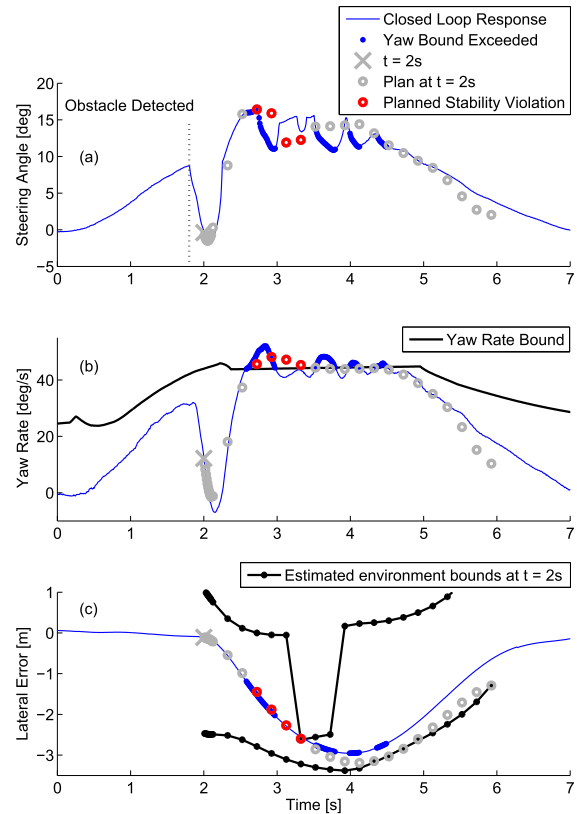

Fig. 15. Pop-up obstacle test case. (a) Steering input. (b) Yaw rate. (c) Lateral error. The limited distance to the obstacle requires temporary violation of stability criteria to avoid running off the road.

to bring the state back within the bound. In this example, the vehicle is never in danger of collision, so the stability constraints are continuously enforced.

B. Known Obstacle

In this example, the vehicle drives around the same turn, now with an obstacle present. The obstacle is always present and becomes incorporated as an environmental constraint when the ∼4 s prediction horizon reaches its location. Similar to Fig. 13, Fig. 14 shows the resulting steering inputs, yaw rate, and lateral error during the maneuver. The lateral error is plotted with discretized environmental bounds at $t = 2$ s, which is similar to Fig. 13(c) except for the three protruding points now representing the obstacle.

The long prediction horizon resulting from the variable time step approach provides ample time for the controller to deviate from the path, allowing the controller to find a solution at $t = 2$ s that remains within the stability envelope. Throughout the maneuver, the controller constantly mediates stabilization and collision avoidance. Just after $t = 3$ s, for example, the controller reduces steering to ensure obstacle avoidance. The following three small countersteers are reactions to predicted stability violations.

C. Pop-Up Obstacle

In this case, an obstacle is detected as the vehicle enters the turn, emulating a previously obstructed stationary object or an

obstacle that entered the road, such as a deer. This example highlights the controller's ability to immediately react and even violate stability criteria in order to still avoid collision. Fig. 15 shows the resulting steering inputs, yaw rate, and lateral error.

Obstacle detection at $t = 1.8$ s results in the plotted MPC plan at $t = 2$ s. In this case, the lateral error in Fig. 15(c) at $t = 2$ s is near zero, unlike the previous case when the obstacle is known and the controller is already 1 m off the nominal path. The limited time-to-collision resulting from the pop-up obstacle necessitates a violation of stability criteria, allowing greater yaw than normally desired to avoid running off the road. The steering input around $t = 2.5$ s does not initially countersteer to correct the yaw bound violation, as in the previous tests. The vehicle's actual yaw rate exceeds the predicted yaw rate during this time, positioning the vehicle to avoid road departure earlier than planned. At $t = 2.75$ s, a path no longer requiring stability violation is calculated, and the controller countersteers to eliminate stability violations. Additional countersteers through the remainder of the maneuver minimize stability violations.

The intervening 200 ms from obstacle detection at $t = 1.8$ s to the plan at $t = 2$ s consists of 20 executions of the MPC controller at 10-ms intervals. The controller converges to a new solution during the first ten execution steps, during which trust region bounds (18) actively limit allowable changes in rear slip angle. During convergence, model linearizations remain accurate and consistent even while the vehicle operates in

this highly nonlinear regime of the tire curve. One significant benefit of such a control scheme is that the controller is capable of using these models to choose intelligent steering inputs from the first time step of obstacle detection onward, as demonstrated by the decreasing steering angle beginning immediately after obstacle detection at $t = 1.8$ s.

IX. CONCLUSION

This paper presents a single controller capable of mediating among the sometimes conflicting objectives of collision avoidance, vehicle stabilization, and path tracking. The controller immediately reacts to new emergency scenarios and can use all of the vehicle's performance capability to avoid an accident. Prioritizing collision avoidance higher than any other objective, including stabilization, leads a different paradigm from the existing approaches. Experiments on an autonomous vehicle demonstrate the performance capability of the controller but more generally highlight the advantage of this prioritization approach to collision avoidance and stabilization.

ACKNOWLEDGMENT

REFERENCES

[1] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, 2008.

[2] M. Montemerlo *et al.*, "Junior: The stanford entry in the urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 569–597, 2008.

[3] J. Levinson *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2011, pp. 163–168.

[4] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for Bertha—A local, continuous method," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 450–457.

[5] E. Liebemann, K. Meder, J. Schuh, and G. Nenninger, "Safety and performance enhancement: The bosch electronic stability control (ESP)," SAE Tech. Paper 2004-21-0060, 2004.

[6] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 566–580, May 2007.

[7] A. Katriniok, J. P. Maschuw, F. Christen, L. Eckstein, and D. Abel, "Optimal vehicle dynamics control for combined longitudinal and lateral autonomous vehicle guidance," in *Proc. IEEE Eur. Control Conf.*, Jul. 2013, pp. 974–979.

[8] P. Falcone, F. Borrelli, H. E. Tsengz, J. Asgari, and D. Hrovat, "A hierarchical model predictive control framework for autonomous ground vehicles," in *Proc. IEEE Amer. Control Conf.*, Jun. 2008, pp. 3719–3724.

[9] 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.*, Sep. 2010, pp. 265–272.

[10] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optim. Control Appl. Methods*, vol. 36, no. 5, pp. 628–647, 2014.

[11] Y. Gao, A. Gray, A. Carvalho, H. E. Tseng, and F. Borrelli, "Robust nonlinear predictive control for semiautonomous ground vehicles," in *Proc. IEEE Amer. Control Conf.*, Jun. 2014, pp. 4913–4918.

[12] V. Turri, A. Carvalho, H. E. Tseng, K. H. Johansson, and F. Borrelli, "Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Oct. 2013, pp. 378–383.

[13] S. Erlien, S. Fujita, and J. C. Gerdes, "Safe driving envelopes for shared control of ground vehicles," *Adv. Automotive Control*, vol. 46, no. 21, pp. 831–836, 2013.

[14] M. Brown, J. Funke, S. Erlien, and J. C. Gerdes, "Safe driving envelopes for path tracking in autonomous vehicles," *Control Eng. Pract.*, May 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066116300831

[15] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes, "Prioritizing collision avoidance and vehicle stabilization for autonomous vehicles," in *Proc. IEEE Intell. Vehicles Symp.*, Jul. 2015, pp. 1134–1139.

[16] J. P. Timings and D. J. Cole, "Minimum maneuver time calculation using convex optimization," *J. Dyn. Syst., Meas., Control*, vol. 135, no. 3, p. 031015, 2013.

[17] J. Funke and J. C. Gerdes, "Simple clothoid lane change trajectories for automated vehicles incorporating friction constraints," *J. Dyn. Syst., Meas., Control*, vol. 135, no. 3, p. 031015, 2013.

[18] H. B. Pacejka, "Tire characteristics and vehicle handling and stability," in *Tire and Vehicle Dynamics*, 3rd ed. London, U.K.: Butterworth-Heinemann, 2012, pp. 1–58.

[19] H. Pacejka, *Tire and Vehicle Dynamics*. New York, NY, USA: Elsevier, 2005.

[20] Y.-H. J. Hsu, S. M. Laws, and J. C. Gerdes, "Estimation of tire slip angle and friction limits using steering torque," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 4, pp. 896–907, Jul. 2010.

[21] C. E. Beal and J. C. Gerdes, "Model predictive control for vehicle stabilization at the limits of handling," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1258–1269, Jul. 2013.

[22] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, vol. 3. Reading, MA, USA: Addison-Wesley, 1998.

[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[24] J. Mattingley and S. Boyd, "CVXGEN: A code generator for embedded convex optimization," *Optim. Eng.*, vol. 13, no. 1, pp. 1–27, 2012.

Joseph Funke received the B.S. degree in electrical and computer engineering from the Franklin W. Olin College of Engineering, Needham, MA, USA, in 2010, the M.S. degree in mechanical engineering from Stanford University, Stanford, CA, USA, in 2012, and the Ph.D. degree in mechanical engineering in 2015, with a focus on obstacle avoidance for automated vehicles.

Dr. Funke was a recipient of the National Science Foundation Graduate Research Fellowship Award in 2011.

Matthew Brown received the B.S. degree in mechanical engineering from the University of California at Berkeley, Berkeley, CA, USA, in 2013. He is currently pursuing the Ph.D. degree in mechanical engineering with Stanford University, Stanford, CA, USA.

His current research interests include architecture of autonomous vehicle systems and the interaction between trajectory planning and following.

Dr. Brown was a recipient of the National Science Foundation Graduate Research Fellowship Award in 2013.

Stephen M. Erlien received the B.S. degree in computer science and mechanical engineering from the University of Wisconsin-Madison, Madison, WI, USA, in 2008, the M.S. degree in mechanical engineering from Stanford University, Stanford, CA, USA, in 2012, and the Ph.D. degree in mechanical engineering in 2015, with a focus on human–computer shared control of dynamic systems and constraint-based control using MPC.

Dr. Erlien was a recipient of the National Science Foundation Graduate Research Fellowship Award in 2009.

J. Christian Gerdes 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, and the Director of the Center for Automotive Research at Stanford, Stanford University. He is a Co-Founder and a Principal Scientist with 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 racetrack with students, instrumenting historic race cars, or trying out their latest prototypes for the future.

Prof. 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.