

Spatial Envelope MPC: High Performance Driving without a Reference

Siyuan Yu[†], Congkai Shen[†], Yufei Xi, James Dallas, Michael Thompson, John Subosits, Hiroshi Yasuda, Tulga Ersal*

Abstract—This paper presents a novel envelope-based model predictive control (MPC) framework designed to enable autonomous vehicles to handle high-performance driving across a wide range of scenarios without a predefined reference. In high-performance autonomous driving, safe operation at the vehicle’s dynamic limits requires a real-time planning and control framework capable of accounting for key vehicle dynamics and environmental constraints when following a predefined reference trajectory is suboptimal or even infeasible. State-of-the-art planning and control frameworks, however, are predominantly reference-based, which limits their performance in such situations. To address this gap, this work first introduces a computationally efficient vehicle dynamics model tailored for optimization-based control and a continuously differentiable mathematical formulation that accurately captures the entire drivable envelope. This novel model and formulation allow for the direct integration of dynamic feasibility and safety constraints into a unified planning and control framework, thereby removing the necessity for predefined references. The challenge of envelope planning, which refers to maximally approximating the safe drivable area, is tackled by combining reinforcement learning with optimization techniques. The framework is validated through both simulations and real-world experiments, demonstrating its high performance across a variety of tasks, including racing, emergency collision avoidance and off-road navigation. These results highlight the framework’s scalability and broad applicability across a diverse set of scenarios.

Index Terms—Model predictive control, autonomous vehicle, reference-free trajectory planning and control, autonomous racing, emergency collision avoidance, off-road navigation

I. INTRODUCTION

High-performance autonomous driving technology has advanced rapidly and significantly in the past decade [1]. The grand opening of the first autonomous racing competition has thrust autonomous driving technology into the spotlight, showcasing performance that rivals expert human drivers in tasks demanding high levels of precision and speed [2]. Recent developments in autonomous driving have enabled reliable performance in advanced collision avoidance capabilities [3]–[5], complex drifting maneuvers [6]–[8], and challenging off-road navigation [9]–[11]. These advancements have greatly expanded the potential applications of autonomous vehicles

[†]These authors contributed equally to this work.

* Corresponding author

S. Yu, C. Shen, Y. Xi, and T. Ersal are with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109. (email: {johnsy, cosh, xiyufei, tersal}@umich.edu)

J. Dallas, M. Thompson, J. Subosits, H. Yasuda are with Toyota Research Institute, Los Altos, CA 94022. (email: {james.dallas, michael.thompson, hiroshi.yasuda, john.subosits}@tri.global)

Toyota Research Institute provided funds to support this work.

across various fields. To execute these extreme maneuvers safely, the vehicle must operate at the limits of its capabilities. However, this is a challenge, because even a small deviation from the desired trajectory can result in catastrophic outcomes.

In this regard, the autonomous system should be designed to effectively utilize all feasible operational regions including the limits, ensuring performance is not unnecessarily sacrificed for safety via overly conservative constraints. For instance, every inch of the race track is critical for drivers to adjust their strategies and enhance their performance. In addition, the algorithm’s design should ensure scalability when applied to a diverse set of scenarios with varying levels of complexity. In this context, the term ‘scalability’ refers to the algorithm’s ability to handle an increasing amount of tasks or scenarios without a significant compromise in performance or without significant redesign. It is also desirable that the system be capable of generating optimal trajectories online in real-time without having to depend on a predefined reference, because deviations from the original plan may render the predefined reference suboptimal or even infeasible.

However, as the literature review in Sec. II reveals, existing methods typically rely on predefined references and therefore either limit vehicle performance to the quality of that reference or do not provide optimal and scalable solutions in performance-demanding scenarios.

To address this gap, this paper presents a novel, spatial envelope model predictive control (MPC) framework for reference-free high performance driving. The proposed framework builds upon a new, computationally efficient vehicle dynamics model tailored for closed-loop optimization based planning and control, capturing the essential dynamics required for aggressive maneuvers. A twice continuously differentiable mathematical formulation of the entire driving envelope is introduced to conservatively estimate the drivable region to be used in MPC. This enables the MPC to break from the restrictive constant-speed assumption previously used in spatial envelope MPC [5], [12]–[14], and instead optimize speed, as well, to maximize performance while maintaining safety.

To the authors’ knowledge, this is the first published MPC algorithm that is experimentally validated for safe and effective high-performance driving at the handling limits in a fully reference-free setting. The algorithm is validated across a wide range of scenarios, including racing, off-road navigation and emergency collision avoidance, demonstrating both generality and real-world applicability.

Finally, a new spatial envelope planning technique is introduced to further enhance applicability. A hybrid approach that

combines optimization-based formulation with reinforcement learning is developed to segment the drivable area into blocks, enabling scalable planning in complex environments.

The original contributions are summarized as follows:

- 1) A validated 3-DoF single-track dynamic model that accounts for longitudinal load transfer and the friction circle limit, while remaining computationally efficient in a fully reference-free setting.
- 2) A hard constraint formulation to mathematically express the spatial envelope with guaranteed conservativeness.
- 3) A real-time Model Predictive Control (MPC) formulation that leverages the first two contributions to optimize vehicle trajectories online without any predefined path.
- 4) A reinforcement learning approach to design a set of blocks to approximate arbitrary shapes of spatial envelopes in real time.
- 5) Validation of the proposed MPC formulation in racing, emergency collision avoidance and off-road environments.

The rest of the paper is organized as follows. Sec. II reviews the relevant literature. Sec. III-A describes the 3 DoF single-track vehicle dynamics. Sec. III-B describes the MPC formulation including the conservative spatial envelope constraints. Sec. III-C describes the real-time spatial envelope planner. Sec. IV describes the model fidelity test. The results and discussion of the proposed spatial envelope MPC are presented in Sec. IV. From Sec. V-A to E, the simulation and experimental results of spatial envelope MPC are conducted and analyzed in multiple scenarios. In Sec. III-F, the proposed spatial envelope planning technique is presented and discussed. Finally, Sec. VI concludes the study.

II. BACKGROUND

A conventional autonomy stack typically divides the navigation task into two layers. The first layer focuses on the generation of a collision-free path or trajectory as a reference, whereas the second layer is designed to track this reference [15].

One common approach for the first layer is graph search. In [16], an improved A* algorithm is used to plan a smooth path for the vehicle in urban areas. To tackle dynamic environments, D* [17] is used to reuse previous information. To account for nonholonomic constraints, the hybrid-state A* [18] uses a kinematic model to expand the graph and the Reeds-Sheeps curve as the heuristic [19]. However, the size of the graph increases exponentially with higher resolutions, hindering scalability. In [20], the Rapidly Exploring Randomized Tree (RRT), a type of incremental sampling method, gradually constructs a tree graph towards the goal point based on heuristics and randomized sampling. In [21], RRT* introduces a rewiring function to achieve asymptotic optimality. Due to the flexibility of the steering function, several extensions of the RRT-based algorithm have been developed, such as incorporating a kinematic vehicle model in [22] and considering dynamic constraints in [23]. Although RRT-based methods have been widely adopted, their inherent sampling nature creates a trade-off between computation time and solution quality, and the

solution is not repeatable. Thus, despite their success and widespread use in motion planning, graph search methods suffer from issues such as lack of considerations in dynamic constraints, the curse of dimensionality, and nonrepeatability.

To fully exploit the mobility of the vehicle, a dynamic model can be leveraged to understand its limits. In [12], a linear single-track model is utilized to constrain the vehicle's motion within the stable region, assuming constant longitudinal speed and small slip angles. In [24], a linear model incorporating load transfer is developed to address the lane-keeping problem using curvilinear coordinates along the lane. In [25], a similar model is employed in a racing application, but it is nonlinear because the longitudinal speed is not assumed to be constant. Curvilinear coordinates allow for the natural expression of race progression and track boundaries. However, the model's accuracy is compromised when the vehicle significantly deviates from the reference line. In [26], the friction circle limit is considered by bounding the maximum lateral tire force within the Fiala brush tire model despite the fact that this approach can introduce numerical issues during the optimization process. Thus, existing vehicle models either oversimplify the dynamics or lack computational suitability for a reference-free framework.

Model predictive control (MPC) effectively harnesses the benefits of having a dynamical model to optimize the vehicle's trajectory and integrate this knowledge with the desired costs and constraints. Its capability to consider obstacle constraints naturally delineates the drivable areas on the map so that it can plan safe and optimal trajectories. In [27]–[29], MPC is integrated with obstacle avoidance constraints to ensure safe vehicle operation. To precisely formulate the hazard zones, researchers utilized exact polygons to define collision avoidance constraints, thereby maximizing the traversable area [30]. Expanding on static obstacle avoidance scenarios, moving obstacle avoidance has also been addressed [31], [32]. Beyond typical on-road driving scenarios, MPC has also been employed to accommodate a terramechanics-based vehicle model for navigating off-road environments without predefined trajectories [9]. Despite the success MPC has achieved in this field, these solutions have only been validated in relatively simple scenarios and do not scale well with the increasing complexity that vehicles might encounter in real-world environments.

To avoid these challenges, a conventional autonomy stack may choose to employ MPC only as a tracker in the second layer. In studies such as [24], [33], linear MPC is used to track the vehicle along a given trajectory. To further enhance the mobility of the vehicle model, nonlinear MPC, which incorporates nonlinear vehicle dynamics, is introduced to handle vehicle control [12], [34], [35]. The maturity of this technique has contributed to the delivery of state-of-the-art, high-performance autonomous driving solutions. In [36], tire forces between longitudinal and lateral directions are coordinated using nonlinear MPC with a Frenet coordinate-based vehicle model to track a given trajectory. Building on this, researchers adopted a hierarchical approach in which the predictive control layer integrates a single-track model with longitudinal weight transfer dynamics, while the chassis

control layer manages lateral brake distribution to track a pre-defined racing trajectory. [37]. This divide-and-conquer strategy has demonstrated significant advantages in scalability by addressing collision concerns in the planning layer and limiting the controller to trajectory following. However, it also raises new issues. Namely, the framework's performance depends on the quality of the path produced by the high-level planner. Thus, planning optimal and dynamically feasible trajectories in real-time remains a challenge.

To retain the scalability benefits offered by the trajectory tracking method without overly restricting the vehicle's operating region, researchers developed a spatial envelope based MPC method that explicitly defines the drivable area as spatial constraint for the vehicle [13], [38], [39]. Rather than informing the MPC of hazard zones, this method constrains the MPC to plan trajectories exclusively within the predefined safe area. The advantage of prescribing a safe area lies in its ability to effectively reduce the number of constraints needed to enforce the safety requirements. Researchers in [14], [40] employ this concept, adopting both linear and nonlinear MPC for emergency collision maneuvers, whereas the work in [5] extends the envelope-based approach to curved roads for broader applicability. However, the envelope-based MPC discretizes the safe driving area into segments, constraining trajectory points to remain within a single segment. This approach requires identifying the relevant segment for each specific trajectory point prior to solving the MPC, which presents a problem, because segment identification is neither a continuous nor differentiable process. Consequently, these methods often assume a fixed speed, further restricting the vehicle's operating space by disallowing longitudinal maneuvers. This limitation significantly hinders the practical application of this concept.

Another problem to address with spatial envelope MPC is the design of the envelopes. In [41], researchers defined a drivable envelope as a set of polygonal sections within which every trajectory remains inside the safety region. These envelopes were defined heuristically based on a pre-planned trajectory. However, this method does not encompass the entire drivable region along the route. In [5], researchers sampled parallelograms from road boundaries to form the drivable envelopes. Nevertheless, the manual determination of the sampling interval restricts the applicability of this approach to diverse road conditions. For optimizing polygonal shapes within a given boundary, similar efforts have been made in the field of geometric optimization. For example, algorithms have been developed to find the largest area rectangle within a given contour, and methods for solving the rectangle packing problem have been proposed [42]–[45]. However, these algorithms either cannot ensure that the resulting rectangles intersect or cannot provide solutions for rectangles of varying sizes, limitations that are problematic for planning polygons for drivable envelopes.

In contrast to the conventional autonomy stack architecture, the end-to-end approach has become another trend due to rapid developments in machine learning. In [46], the PilotNet CNN architecture is proposed for autonomous driving. The PilotNet directly models the relationship between input images and

steering values. However, the CNN architecture cannot capture temporal information. In [47], a deep neural network (DNN) called AdmiralNet, a convolutional neural network (CNN) integrated with a Long Short-Term Memory (LSTM) network, is used to handle time-varying information and is trained on annotated data generated by human operators. However, employing humans to generate training data is expensive, especially for a DNN, which requires a large amount of data. Instead of treating DNN as a regression tool, additional research efforts have focused on the field of reinforcement learning. In [48] and [49], Deep Deterministic Policy Gradient (DDPG), one of the most representative actor-critic (AC) algorithms, is used to learn the policies. Although DDPG is effective for continuous control, it suffers from the overestimation of Q values. To address this issue, Soft Actor-Critic (SAC) is proposed in [50], which adopts a maximum entropy framework to encourage exploration and improve robustness. However, for the end-to-end approach, interpreting the learned policy is still an open question. Additionally, performance is not guaranteed when out-of-distribution events occur due to its data-driven nature.

Based on the literature review above, a key gap has been identified: the absence of a real-time reference-free trajectory planning and control framework that can push the vehicle to its limits while remaining scalable to a diverse set of scenarios. Bridging this gap is essential for advancing high-performance autonomous driving and is the goal of this work.

III. METHODS

A. Vehicle Dynamics

In this section, a single-track vehicle model is introduced that effectively captures the essential dynamics in highly dynamic maneuvers. The model is also specifically tailored for integration with optimization-based motion controllers, significantly enhancing its real-time capability. The model is illustrated in Fig. 1-A.

The state and control vectors are defined as:

$$\xi := \begin{bmatrix} x \\ y \\ v \\ r \\ \psi \\ u_x \\ \delta_f \\ a_x \end{bmatrix} = \begin{bmatrix} \text{global x position of C.G. point} \\ \text{global y position of C.G. point} \\ \text{lateral speed} \\ \text{yaw rate} \\ \text{yaw angle} \\ \text{longitudinal speed} \\ \text{front tire steering angle} \\ \text{longitudinal acceleration} \end{bmatrix} \quad (1)$$

$$\zeta := \begin{bmatrix} \dot{\delta}_f \\ j_x \end{bmatrix} = \begin{bmatrix} \text{front tire steering rate} \\ \text{longitudinal jerk} \end{bmatrix} \quad (2)$$

The vehicle dynamics are represented in state space as:

$$\dot{\xi} = \mathcal{V}[\xi(t), \zeta(t)] = A(\xi) + B\zeta \quad (3)$$

$$A(\xi) = \begin{bmatrix} u_x \cos \psi - v \sin \psi \\ u_x \sin \psi + v \cos \psi \\ ((F_{yf} \cos \delta_f + F_{xf} \sin \delta_f + F_{yr})/M - u_x r) \\ ((F_{yf} \cos \delta_f + F_{xf} \sin \delta_f)L_f - F_{yr}L_r)/I_{zz} \\ r \\ a_x + rv - \frac{F_{yf} \sin \delta_f}{M} \\ 0 \\ 0 \end{bmatrix} \quad (4)$$

$$B^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Here M is the vehicle total mass, I_{zz} is the moment of inertia, and L_f and L_r are distances from the vehicle center of gravity to the vehicle front and rear axles, respectively. F_{yf} and F_{yr} are the lateral forces of the front and rear tires. F_{xf} is the longitudinal force of the front tire. Note that a_x is defined as $\frac{F_x}{M}$ instead of $\frac{du_x}{dt}$, where F_x is the total tire force along the longitudinal direction. By this definition, longitudinal load transfer is expressed in a linear form, which speeds up the computation of lateral tire forces and hard constraints for the friction circle. The vehicle is assumed to be rear-wheel driven, with braking applied on both front and rear wheels with a specific braking ratio b_r . Thus, the front-to-rear distribution of longitudinal tire forces is expressed as follows:

$$F_{xf} = \begin{cases} 0 & \text{if } F_x \geq 0 \\ b_r F_x & \text{if } F_x < 0 \end{cases} \quad (6)$$

$$F_{xr} = \begin{cases} F_x & \text{if } F_x \geq 0 \\ (1 - b_r) F_x & \text{if } F_x < 0 \end{cases} \quad (7)$$

where F_{xf} and F_{xr} are the front and rear longitudinal tire forces, respectively. However, these expressions are not suitable for optimal control due to the if-condition not being differentiable. Therefore, a proximal representation of the if-condition is employed using the sigmoid function:

$$\mathbb{1} = (1 - \frac{1}{1 + e^{-p_f F_x}}) \quad (8)$$

$$F_{xf} = \mathbb{1} \cdot b_r F_x \quad (9)$$

$$F_{xr} = F_x - F_{xf} \quad (10)$$

where p_f is a tunable parameter that controls the sharpness of the transition between the **if** and **else** conditions. It is also used in Eq. (15). The longitudinal load transfer is expressed as follows:

$$F_{zf} = \frac{L_r}{L_f + L_r} Mg - K_z a_x \quad (11)$$

$$F_{zr} = \frac{L_f}{L_f + L_r} Mg + K_z a_x \quad (12)$$

$$K_z = \frac{Mgh}{L_f + L_r} \quad (13)$$

where g is the gravitational constant.

By derating the lateral tire force with the longitudinal tire force as discussed in [51], the formulation is structured such that the longitudinal tire force required by a_x is always satisfied first and the maximum lateral force is then derived considering that the combined total tire force cannot exceed

the maximum allowable friction force (see Fig. 1-B1). The maximum lateral tire force could thus be expressed as:

$$F_{y\bullet\max} = \sqrt{1 - \left(\frac{F_{x\bullet}}{\mu_\bullet F_{z\bullet}} \right)^2} \mu_\bullet F_{z\bullet} \quad (14)$$

where \bullet is a placeholder for f, r , representing the front and rear axles, respectively, and μ is the friction coefficient. However, during the initialization of the optimization process, it is possible that the magnitude of $\mu_\bullet F_{z\bullet}$ becomes smaller than $F_{x\bullet}$, leading to numerical errors in the solver. To avoid this issue, a softplus function is applied prior to taking the square root:

$$F_{y\bullet\max} = \sqrt{\frac{\log(1 + e^{p_f(1 - (\frac{F_{x\bullet}}{\mu_\bullet F_{z\bullet}})^2)})}{p_f}} \mu_\bullet F_{z\bullet} \quad (15)$$

When $|\mu_\bullet F_{z\bullet}| > |F_{x\bullet}|$, Eq. (15) is approximately identical to Eq. (14). When $|\mu_\bullet F_{z\bullet}| < |F_{x\bullet}|$, $F_{y\bullet\max} \approx 0$.

A sigmoid function is used to relate the lateral tire force $F_{y\bullet}$ and the slip angle α_\bullet as depicted in Fig. 1-B2. The analytical expression is shown below:

$$F_{y\bullet} = -2F_{y\bullet\max} \left(\left(1 + e^{\frac{-2C_{a\bullet}\alpha_\bullet}{F_{y\bullet\max}}} \right)^{-1} - 0.5 \right) \quad (16)$$

where $C_{a\bullet}$ denotes the tire cornering stiffness. The sigmoid function captures the key considerations of the Fiala tire model used in [37]. Namely, the slope around the linear region ($\alpha_\bullet \ll 1$) is $C_{a\bullet}$ and the magnitude is bounded by $F_{y\bullet\max}$. With this formulation, the vehicle dynamics achieves C^2 continuity, which the optimizer favors, thus significantly improving computational speed in optimization.

B. Spatial Envelope Model Predictive Control (Spatial Envelope MPC)

The nonlinear Model Predictive Control (MPC) framework is formulated such that the controller is capable of optimizing and tracking the trajectories in real time without any predefined trajectories. It directly outputs the control commands to the vehicle.

The optimal control problem (OCP) solved within the MPC over a receding horizon is given as follows.

$$\underset{\xi, \zeta}{\text{minimize}} \quad J = \int_{t_0}^{t_f} \mathcal{I}[\xi(t), \zeta(t)] dt + \mathcal{G}[\xi(t_f)] \quad (17)$$

$$\text{subject to} \quad \dot{\xi}(t) = \mathcal{V}[\xi(t), \zeta(t)] \quad (18)$$

$$\xi_{\min} \leq \xi(t) \leq \xi_{\max} \quad (19)$$

$$\zeta_{\min} \leq \zeta(t) \leq \zeta_{\max} \quad (20)$$

$$\mathcal{T}_{\mathcal{F}}[\xi(t)] \leq \mathbf{0} \quad (21)$$

$$\mathcal{E}[\xi(t)] \leq \mathbf{0} \quad (22)$$

J represents the cost function, \mathcal{I} denotes the stage cost for each point along the predicted trajectory, and \mathcal{G} signifies the cost-to-go term at the final trajectory point. The Optimal Control Problem (OCP) is constrained to adhere to the vehicle dynamics \mathcal{V} , while the states ξ and controls ζ are bounded as specified in Eq. (19) and (20). $\mathcal{T}_{\mathcal{F}}$ represents the constraints on the tire forces. \mathcal{E} imposes spatial constraints to ensure the vehicle remains within the safe envelope. All these terms are detailed next.

1) Hard Constraints: Three types of hard constraints are considered in the MPC. Namely, linear state and control constraints, tire force constraints, and spatial envelope constraints.

Linear State and Control Constraints: Three states are constrained with constant lower and upper bounds to ensure the safe operation of the vehicle:

$$v_{\min} \leq v \leq v_{\max} \quad (23)$$

$$r_{\min} \leq r \leq r_{\max} \quad (24)$$

$$\delta_{f,\min} \leq \delta_f \leq \delta_{f,\max} \quad (25)$$

Another constraint is applied on the acceleration state to capture the fact that as the speed increases, the maximum longitudinal acceleration the vehicle can achieve becomes smaller due to the engine power limits. The typical behavior of a straight-line acceleration test is illustrated in Fig. 1-C1. The shape is close to a saturating exponential function, characterized by the saturating value p_b and time constant $\frac{1}{p_a}$. Thus, the vehicle's acceleration capability is approximated as a linear constraint on a_x as a function of u_x as expressed below and depicted in Fig. 1-C2:

$$a_x \leq -p_a(u_x - p_b) \quad (26)$$

The control bounds are set to capture the actuator limits:

$$\dot{\delta}_{f,\min} \leq \dot{\delta}_f \leq \dot{\delta}_{f,\max} \quad (27)$$

$$j_{x,\min} \leq j_x \leq j_{x,\max} \quad (28)$$

Tire Force Constraint: As shown in Sec. III-A, the friction circle is naturally embedded in Eq. (14) and is further approximated in Eq. (15). Therefore, the original friction circle constraint

$$F_{x,\bullet}^2 + F_{y,\bullet}^2 \leq (\mu_{\bullet} F_{z,\bullet})^2 \quad (29)$$

can be now turned into

$$|F_{x,\bullet}| \leq \mu_{\bullet} F_{z,\bullet} \quad (30)$$

Eq. (30) is expanded based on whether the vehicle is accelerating or braking based on Eq. (6), (7), (11) and (12). Specifically, for the acceleration case, we have:

$$\begin{aligned} 0 &\leq \mu_f F_{z,f} \\ Ma_x &\leq \mu_r \left(\frac{L_f}{L_f + L_r} Mg + K_z a_x \right) \end{aligned} \quad (31)$$

From this equation, an upper bound for a_x is derived:

$$a_x \leq \min \left(\frac{L_r Mg}{(L_f + L_r) K_z}, \frac{\mu_r Mg \frac{L_f}{L_f + L_r}}{M - \mu_r K_z} \right) \quad (32)$$

Similarly, for the braking case we have:

$$\begin{aligned} -b_r Ma_x &\leq \mu_f \left(\frac{L_r}{L_f + L_r} Mg - K_z a_x \right) \\ -(1 - b_r) Ma_x &\leq \mu_r \left(\frac{L_f}{L_f + L_r} Mg + K_z a_x \right) \end{aligned} \quad (33)$$

This leads to the following lower bound for a_x :

$$a_x \geq \max \left(\frac{-\mu_r Mg \frac{L_f}{L_f + L_r}}{M(1 - b_r) + \mu_r K_z}, \frac{-\mu_f Mg \frac{L_r}{L_f + L_r}}{Mb_r - \mu_f K_z} \right) \quad (34)$$

Thus, the original nonlinear friction circle constraint Eq. (29) is reduced to a set of linear constraints on longitudinal acceleration a_x , greatly improving the computational efficiency of the MPC.

Spatial Envelope Constraints: The spatial envelope is defined as the region through which the vehicle can traverse safely. In general, the envelope can take any shape. Accordingly, no assumption is made about the envelope shape in this study. Thus, methods like using a cubic function to fit the lane boundary in [52] are not suitable, because they require the global X and Y positions to have a one-to-one relationship.

In this work, the need to account for an arbitrary envelope shape is addressed by segmenting it into a sequence of blocks. Fig. 1-D1 illustrates an example, where a track segment is divided into three blocks. The constraint for the vehicle to stay within one of the blocks is defined as follows:

$$g_b^j(\xi) = d_b^j(\xi) - 1 \leq 0 \quad (35)$$

$$\Delta x_b^j = x - x_b^j \quad (36)$$

$$\Delta y_b^j = y - y_b^j \quad (37)$$

$$\begin{aligned} d_b^j(\xi) = & \left(\left(\frac{\cos(\psi_b^j) \Delta x_b^j + \sin(\psi_b^j) \Delta y_b^j}{L_b^j} \right)^p \right. \\ & \left. + \left(\frac{\cos(\psi_b^j) \Delta y_b^j - \sin(\psi_b^j) \Delta x_b^j}{W_b^j} \right)^p \right)^{\frac{1}{p}} \end{aligned} \quad (38)$$

Here, x and y represent the x and y positions of the vehicle's center of gravity (C.G.) and are the entries of ξ . The variables x_b^j , y_b^j , ψ_b^j , L_b^j , and W_b^j correspond to the block center's x-y positions, block yaw angle, half block length, and half block width, respectively. The superscript j represents the j^{th} block. The parameter p defines the p-norm, which is set to 4 to make hyperellipses. A higher value of p can make the block more closely resemble a rectangle, but it increases the chance of causing ill-conditioning in the optimization. The satisfaction of Eq. (35) indicates that the vehicle is inside the specific block, shown as the teal region in Fig. 1-D2. If the vehicle is inside any of these blocks, the condition is equivalent to constraining it inside the entire envelope, shown as the teal region in Fig. 1-D3. The expression for the OR operation over multiple block constraints is discussed below.

Lemma 1. Let there be n conditions in the form of $g^j(\xi) \leq 0$, $j = 1, 2, \dots, n$. Then, the OR operation, where at least one of the conditions $g^j(\xi)$ is satisfied, can be expressed as a single inequality constraint as follows:

$$g_{\min}(\xi) = \min\{g^1(\xi), g^2(\xi), \dots, g^n(\xi)\} \leq 0 \quad (39)$$

Proof. Suppose that at least one of the conditions is satisfied. Then, $\exists j^* \in [1, n]$, $g^{j^*}(\xi) \leq 0$. Since $g_{\min}(\xi)$ represents the minimum value among the expressions $g^j(\xi)$, it follows that $g_{\min}(\xi) \leq g^{j^*}(\xi) \leq 0$.

Conversely, assume that Eq. (39) is satisfied, but none of the conditions $g^j(\xi)$ are satisfied. This leads to a contradiction, since $g_{\min}(\xi) \leq 0$ implies that $\exists j^* \in [1, n]$ such that $g^{j^*}(\xi) = g_{\min}(\xi) \leq 0$. \square

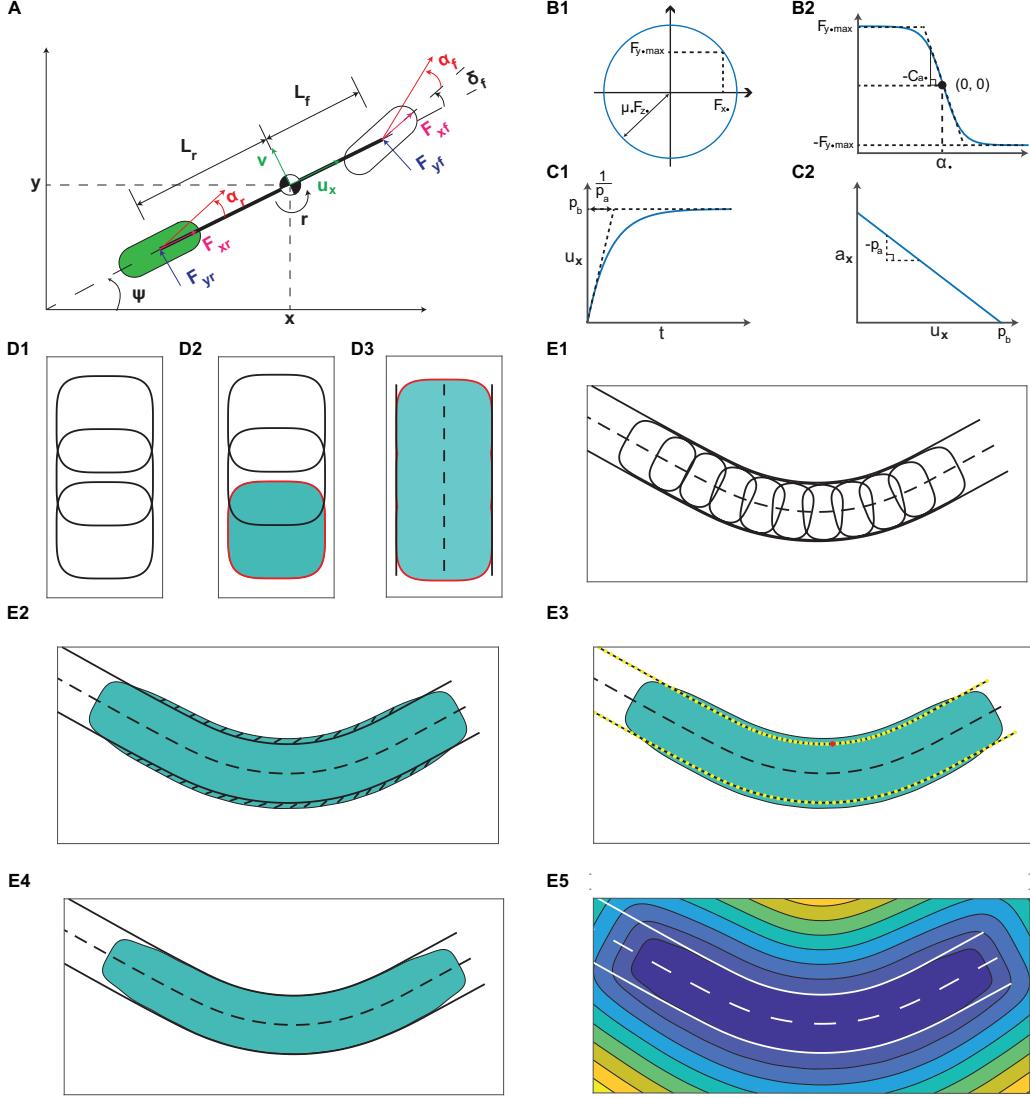


Fig. 1. The plots for vehicle dynamics, tire forces, acceleration constraints, spatial envelope hard constraints, and spatial envelope cost (A) Free-body diagram of the three-DoF dynamical single-track model. The green tire means the vehicle is rear-wheel driven. (B1) The friction circle limit. (B2) The sigmoid representation of the lateral tire force. (C1) Longitudinal speed versus time in a straight acceleration test. (C2) Maximum acceleration deduced from the straight acceleration test. (D1) Three block constraints stacked together. (D2) The feasible set for one block constraint. (D3) The feasible set when any one of the block constraints is satisfied. The shape of the teal region forms a lane. (E1) A portion of the race track. The track is segmented by ten blocks. (E2) A contour plot of KS aggregation over the blocks, where the teal region is the feasible set. The hatched area is the critical set Ω_c . (E3) The yellow dots are the points on lane boundaries and the red point has the minimal LogSumExp value. (E4) The feasible set given by the conservative spatial envelope hard constraint. (E5) A contour plot of the spatial envelope cost.

Using Lemma 1, the expression for the vehicle staying inside the envelope is given by:

$$g_{\min, b}(\xi) = \min\{g_b^1(\xi), g_b^2(\xi), \dots, g_b^{n_{\text{block}}}(\xi)\} \leq 0 \quad (40)$$

where n_{block} is the total number of blocks. Note that Eq. (40) is not differentiable due to the minimization operation on the functions. Therefore, a smooth minimization technique is required to closely approximate $g_{\min, b}(\xi)$ while ensuring it is twice continuously differentiable.

One candidate is the Kreisselmeier–Steinhauser (KS) aggregation [53] used in constrained optimization, which is also referred to as LogSumExp (LSE) [54] in machine learning. In the context of constrained optimization, KS aggregation is used to conservatively estimate the maximum. In the context of machine learning, the LSE function is used for both maximum

and minimum approximations. However, the conservativeness property is not guaranteed for the minimum approximation. Thus, an expression for the spatial envelope constraint with conservativeness guarantee is derived next.

Lemma 2. *Given two sets $\Omega^i = \{\xi | g^i(\xi) \leq 0\}$ and $\Omega^j = \{\xi | g^j(\xi) \leq 0\}$, if $g^i(\xi) \leq g^j(\xi) \forall \xi$, then $\Omega^j \subseteq \Omega^i$.*

Proof. By definition, $\forall \xi \in \Omega^j$, $g^j(\xi) \leq 0$. Given that $g^i(\xi) \leq g^j(\xi) \forall \xi$, it follows that $g^i(\xi) \leq 0 \forall \xi \in \Omega^j$. Therefore, if $\xi \in \Omega^j$, then $\xi \in \Omega^i$. This implies $\Omega^j \subseteq \Omega^i$. \square

Lemma 3. *Given a set of functions $G = \{g^j(\xi), j \in [1, n]\}$, consider the function $g_{\max}(\xi)$ defined as follows:*

$$g_{\max}(\xi) = \max\{g^1(\xi), g^2(\xi), \dots, g^n(\xi)\} \leq 0 \quad (41)$$

The LSE over the set G with positive ρ_{LSE} satisfies the following properties:

$$g_{\max}(\xi) \leq g_{\text{LSE}}(\xi) \leq g_{\max}(\xi) + \frac{\ln(n)}{\rho_{\text{LSE}}} \quad (42)$$

where $g_{\text{LSE}}(\xi)$ is the expression of LSE, and the first inequality is strict unless $n = 1$.

$$g_{\text{LSE}}(\xi) = \frac{1}{\rho_{\text{LSE}}} \ln \left(\sum_{j=1}^n e^{\rho_{\text{LSE}} g^j(\xi)} \right), \forall \rho_{\text{LSE}} > 0 \quad (43)$$

and ρ_{LSE} is the tuning parameter.

Proof. Define $M = g_{\max}(\xi)$ and rewrite the sum inside the logarithm in (43):

$$\begin{aligned} g_{\text{LSE}}(\xi) &= \frac{1}{\rho_{\text{LSE}}} \ln \left(\sum_{j=1}^n e^{\rho_{\text{LSE}} g^j(\xi)} \right) \\ &= \frac{1}{\rho_{\text{LSE}}} \ln \left(\sum_{j=1}^n \left(e^{\rho_{\text{LSE}}(g^j(\xi) - M)} e^{\rho_{\text{LSE}} M} \right) \right) \\ &= \frac{1}{\rho_{\text{LSE}}} \left[\ln \left(\sum_{j=1}^n e^{\rho_{\text{LSE}}(g^j(\xi) - M)} \right) + \rho_{\text{LSE}} M \right] \\ &= M + \frac{1}{\rho_{\text{LSE}}} \ln \left(\sum_{j=1}^n e^{\rho_{\text{LSE}}(g^j(\xi) - M)} \right) \end{aligned} \quad (44)$$

Since $g^j(\xi) - M \leq 0$ for all j , and there exists at least one j such that $g^j(\xi) = M$, we have:

- $e^{\rho_{\text{LSE}}(g^j(\xi) - M)} \in (0, 1]$
- $\sum_{j=1}^n e^{\rho_{\text{LSE}}(g^j(\xi) - M)} \in [1, n]$

Thus,

$$\ln \left(\sum_{j=1}^n e^{\rho_{\text{LSE}}(g^j(\xi) - M)} \right) \in [0, \ln n] \quad (45)$$

Note that Eq. (45) becomes 0 only when $n = 1$. Based on Eq. (45), the following inequalities are derived.

$$g_{\max}(\xi) \leq g_{\text{LSE}}(\xi) \leq g_{\max}(\xi) + \frac{\ln n}{\rho_{\text{LSE}}} \quad (46)$$

which proves inequality (42). The closed-form expression for $g_{\text{LSE}}(\xi)$ is given directly in (43) by definition. \square

Lemma 4. Given the function set defined in Lemma 3 and g_{\min} defined in Eq. (39), the LSE over the set G with negative ρ_{LSE} satisfies the following properties:

$$g_{\min}(\xi) + \frac{\ln(n)}{\rho_{\text{LSE}}} \leq g_{\text{LSE}}(\xi) \leq g_{\min}(\xi) \quad (47)$$

$$g_{\text{LSE}}(\xi) = \frac{1}{\rho_{\text{LSE}}} \ln \left(\sum_{j=1}^n e^{\rho_{\text{LSE}}(g^j(\xi))} \right), \forall \rho_{\text{LSE}} < 0 \quad (48)$$

where the second inequality in Eq. (47) is strict unless $n = 1$.

Proof. Given the function set G and a negative ρ_{LSE} , define the negated function set \tilde{G} , and the negated tuning parameter $\tilde{\rho}_{\text{LSE}}$:

$$\tilde{G} = \{\tilde{g}^j(\xi), j \in [1, n]\} \quad (49)$$

$$\tilde{g}^j(\xi) = -g^j(\xi), j \in [1, n] \quad (50)$$

$$\tilde{\rho}_{\text{LSE}} = -\rho_{\text{LSE}} > 0 \quad (51)$$

Based on Lemma 3, the LSE over \tilde{G} obeys the following inequalities:

$$\tilde{g}_{\max}(\xi) \leq \tilde{g}_{\text{LSE}}(\xi) \leq \tilde{g}_{\max}(\xi) + \frac{\ln(n)}{\tilde{\rho}_{\text{LSE}}}, \forall \tilde{\rho}_{\text{LSE}} > 0 \quad (52)$$

$$\tilde{g}_{\text{LSE}}(\xi) = \frac{1}{\tilde{\rho}_{\text{LSE}}} \ln \left(\sum_{j=1}^n e^{\tilde{\rho}_{\text{LSE}}(\tilde{g}^j(\xi))} \right) \quad (53)$$

$$\tilde{g}_{\max}(\xi) = \max\{\tilde{g}^1(\xi), \tilde{g}^2(\xi), \dots, \tilde{g}^n(\xi)\} \quad (54)$$

Note that the first inequality in Eq. (52) is strict unless $n = 1$. Based on Eq. (50), the following relationship is derived:

$$\tilde{g}_{\max}(\xi) = -g_{\min}(\xi) \quad (55)$$

Based on Eq. (55) and Eq. (52), the property of LSE over G with negative ρ_{LSE} is derived:

$$g_{\min}(\xi) + \frac{\ln(n)}{\rho_{\text{LSE}}} \leq -\tilde{g}_{\text{LSE}}(\xi) = g_{\text{LSE}}(\xi) \leq g_{\min}(\xi) \quad (56)$$

where the second inequality is strict unless $n = 1$. \square

Similar proofs for Lemma 3 and 4 have been presented in [55] for control barrier function constraint aggregation. Nevertheless, these steps are retained here for completeness and consistent notation throughout the paper.

Lemma 5. Consider the safety set Ω_s defined as $\Omega_s = \{\xi | g_{\min}(\xi) \leq 0\}$; the feasible set Ω_0 defined as $\Omega_0 = \{\xi | g_{\text{LSE}}(\xi) \leq 0\}$; and the critical set defined as $\Omega_c = \Omega_s^C \cap \Omega_0$. If $\Omega_c \neq \emptyset$, then $\Omega_0 \not\subseteq \Omega_s$. Otherwise, $\Omega_0 \subseteq \Omega_s$.

Proof. If $\Omega_c \neq \emptyset$, then $\exists \xi \in \Omega_0$ such that $\xi \notin \Omega_s$, implying $\Omega_0 \not\subseteq \Omega_s$.

If $\Omega_c = \emptyset$, then $\forall \xi \in \Omega_0$, $\xi \in \Omega_s$, implying $\Omega_0 \subseteq \Omega_s$. \square

Theorem 1. Let Ω_s , Ω_0 , and Ω_c be defined as in Lemma 5. Let $\tilde{\Omega}_0$ be defined as

$$\tilde{\Omega}_0 = \{\xi | g_{\text{LSE}}(\xi) - \epsilon_0 < 0\} \quad (57)$$

$$\epsilon_0 = \min_{\xi \in \Omega_c} g_{\text{LSE}}(\xi) \quad (58)$$

Then, if $\Omega_c \neq \emptyset$, then $\tilde{\Omega}_0 \subseteq \Omega_s$.

Proof. By the definitions of Ω_c and Ω_0 , $\forall \xi \in \Omega_c$, $\xi \in \Omega_0$, and thus $\forall \xi \in \Omega_c$, $g_{\text{LSE}}(\xi) \leq 0$. This implies that $\epsilon_0 \leq 0$ in Eq. (58). Based on Lemma 2, $\tilde{\Omega}_0 \subseteq \Omega_0$ because $g_{\text{LSE}}(\xi) \leq g_{\text{LSE}}(\xi) - \epsilon_0$. Define a new critical set as $\tilde{\Omega}_c = \Omega_s^C \cap \tilde{\Omega}_0$. The new set satisfies $\tilde{\Omega}_c \subseteq \Omega_c$ because $\tilde{\Omega}_0 \subseteq \Omega_0$. According to the definition of ϵ_0 in Eq. (58), ϵ_0 is the minimal LSE value in the set Ω_c , implying $g_{\text{LSE}}(\xi) \geq \epsilon_0 \forall \xi \in \Omega_c$. Since $\tilde{\Omega}_c \subseteq \Omega_c$, it follows that $\forall \xi \in \tilde{\Omega}_c$, $\xi \in \Omega_c$, and thus $g_{\text{LSE}}(\xi) \geq \epsilon_0 \forall \xi \in \tilde{\Omega}_c$. Since $(g_{\text{LSE}}(\xi) \geq \epsilon_0) \cap (g_{\text{LSE}}(\xi) < \epsilon_0) = \emptyset$, it follows that

$\tilde{\Omega}_c \cap \tilde{\Omega}_0 = \emptyset$. If $\tilde{\Omega}_0 = \emptyset$, then $\tilde{\Omega}_0 \subseteq \Omega_s$. Otherwise, $\tilde{\Omega}_c = \emptyset$, and according to Lemma 5, $\tilde{\Omega}_0 \subseteq \Omega_s$. \square

Theorem 1 proves that subtracting ϵ_0 ensures the new feasible set is strictly contained within the safety set, which is essential for maintaining safety. The value ϵ_0 is defined as the minimum of the LSE function over the critical set. The critical set is defined as the region outside the safety set but within the original feasible set specified by the LSE function.

A portion of a racetrack is shown in Fig. 1-E1 through E4 to demonstrate the conservativeness of the spatial envelope constraint. In Fig. 1-E1, the blocks are used to segment the envelope. Each block overlaps with at least one other block to ensure connectivity. Based on Lemma 4, the LSE is used to approximate the function $g_{\text{min},b}(\xi)$ defined by Eq. (40) as follows:

$$g_{\text{lse},b}(\xi) = \frac{1}{\rho_{\text{LSE}}} \ln \left(\sum_{j=1}^{n_{\text{block}}} e^{\rho_{\text{LSE}}(g_b^j(\xi))} \right), \quad \forall \rho_{\text{LSE}} < 0 \quad (59)$$

The feasible region defined by $g_{\text{lse},b}(\xi) \leq 0$ is depicted as the teal region in Fig. 1-E2. Although the blocks are within the lane boundaries, the feasible region described by Eq. (59) extends beyond these boundaries. According to Theorem 1, the critical set Ω_c must be identified first. The critical set lies outside the lane boundaries but within the feasible set, as illustrated by the hatched area in Fig. 1-E2. Within Ω_c , the minimal LSE value ϵ_0 is identified, and the minimizer is marked with a red dot in Fig. 1-E3. Note that in this work, in all scenarios considered, the turning of the lane is not sharp, and the lane width is large enough. By construction, the LSE attains its minima on the safety boundary (lane boundaries); hence the minimizer over the critical set lies on that boundary segment. Therefore, in this work, only the lane boundary points are evaluated to find the minimal LSE value. These boundary points are depicted as yellow dots in Fig. 1-E3. Then, the conservative spatial envelope constraint is expressed as follows:

$$g_{\text{envelope}}(\xi) = g_{\text{lse},b}(\xi) - \epsilon_0 < 0 \quad (60)$$

The resultant feasible set is shown as the teal region in Fig. 1-E4. The feasible region lies within the lane boundaries, thereby illustrating the conservativeness of the developed spatial envelope constraint.

During optimization, the states are treated as discrete rather than continuous design variables. Consequently, the i^{th} prediction time is denoted as $\mathcal{T}(i)$, and the corresponding state is expressed as $\xi(\mathcal{T}(i))$. Thus, the following conservative spatial envelope constraint is applied in the optimization.

$$g_{\text{envelope}}(\mathcal{T}(i)) = \frac{1}{\rho_{\text{LSE}}} \ln \left(\sum_{j=1}^{n_{\text{block}}} e^{\rho_{\text{LSE}}(g_b^j(\xi(\mathcal{T}(i))))} \right) - \epsilon_0 < 0 \quad (61)$$

2) *Cost Function*: : The cost function in Eq. (17) is further expanded as:

$$J = J_{\text{state}} + J_{\text{control}} + J_{\text{envelope}} + J_{\text{specific}} \quad (62)$$

State Cost (J_{state}) is the cost term for vehicle states:

$$J_{\text{state}} = \int_{t_0}^{t_f} \left(w_{\delta_f} \delta_f^2 + w_{a_x} a_x^2 + w_v v^2 + w_\kappa \left(\frac{r}{u_x} \right)^2 \right) dt \quad (63)$$

The first two terms regulate the control inputs (steering, acceleration) to discourage excessive steering and acceleration, as in racing the vehicle should drive as straight and as close to its top speed as possible to minimize the lap time. The third term aims to minimize the vehicle's lateral velocity to prevent the vehicle from sliding as a safety and stability consideration. The last term minimizes the path curvature, again in line with the goal of driving as straight as possible to allow for maximum speed.

Control Cost (J_{control}) is expressed as:

$$J_{\text{control}} = \int_{t_0}^{t_f} \left(w_{\dot{\delta}_f} \dot{\delta}_f^2 + w_{\dot{j}_x} \dot{j}_x^2 \right) dt \quad (64)$$

The two terms in the expression minimize the rate of change of control inputs to ensure smooth steering and acceleration.

Spatial Envelope Cost (J_{envelope}) brings the vehicle inside the envelope to ensure safety. It is given as:

$$J_{\text{envelope}} = \sum_{i=1}^{n_{\text{ds}}} w_{\text{envelope}} \log \left(1 + e^{-\theta_{\text{hp}}(g_{\text{sm}} + g_{\text{envelope}}(\mathcal{T}(i)))} \right) \quad (65)$$

Here, n_{ds} is the number of collocation points and θ_{hp} is the hyper-parameter for the softplus function. By employing a softplus function, the cost is designed to remain approximately constant when the vehicle is within the envelope and to increase linearly with respect to the distance away from the envelope when the vehicle is outside the envelope, similar to [56]. This is illustrated by the contour plot of the cost along a portion of the racetrack in Fig. 1-E5. As long as the vehicle remains within the racetrack, the cost is approximately zero. Conversely, when the vehicle veers off the track, the cost increases linearly as shown by the color change on the plot. This cost promotes the planned trajectories to remain within the envelope.

Specific Cost (J_{specific}) is a placeholder where costs related to specific scenarios can be included. The detailed expression of this type of cost is discussed in later sections when specific scenarios are studied.

3) *Solving Strategy*: In all scenarios tested in this work, the MPC framework plans for $T_p = 6.75$ s into the future with 25 collocation points, resulting in 24 time intervals denoted as T_1, T_2, \dots, T_{24} . These values were selected via a sweep over T_p and collocation grid densities to balance look-ahead, real-time feasibility, and numerical integration error. The time intervals are distributed as:

$$T_i = \begin{cases} 0.15 & \text{if } i \leq 15 \\ 0.5 & \text{otherwise} \end{cases}, \quad i = 1, \dots, 24 \quad (66)$$

with i being an index for the time interval. The shorter time interval at the beginning of the prediction horizon ensures that the propagation error does not become significant enough to jeopardize vehicle operation. A longer time horizon at the

end allows the MPC to look further into the future without incurring a high computational cost, facilitating earlier responses to upcoming turns. This approach offers an advantage over using the same time interval throughout the prediction horizon, as the latter either fails to provide adequate foresight or imposes additional computational burden on the solver to achieve the same propagation accuracy. The backward Euler integration scheme is employed between each time step due to its computational efficiency and unconditional stability. The nonlinear optimization problem is converted into a nonlinear program using the JuMP package in Julia and solved with Ipopt [57]. This implementation enables the algorithm to plan trajectories at a frequency of at least 10 Hz.

C. Spatial Envelope Design

The problem of designing the blocks that form a drivable spatial envelope poses a trade-off. On the one hand, to ensure fast computation of the LSE function, a small number of blocks is desired. On the other hand, to cover the drivable region as accurately as possible, a large number of small blocks may be needed.

This trade-off problem is simplified by breaking it into two subproblems: designing an optimal block for segments of the given road boundary recursively, and combining the designed blocks.

For designing a single block, possible solutions include optimization and machine learning. While optimization produces a high quality result at the cost of a longer solving time, machine learning methods are often faster, but the solution quality is not guaranteed. To address the drawbacks of both solutions, this work proposes a combined approach as illustrated in Fig. 2. The envelope is constructed by sequentially designing and connecting individual blocks. Initially, road boundary data are segmented into overlapping regions. For each region, reinforcement learning (RL) is used to provide a general estimate of block parameters L, W, C, θ , which refer to block half length, half width, left center, and block yaw angle. These estimates are then refined through an optimizer, maximizing the block's area inside the road segment. The start point of the next road segment is set to be the center of the designed block. The blocks are thus planned recursively until the end of the road boundary data.

1) Reinforcement Learning Based Policy Optimization:

This work proposes using Proximal Policy Optimization (PPO) for spatial envelope planning policy training [58].

Road Generator: The observation data for the RL agent is generated using a road generator. This generator first randomizes a curvature array κ_i , offset by another random value ϵ_i . Based on the curvatures, the centerline of the road C_i is formed as follows:

$$C_{0,x} = 0, \quad C_{i,x} = C_{i-1,x} + \cos \kappa_i \quad (67)$$

$$C_{0,y} = 0, \quad C_{i,y} = C_{i-1,y} + \sin \kappa_i \quad (68)$$

To avoid unrealistic variations in road width, five random widths between the given maximum and minimum road widths are assigned to random indices. The widths at other indices are interpolated linearly between these initial values. The width

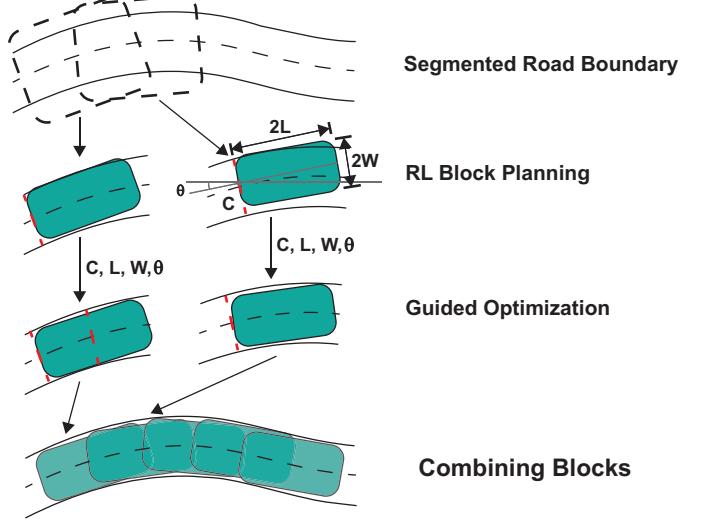


Fig. 2. Spatial envelope planning framework. Block left center, yaw angle, half length, and half width are passed from RL to optimizer as initialization parameters.

array is then smoothed as w_i using a Savitzky–Golay filter [59]. The left and right boundary points l_i, r_i are computed using the following equations:

$$l_{i,x} = C_{i,x} - w_i \sin \kappa_i \quad (69)$$

$$l_{i,y} = C_{i,y} + w_i \cos \kappa_i \quad (70)$$

$$r_{i,x} = C_{i,x} + w_i \sin \kappa_i \quad (71)$$

$$r_{i,y} = C_{i,y} - w_i \cos \kappa_i \quad (72)$$

Reward Function: The reward function is designed to encourage the designed block to maximize the area within the road boundaries while penalizing any portion that extends beyond the boundaries. The road is segmented into quadrilaterals Q_i , each formed by two points from the left boundary and two points from the right boundary. The reward function χ is expressed as:

$$\chi = (A_{\text{in}} - A_{\text{out}})2L \quad (73)$$

where A_{in} is the area of the designed block within the road boundaries, A_{out} is the area outside the boundaries, and L is the half length of the block. This formulation encourages longer blocks.

For simplicity, each block is approximated as a rectangle with the same left center, half width, half length, and yaw angle. This approximation is conservative, since the area of the 4th order norm block is fully included inside the approximated rectangle.

2) Policy Guided Block Optimization: The final block design is produced by solving an optimization problem to maximize the block size. The optimization formulation is as follows:

$$\begin{aligned} & \text{maximize} && LW \\ & \text{s.t.} && g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, m \\ & && h_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, p \end{aligned} \quad (74)$$

Block parameters L and W are re-scaled by a factor and then used to initialize the optimization. The factor is less than 1

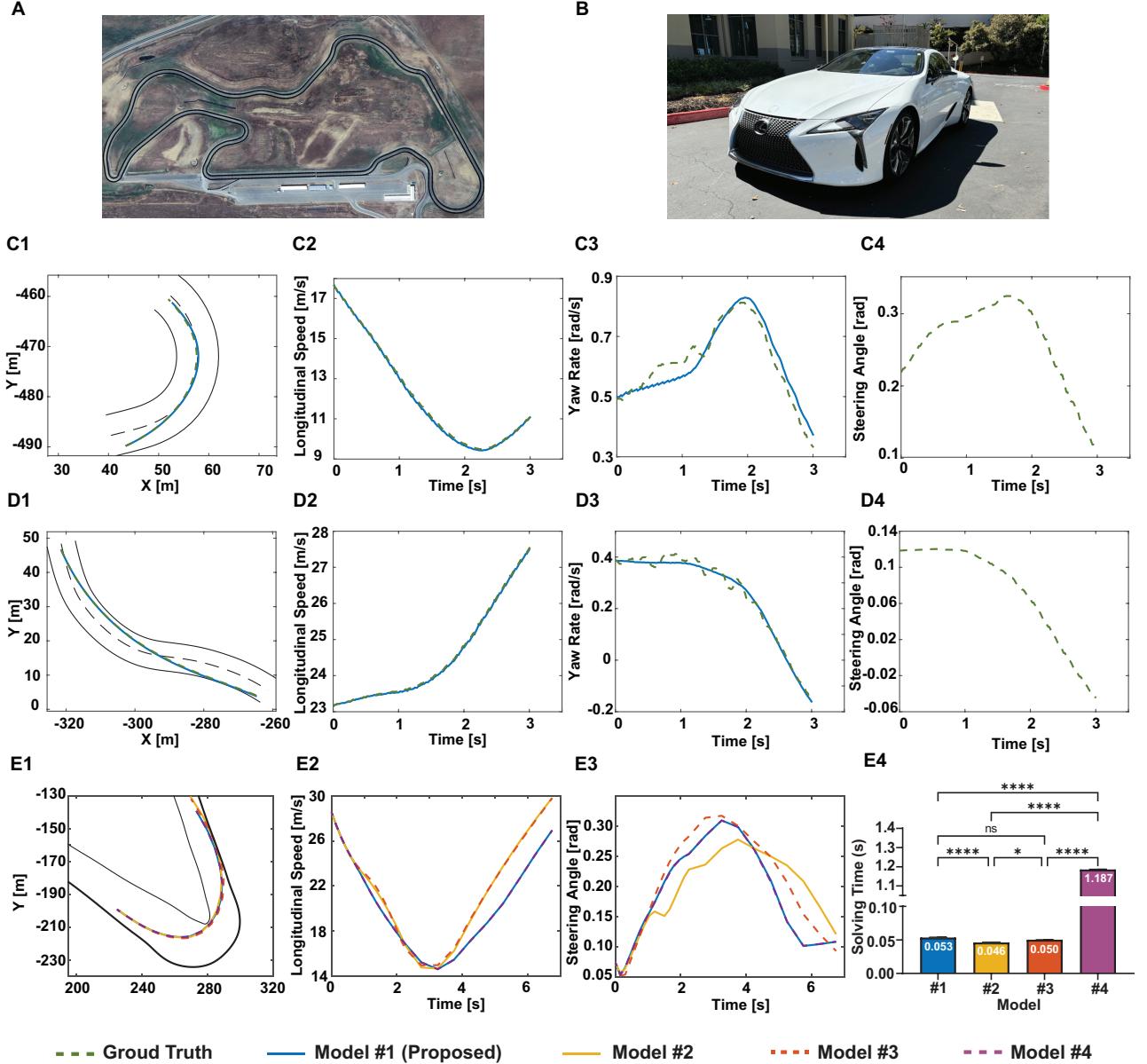


Fig. 3. The results in Sec. IV. (A) An aerial view of Thunderhill West from the satellite. (B) A snapshot of Toyota Research Institute 2019 LC 500. (C1 - D4) The plots comparing the proposed model with the real-vehicle data in open-loop simulation. The green dashed line represents the real-vehicle data and the blue solid line represents the proposed model. The subplots C1 - C4 show the results in a hairpin turn, while the subplots E1 - E4 show the results in a high-speed turn. (C1 & E1) The X-Y path comparisons in the hairpin and high-speed turn, respectively. (C2 & E2) The longitudinal speed comparisons in the hairpin and high-speed turn, respectively. (C3 & E3) The yaw rate comparisons in the hairpin and high-speed turn, respectively. (C4 & E4) The steering angle commands used by the real vehicle in the hairpin and high-speed turn, respectively. (E1 - E4) The comparisons of four vehicle models in one-time optimization. The blue solid line presents Model #1, which is referred to as the proposed model. The yellow solid, red dashed, and magenta dashed lines represent the benchmark Models #2, #3, and #4, respectively. (E1) The comparisons in optimal X-Y paths. (E2) The comparisons in optimal longitudinal speeds. (E3) The comparisons in optimal steering angles. (E4) Statistical comparisons in solving times.

to prevent infeasible initial conditions. Please refer to V-E for the values.

3) Iterative Block Design: The spatial envelope for driving is generated by iteratively designing the blocks. The starting position of the last block is set to be on the road cross-section line passing through a point S_k defined by the last block. S_k is L ahead of the last left center point in the length direction.

IV. VALIDATION OF THE VEHICLE MODEL

This section evaluates vehicle model fidelity through two studies. In the first study, the proposed vehicle model is com-

pared to real-vehicle data in open-loop simulations to validate its accuracy in prediction. The second study establishes the effectiveness of the proposed model as a prediction model for optimal control by comparing its performance against three benchmark models.

In the first study, a real vehicle is tested at Thunderhill West, an aerial view of which shown in Fig. 3-A. The vehicle is a Toyota Research Institute 2019 LC 500, shown in Fig. 3-B. Vehicle telemetry data were collected by manually driving along the track, and two regions were selected for open-loop

comparisons: one at a hairpin turn (Fig. 3-C1) and the second at a high-speed turn (Fig. 3-D1). In the open-loop simulation, the initial states of the proposed vehicle model (Model #1) are set identical to the initial states of the real vehicle, and the control inputs are linearly interpolated based on the time series from the recorded driver inputs. From Fig. 3-C1 to D4, the green dashed line represents the trajectory of the real vehicle, while the blue solid line represents that of the proposed model. Based on Fig. 3-C1, the difference in path between the real vehicle and the proposed model is capped at 45 cm. The longitudinal speeds are identical, as shown in Fig. 3-C2, since the control input of longitudinal jerk is derived from the speed data. Comparisons in yaw rate are shown in Fig. 3-C3. Noticeable differences in yaw rate might be due to the omission of chassis suspension in the proposed model. In this model, the roll and pitch angles are assumed to be zero. On the contrary, these angles can change in reality, especially in the hairpin where lateral acceleration is large. The steering angle of the real vehicle is shown in Fig. 3-C4. The second control input of steering rate for the proposed model is derived from this data. Similar comparisons in the high-speed turn are shown from Fig. 3-D1 to D4. In the high-speed turn, the model predicts the global locations accurately with a maximum error of 61 cm. Compared to the performance in the hairpin turn, the yaw rates are closer perhaps due to the suspension being less excited in the high-speed turn than in the hairpin turn. In the real vehicle data, the small oscillations observed at the beginning are at a frequency of about 2.5 Hz and likely correspond to the suspension [60].

In the second study, four models are used as prediction models for a one-time optimization. Model #1 is the proposed vehicle model. Model #2 considers longitudinal load transfer with the pure-slip tire model. Model #3 ignores longitudinal load transfer and the friction circle limit. Model #4 represents the model used in [61]. Models #2 and #3 are included as an ablation study. In [61], both longitudinal load transfer and the friction circle limit are considered. However, unlike Eq. (32) and Eq. (34), the expression for load transfer is not derived as linear constraints. Instead, Model #4 adheres to the friction circle constraint given by Eq. (29). Furthermore, the softplus function of Eq. (15) is not used to calculate the maximum lateral tire force.

The optimal trajectories of each model are shown in Fig. 3-E1 to E3. Model #1 and Model #4 exhibit identical optimal trajectories, as depicted in Fig. 3-E1 to E3, since they share the same equations of motion. Model #2 and Model #3 neglect the friction circle limit, resulting in higher mobility predictions compared to Model #1 and #4. On the other hand, these two models show similar paths, and their predicted paths are longer than those of Model #1 and Model #4 in Fig. 3-E1. This behavior is explained by the similar predicted longitudinal speeds of Model #2 and Model #3, which are larger than those of Model #1 and Model #4, as shown in Fig. 3-E2. In short, the vehicle is predicted to have higher speed during corners.

By including the load transfer and neglecting the friction circle limit, Model #2 can have larger maximum lateral tire forces on the front wheels than on the rear wheels during deceleration, which increases the likelihood of oversteer due to

load transfer. Meanwhile, Model #2 has greater maximum lateral tire forces on the rear wheels than on the front wheels during acceleration, which increases the likelihood of understeer. These behaviors explain why the steering angle of Model #2 is smaller than that of Model #1 and Model #4 before 3 seconds and larger than the other two models afterward, as shown in Fig. 3-E3.

Model #3 neglects both load transfer and the friction circle, so the imbalance in maximum lateral tire forces between the front and rear tires is less significant than in Model #2, making the vehicle closer to neutral-steer. As a result, the steering angle of Model #3 is closer to the proposed model than to Model #2, as shown in Fig. 3-E3.

Pairwise comparisons of solving times, conducted as post-hoc tests following Analysis of Variance (ANOVA), are displayed in Fig. 3-E4. In Fig. 3-E4, the asterisks over the plots demonstrate the statistical significance level between the planners: * indicates $p < 0.05$, ** indicates $p < 0.01$, *** indicates $p < 0.001$, and **** indicates $p < 0.0001$, where p is the p-value reported by ANOVA. Non-significant differences are denoted with ns. Each model is optimized 100 times. Although Model #1 and Model #4 follow the same assumptions, Model #4 requires more time — 1.14 s longer — due to different mathematical expressions. Compared to Model #1, Model #2 and Model #3 make some simplifications, reducing solving time by 7 ms and 3 ms, respectively, but at the cost of fidelity.

In summary, the proposed vehicle model aligns well with real-vehicle data in open-loop simulations. Neglecting the longitudinal load transfer and friction circle limit can lead to an overestimation of vehicle mobility, while the reduction in solving time due to these simplifications is minimal. The proposed model accounts for both load transfer and the friction circle limit, and it is more computationally efficient than the benchmark model in [61].

V. SIMULATION & EXPERIMENTAL RESULTS

A. Race Track: High Performance Racing

To test the proposed algorithm to its limits, racing applications are the optimal choice, as their primary goal is to safely exploit the vehicle's dynamic limits to achieve maximum speed on a racetrack. The boundaries on the racetrack naturally define the drivable region, thereby creating the envelope. Simulation studies are conducted in Sec. V-A1. To further validate the proposed algorithm, two physical experiments are conducted on two tracks as described in Sec. V-A2.

For these racing scenarios, a cost-to-go term is added to the cost function to meet the objectives of racing in addition to using the standard formulation described in Sec. III-B. This cost-to-go term is described next.

Cost to Go: To motivate the MPC to drive the vehicle as fast as possible, a cost-to-go term is designed to promote maximization of the vehicle's travel distance in each prediction horizon. Assuming the vehicle's starting position (x_0, y_0) corresponds to the closest centerline point (c_0) at the arc length s_0 on the track, the farthest centerline point used to calculate the cost-to-go term corresponds to $s_f = s_0 + u_{x_{\max}} T_p$. In Fig.

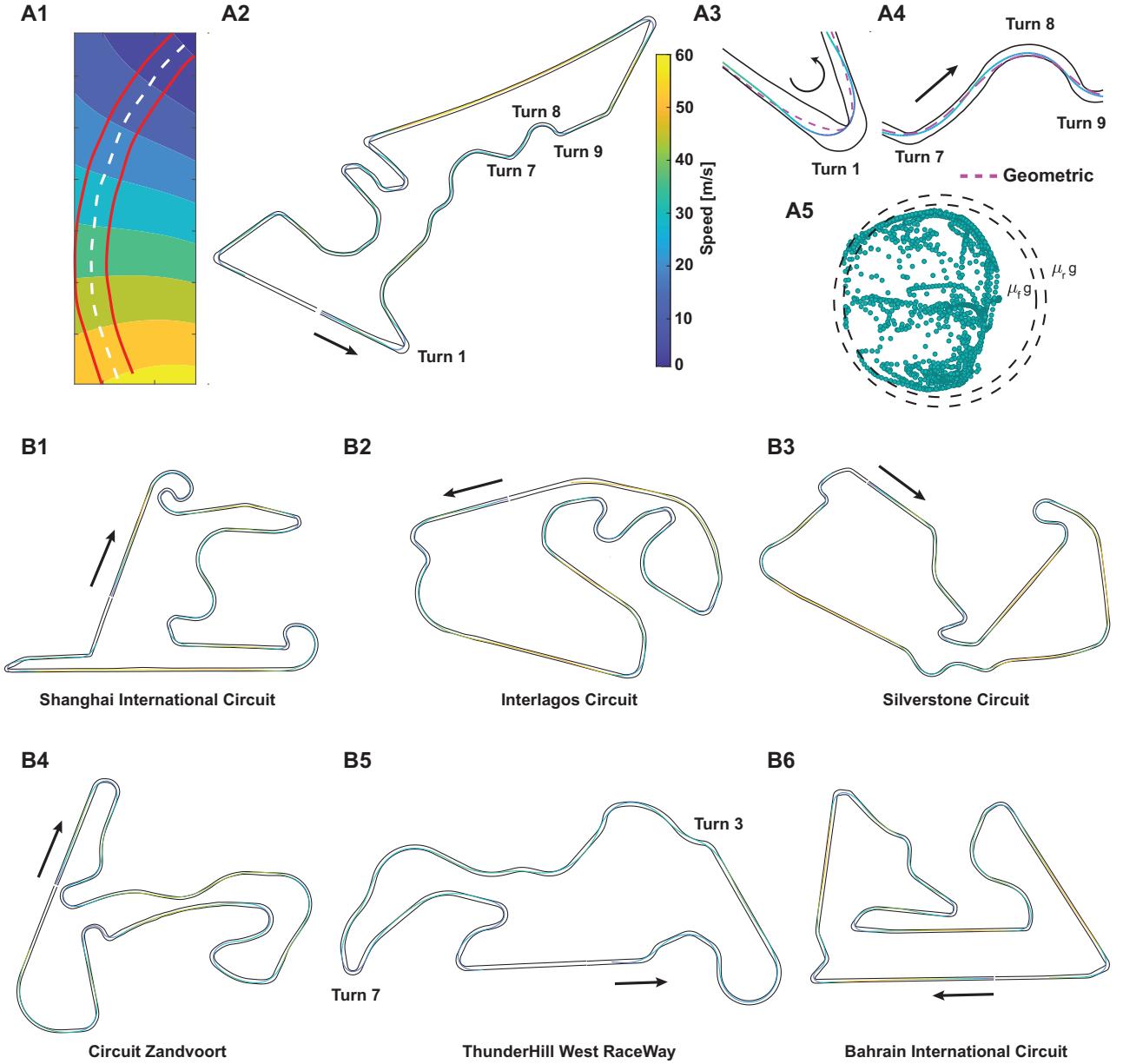


Fig. 4. The results in Sec. V-A1. **(A1)** The contour for the cost-to-go. **(A2)** The trajectory of the vehicle in the Circuit of The Americas (COTA). The color on the line represents the speed of the vehicle. **(A3-A4)** The detailed plot of select turns in COTA. The dashed magenta line represents the geometric line that has the least curvature. **(A5)** C.G. acceleration plot. **(B1-B6)** The trajectories of the vehicle in Shanghai International Circuit, Interlagos Circuit, Silverstone Circuit, Circuit Zandvoort, Thunderhill West RaceWay, Bahrain International Circuit. The line colors follow the same color bar in subplot A2.

4-A1, the bottom point corresponds to s_0 , and the top right point at the end of the lane corresponds to s_f . For each point on the centerline (corresponding to arc length s_p), a total of 15 points (including the centerline point itself) are uniformly sampled across the track width in the direction perpendicular to the centerline. All sampled points are labeled with the same arc length s_p .

For all points with arc length s_p , the cost-to-go terms are defined as $(s_f - s_p)$. In this setting, the cost is highest at the starting point and reduces to zero at the end. At a given arc length, the vehicle incurs the same cost regardless of its lateral position within the track bounds. To estimate the cost in optimization, a third-order polynomial regression is performed

to express the cost with respect to its location. Thus, the cost-to-go term is expressed as:

$$J_{\text{go}} = \sum_{i=0}^3 \sum_{j=0}^{3-i} w_{\text{go}}(i, j) x^i y^j \quad (75)$$

Fig. 4-A1 shows the contour plot of the fitted cost. This cost is included as J_{specific} in (62) as a specific cost for racing applications; i.e., $J_{\text{specific}} = J_{\text{go}}$.

1) *Simulation Study:* In Fig. 4-A2, the algorithm is put to test in Circuit of The Americas (COTA), a well-known race track that has been in Formula 1 Grand Prix since 2012. The trajectory of the vehicle is plotted and the color gradient on the trajectory represents the vehicle speed. The vehicle is capable

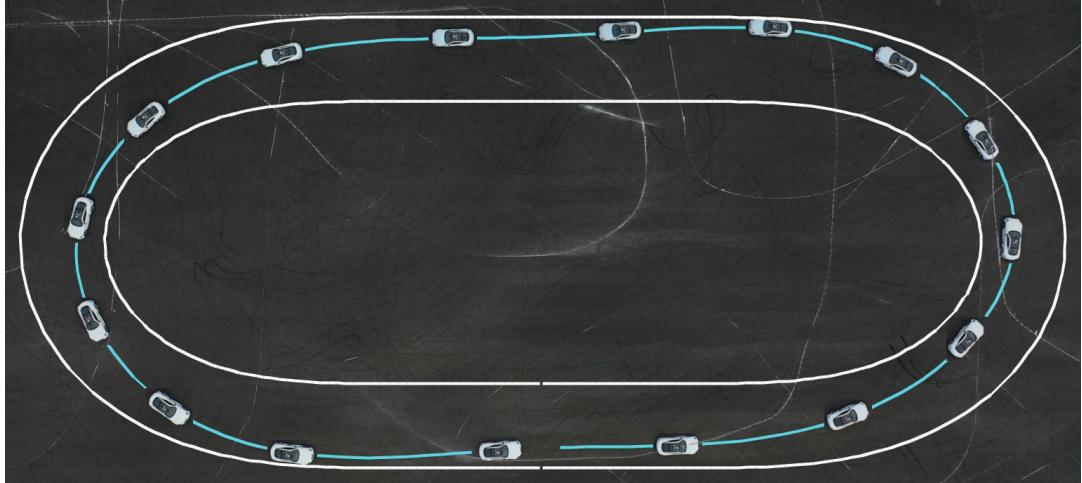
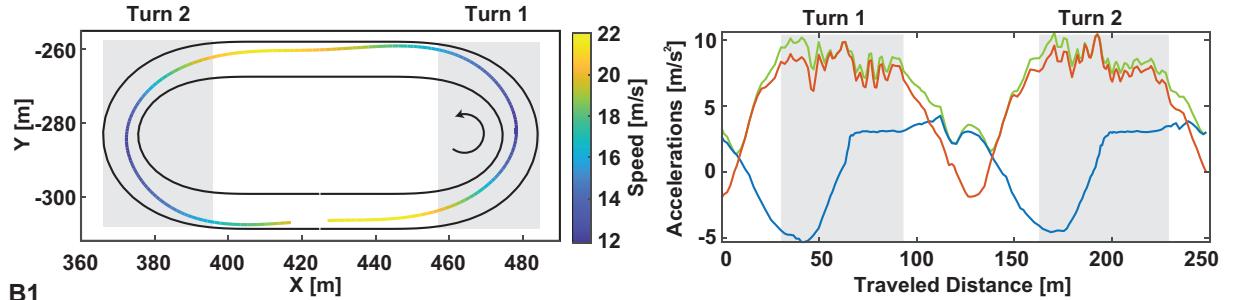
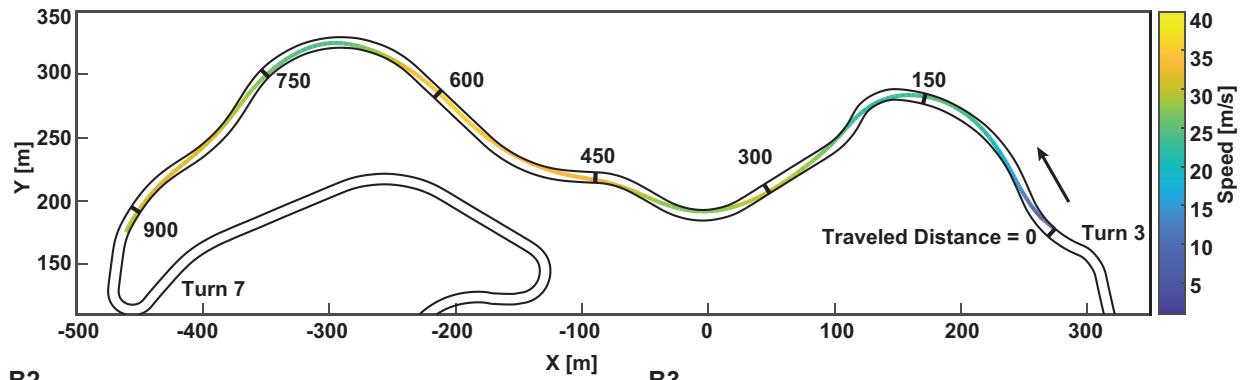
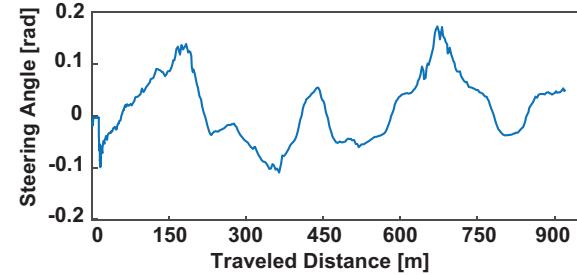
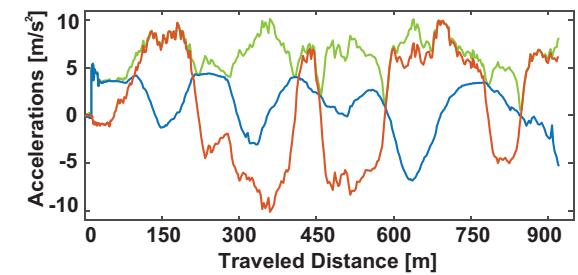
A1**A2****B1****B2****B3**

Fig. 5. The real vehicle testing results in Sec. V-A2. **(A1)** An aerial view showing the path of oval racing. The vehicle poses are superpositioned from the test video with an interval of 1 s. The inner and outer white oval lines represent the actual boundaries of the oval race track. The cyan solid line represents the path of C.G. point. **(A2)** The trajectory of the vehicle in the oval race track. The color on the line represents the speed of the vehicle. The rotation mark means that the vehicle is moving counter-clockwise. Two shaded gray rectangles represent the regions for Turn 1 and Turn 2. **(A3)** The plot of longitudinal, lateral, and total accelerations along the traveled distance in the oval race track. The regions of Turn 1 and 2 are emphasized with shaded gray rectangles. **(B1)** The trajectory of the vehicle in the Thunderhill West Raceway. The color on the line represents the speed of the vehicle. The inner and outer black solid lines represent the boundaries for the C.G. point of the vehicle. The traveled distances are labeled on the path with an interval of 150 m. **(B2)** The plot of steering angle along the traveled distance in Thunderhill West Raceway. **(B3)** The plot of longitudinal, lateral, and total accelerations along the traveled distance in Thunderhill West Raceway.

TABLE I
SIMULATED LAP TIME AND LENGTH OF SPATIAL ENVELOPE MPC

Name	Lap Time (s)	Length (m)
Bahrain	141.96	5200
COTA	153.36	5200
Catalunya	132.56	4500
Interlagos	114.66	4100
Shanghai	149.76	5250
SilverStone	144.66	5700
Thunderhill 2-mile	83.36	2450
Zandvoort	124.86	4100

of accelerating on the straight runs and decelerating before the turns so that it does not lose control.

Several key turns are magnified in Fig. 4-A3 and A4 to demonstrate the performance of the Spatial Envelope MPC. In Fig. 4-A3, a hairpin turn scenario is depicted, in which the vehicle must execute a sequence of dynamic maneuvers: first, applying heavy braking to decelerate, followed by aggressive steering to make the sharp turn, and finally accelerating upon exiting the turn. The trajectory plots in this figure indeed show this desired behavior. The geometric reference line, shown in purple, is derived from [62], where each segment is modeled using third-order polynomials, formulated to solve an optimization problem that minimizes curvature along the path. A notable deviation from the least-curvature geometric line is observed in the vehicle's trajectory, indicating a distinct driving behavior that emerges during the hairpin turn. Compared to the least-curvature geometric line, the vehicle exhibits a late braking pattern when approaching the turn and reaches the apex later. This strategy enables the vehicle to maintain higher exit speeds, resulting in faster acceleration after the turn compared to following the geometric line. This driving technique, commonly referred to as the late-apex strategy, is frequently employed by professional drivers to achieve faster lap times. In this setting, the algorithm is not assisted by prior knowledge but designs the trajectory on the fly, naturally embedding the late-apex strategy within the constraints and cost formulation.

In Fig. 4-A4, the vehicle does not follow the geometric line in relatively high-speed turns, either. After Turn 7, the vehicle intentionally deviates from the apex to slow down further and execute a more aggressive turn, eventually hitting the late-apex point. The rationale is that, after Turn 9, the track transitions into a long straight run where maintaining high initial speed is crucial for the vehicle. Adhering strictly to the geometric line would cause loss of speed during the turns, which is undesirable. The Spatial Envelope MPC predicts 6.75 s into the future for 6.75 s when making decisions to prevent settling on a locally optimal trajectory.

In Fig. 4-A5, the longitudinal and lateral accelerations are sampled throughout the entire lap and are compared to the front and rear friction force limits shown as dashed circles. The vehicle is safely pushed to the limits and is often operating near the limits in this application.

The same formulation is taken without any re-tuning and applied to other world-famous racetracks, as well; namely, Shanghai International Circuit (Fig. 4-B1), Interlagos Circuit (Fig. 4-B2), Silverstone Circuit (Fig. 4-B3), Circuit Zandvoort

(Fig. 4-B4), Thunderhill West Raceway (Fig. 4-B5), and Bahrain International Circuit (Fig. 4-B6). The algorithm can safely operate the vehicle across all six racetracks without modifications to the formulation; the only changing parameter is the boundary of the racetrack. The travel length and lap time for each comparison is reported in Table. I. These results highlight the method's general applicability and zero-shot capability.

2) *Experimental Validation:* To evaluate its real-world performance, the Spatial Envelope MPC algorithm is deployed on a prototype vehicle based on a 2019 LC 500 from Toyota Research Institute, as illustrated in Fig.3-B. The algorithm is tested on an oval race track and the Thunderhill West Raceway.

The oval race track was virtually defined within one of the skid pads in Thunderhill Raceway Park. Fig. 5-A1 illustrates the sequence of vehicle poses at 1-second intervals, demonstrating the algorithm's ability to maintain the vehicle within track boundaries. To further analyze the vehicle's mobility, the speed and acceleration profiles are presented in Fig.5-A2 and A3, respectively. In Fig. 5-A2, a maximum speed of 21.03 m/s is achieved near the entrance of Turns 1 and 2, while the minimum speed of 11.88 m/s occurs near the apex. Fig. 5-A3 shows that the maximum total acceleration of 10.57 m/s² is reached in Turns 1 and 2, constrained by the maximum available friction forces. During Turns 1 and 2, the total acceleration remains close to its maximum, indicating that the algorithm effectively drives the vehicle to its dynamic limits. During the two turns, the majority of the available friction forces are utilized as centrifugal forces to sustain curvilinear motion.

Fig. 5-B1 to B3 present the results of testing the Spatial Envelope MPC algorithm on the Thunderhill West Raceway, a significantly more complex and dynamic track compared to the virtual oval track. In Fig. 5-B1, the vehicle's trajectory is visualized along the track, with speed color gradients along the traveled path. The traveled distance is marked as vertical black lines on the racetrack at 150 m intervals. The vehicle starts from Turn 3 and navigates through the track, eventually reaching Turn 7 (marked in Fig. 4-B5), demonstrating the algorithm's ability to adapt to varying curvatures and track geometries in real time. This specific section is selected because the racetrack as a whole includes challenging terrain topologies, which could significantly impact the performance. because the current study has not yet considered the effects of terrain topologies on the racetrack. However, the section between Turn 3 and Turn 7, while providing significant challenges, is relatively flat, making it an ideal segment for testing. Due to heavy rain prior to the test date, the parameters for the front and rear friction coefficients are defined more conservatively as $(\mu_f, \mu_r) = (0.9, 0.95)$.

Notably, the speed profile along the trajectory shows that the algorithm effectively manages the vehicle's velocity depending on the curvature of the road. The vehicle reaches higher speeds on straight runs, with a maximum of 35.69 m/s (79.86 mph / 128.52 kph) recorded along a high-speed turn section between the 450 m and 600 m markers. The time-to-goal is 35.52 s for a traveled distance of 921 m.

Fig. 5-B2 illustrates the vehicle's steering trajectory along

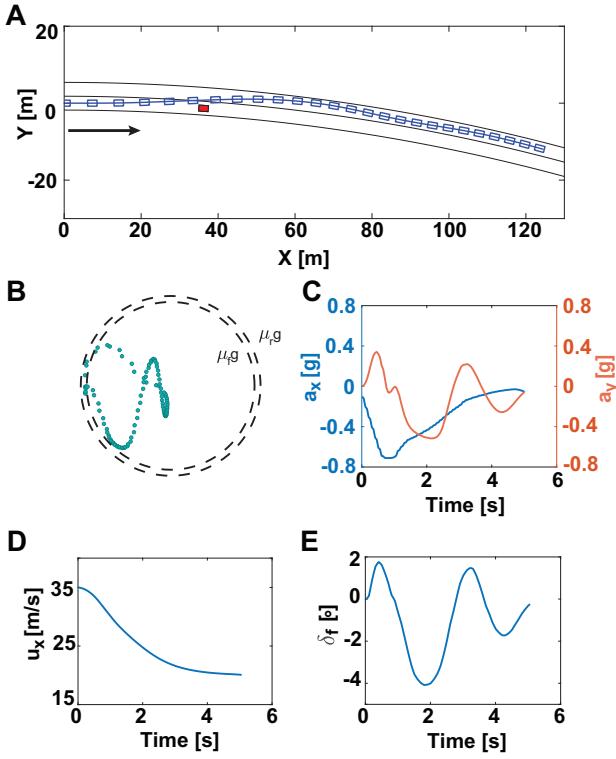


Fig. 6. The results in Sec. V-B. (A) The plot of the X-Y path. (B) The plot for longitudinal and lateral accelerations. (C) The time histories of longitudinal and lateral accelerations. The longitudinal acceleration is labeled on the left and the lateral acceleration is labeled on the right. In subplots B & C, the gravitational constant g is used as units for both longitudinal and lateral accelerations. (D) The time history for the longitudinal speed. (E) The time history for the steering angle.

the race track, where aggressive maneuvers are evident from the significant steering angles executed at high speeds. Fig. 5-B3 shows the vehicle's longitudinal, lateral, and total accelerations over the traveled distance. The total acceleration peaks at around 10 m/s^2 , indicating that the algorithm operates the vehicle near its dynamic limits, especially when negotiating tight corners. The lateral acceleration component dominates during cornering, confirming again that the majority of available friction forces are used to counteract centrifugal forces and maintain traction in the turns. Meanwhile, longitudinal acceleration spikes primarily occur during straight sections when the vehicle accelerates or decelerates before entering a turn. This careful balance between longitudinal and lateral forces showcases the algorithm's ability to push the vehicle to its performance limits while maintaining stability and control throughout the track.

B. On-Road: Emergency Collision Avoidance

A similar spatial envelope constraint concept can also be applied to emergency collision avoidance in on-road scenarios. This section demonstrates this in simulation. As shown in Fig. 6-A, the vehicle (depicted in blue) is traveling through a curved highway at an initial speed of 35 m/s (77 mph). At $t = 0$, a red vehicle that is 35 meters ahead suddenly appears, blocking the original lane and remaining stationary. This situation constitutes an emergency because the vehicle has only

one second to change lanes completely. The controller is also programmed to reduce the vehicle's speed to ensure it can be safely handed back to the human operator after the emergency maneuver. Compared to [5] with constant speed settings, this requirement brings additional challenges to coordinating the longitudinal and lateral tire forces. Intuitively, the benefits of these settings are twofold: 1) The additional degree of control freedom allows for better maneuvering capability, enabling the vehicle to perform more aggressive maneuvers and thereby enhancing safety in emergency scenarios. 2) It provides human drivers with a better takeover experience when the vehicle is stabilized in the other lane.

Speed Cost: To accomplish this, a simple cost function is added in J_{specific} to actively slow down the vehicle:

$$J_{\text{speed}} = \int_{t_0}^{t_f} (u_x - u_{\text{des}})^2 \quad (76)$$

Here the u_{des} is 20 m/s (44 mph) to ensure that the human driver can control the vehicle after the emergency maneuver.

The envelope used in this formulation is shown in Fig. 8-A3.

The resultant path of the collision imminent steering scenario is shown in Fig. 6-A. Vehicle positions are represented by a sequence of blue rectangles, each separated by a 0.2-second interval. The solid blue curve connecting each blue rectangle represents the history of the vehicle's C.G. point. The histories of C.G. accelerations are plotted in Fig. 6-B and C. In Fig. 6-B, the x-axis represents longitudinal acceleration, while the y-axis denotes lateral acceleration. Inner dashed circle indicates the acceleration limit of $\mu_f g$, while the outer dashed circle indicates the limit of $\mu_r g$. Several points in Fig. 6-B are located between the two dashed circles, indicating that the algorithm pushes the vehicle to its dynamic limits to avoid the obstacle. The time histories of two accelerations are shown in Fig. 6-C. To avoid the obstacle, the vehicle initiates a large steering angle, resulting in large lateral acceleration and the peak appears at around 0.5 s. An interesting trade-off between longitudinal and lateral acceleration is observed as the vehicle stabilizes in the second lane at approximately $t = 2 \text{ s}$. Given the speed cost, the vehicle should decelerate as rapidly as possible. However, the algorithm opts to apply less force for deceleration and instead actively redistributes forces in the lateral direction to assist with turning. The time history of longitudinal speed is shown in Fig. 6-D. The speed smoothly decreases from 35 m/s to 20 m/s which is the desired exit speed defined for this scenario. The steering angle trajectory is shown in Fig. 6-E. As mentioned before, the first peak of the steering angle appears at around 0.5 s, corresponding to the first peak of lateral acceleration. This maneuver leads to a large lateral speed and yaw rate to make sure that the vehicle can avoid the obstacle in time. Consequently, greater counter-steering is necessary to stabilize the vehicle and keep it within the lane boundary, which is shown as the first valley (at around 2 s) in Fig. 6-E.

In summary, the Spatial Envelope MPC is capable of performing collision avoidance in emergency scenarios on road. This method is capable of applying aggressive maneuvers to

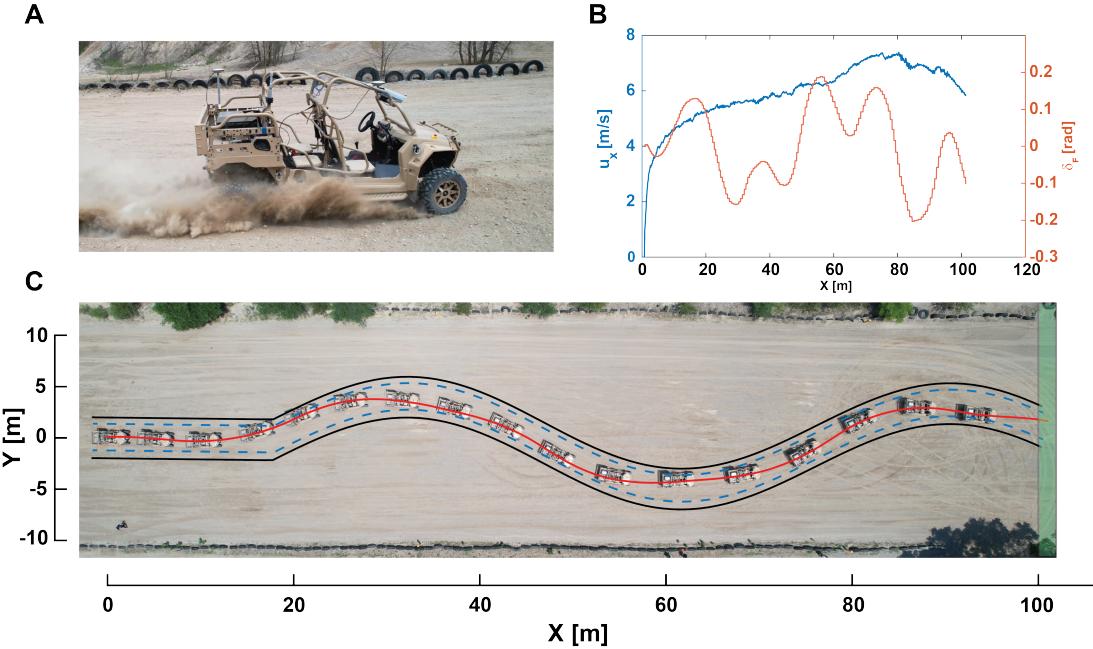


Fig. 7. The results in Sec. V-C. (A) A snapshot of the test vehicle - Polaris MRZR - driving in the off-road environment. (B) The plots of longitudinal speed and steering angle versus the X position. (C) An aerial view showing the X-Y path from one experiment. The vehicle poses are superpositioned from the aerial video. The upper and lower black solid lines represent the exact trail boundaries. The blue dashed lines represent the boundaries for the center of gravity (C.G.) of the vehicle. The red solid line represents the X-Y path of the vehicle's C.G. position. The small green area at the end denotes the ending region of the vehicle.

avoid obstacles while preserving vehicle stability and ensuring safety. Experimental results confirm that the vehicle adheres to dynamic limits, achieves the desired speed reduction, and executes safe lane-changing maneuvers, thereby demonstrating the efficacy of the proposed method.

C. Off-Road: Trail Navigation

To further demonstrate the real-world implementation capability and general usage of the algorithm, the Spatial Envelope MPC was installed on an off-road vehicle, as shown in Fig. 7-A. The vehicle running autonomously here is a gasoline-powered four-seater Polaris MRZR. As described in Sec. III-A, the vehicle model proposed in this study is primarily designed for on-road purposes. However, as indicated in [63], the on-road tire model fails to accurately capture tire-terrain interactions, leading to prediction errors when applied to off-road conditions. To address these challenges, this section utilizes the vehicle model introduced in [9], [63], which accounts for deformable terrain to ensure safe operation in off-road environments. As illustrated in Fig. 7-C, the scenario involves the vehicle navigating through a predefined trail without crossing the boundaries. This setup tests the algorithm's ability to handle real-world off-road navigation challenges. This trail scenario is commonly encountered in off-road navigation and rally racing, making it a relevant and practical test case for the algorithm. In this context, the two black lines represent the boundaries of the actual vehicle chassis, while the blue dashed lines indicate the derived boundaries for the vehicle's center of gravity (C.G.). As long as the vehicle's C.G. does not intersect the blue dashed lines, the vehicle is considered

to be safely within the trail boundaries. The small green area at the end of the trail represents the vehicle's target position. The vehicle is considered to have reached the goal once its x-coordinate exceeds 100 m. For this scenario, the original formulation in Sec. III-B is modified in the cost definition as follows.

Terminal Cost: The terminal cost defined with J_{specific} is written as:

$$J_{\text{specific}} = \frac{x_f - x_{\text{goal}}}{x_0 - x_{\text{goal}}} \quad (77)$$

where x_0 and x_f are the x position at the start and end point of the prediction horizon, respectively, and x_{goal} is the x position of the goal region. This cost motivates the MPC to plan trajectories leading the vehicle to the goal region as quickly as possible. No prescribed speed is used in this formulation; the algorithm simultaneously determines the speed and steering command for the vehicle.

Fig. 7-B shows the steering and speed trajectories of the vehicle and Fig. 7-C displays the path plot of the vehicle's C.G. along with superimposed bird's-eye view images of the vehicle from a stationary drone footage.

The results demonstrate that the proposed method effectively identifies trajectories that stay safely within the designated boundaries. Instead of following the center of the track, the algorithm autonomously chooses to travel closer to the boundaries, resulting in a reduced path curvature. This approach provides the advantages of increased speed and enhanced time-to-goal performance. This improved performance is also apparent in the steering plot. Although the trail is modeled as a sinusoidal wave, the steering trajectory does not

TABLE II
COMPUTATIONAL PERFORMANCE OF SPATIAL ENVELOPE MPC

Scenario	Solve Time Statistics (ms)	Scenario	Solve Time Statistics (ms)
Sim Racing (COTA)	37.5 ± 12.8	Sim Racing (Bahrain)	35.9 ± 23.1
Sim Racing (Shanghai)	36.6 ± 15.1	Exp Racing (Oval)	63.9 ± 20.4
Sim Racing (Interlagos)	35.8 ± 12.9	Exp Racing (Thunderhill)	71.4 ± 25.9
Sim Racing (Silverstone)	36.4 ± 13.4	Sim Emergency Collision Avoidance	30.5 ± 10.8
Sim Racing (Zandvoort)	38.2 ± 16.6	Exp Off-road Navigation	20.9 ± 9.8
Sim Racing (Thunderhill)	31.7 ± 13.6		

strictly adhere to this pattern. Variations in the steering angle enable the vehicle to achieve maximum speed along the trail.

In summary, the Spatial Envelope MPC demonstrates the capability to operate the vehicle in real-time within an off-road environment. Instead of adhering to a predefined trajectory, this method offers a high-performance solution that enhances off-road mobility within the trail.

D. Computational Performance of Spatial Envelope MPC

Table II summarizes the computational performance of each scenario in terms of mean and standard deviation of solve times. All simulated scenarios were computed on an Alienware R15 desktop with an Intel i9-14900K CPU clocking at 3.2 GHz, and 64 GB of memory, with optimizations performed in a single thread. The algorithm generated optimal control commands in approximately 35-40 ms, which is significantly below the 100 ms target threshold. For the physical tests in racing applications, a ThinkPad laptop with a Xeon(R) W-11855M CPU, clocking at 3.2 GHz, and 32 GB of memory was used. However, the computational performance was limited due to the laptop running on battery power. Therefore, the average solve time for each iteration increased to 60-70 ms, but still met the 100 ms threshold on average. For those instances where the optimization times reached 100 ms, the optimizations were terminated after 100 ms. If no solution was found within this time, the MPC switched to the new initial point and initiated a new calculation while still executing the optimal commands from the last successful solution. Consequently, the maximum optimization time was capped at 100 ms. Such overrun of the 100 ms limit happened in 64 OCPs out of a total of 1158 across the Oval and Thunderhill experiments. For the off-road experiment, a Dell Precision 5500 Workstation laptop with an Intel Core i7-10850H processor, clocked at 2.7 GHz, and 62.6 GB of memory was used to compute the vehicle's trajectories. These computational performances indicate that the algorithm can operate at 10 Hz for all scenarios on modern CPUs.

E. Spatial Envelope Planning

In Fig. 8, the planned spatial envelopes of three design schemes are compared: pure reinforcement learning, optimization, and reinforcement learning guided optimization. In RL

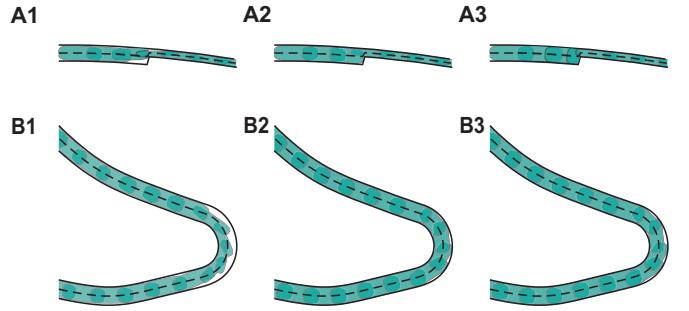


Fig. 8. (A) A segment of the designed spatial envelopes for the Collision Imminent Steering scenario. (B) Spatial envelopes designed by the three methods for Thunderhill West race track. (1) RL policy; (2) Optimization only; (3) RL guided optimization.

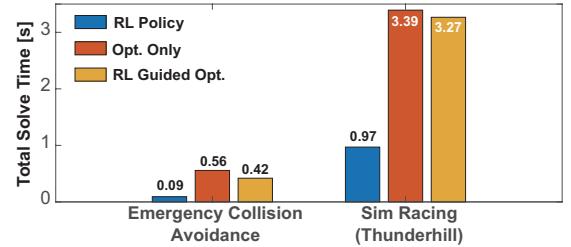


Fig. 9. Comparison of computational times of the three spatial envelope design schemes for Emergency Collision Avoidance and Sim Racing (Thunderhill) scenarios.

guided optimization, parameters L, W from RL are scaled by 0.8 to ensure proper initialization. The computation time is compared in Fig. 9.

The reinforcement learning approach exhibits the fastest computation, but the planned blocks are not necessarily optimal, and the constraints are not guaranteed to be satisfied. Although solely using optimization produces near-optimal results within the constraints, the algorithm is prone to be trapped in local minima, and has the highest computation time. Using reinforcement learning trained policy to guide optimization reduces the risk of local minima, enforces constraint satisfaction, and reduces computational time. This result demonstrates the reliability and effectiveness of the reinforcement learning guided optimization scheme for envelope design.

VI. CONCLUSION

This study introduces a spatial envelope based model predictive control framework designed to push a vehicle to its limits autonomously, safely, and in real time. A new 3 DoF single-track model is introduced. This model is specifically tailored for closed-loop optimization algorithms and captures efficiently the key vehicle dynamics in high-performance driving scenarios. A new mathematical formulation for spatial envelope constraints is developed to plan dynamically feasible and collision-free trajectories while maintaining scalability for complex scenarios. This formulation explicitly defines the entire drivable region, in contrast to the state of the art that constrains each collocation point to a specified track segment. As a result, the new envelope formulation removes the need for a restriction on longitudinal maneuvers. This novelty enables,

for the first time, one-level speed-varying trajectory planning and control within spatial envelope MPC and eliminates the need for a reference. This work also presents a technique to design the envelope effectively.

The vehicle dynamics model is first validated against real-world driving data, demonstrating improved accuracy and computational efficiency compared to benchmark models. Building on this foundation, the Spatial Envelope MPC is evaluated across a variety of high-performance driving scenarios, including both simulated and physical racing experiments, to establish its strong performance and zero-shot generalization capabilities. To further assess its robustness, an emergency collision avoidance scenario is designed, highlighting the controller's ability to coordinate longitudinal and lateral tire forces in critical on-road situations. The algorithm is then transferred to an off-road vehicle and tested on unstructured trails, illustrating its applicability beyond traditional on-road settings. In parallel, a method for effective spatial envelope design is introduced to support constraint formulation within the planning framework. Taken together, these contributions represent a significant step forward in enhancing the mobility and versatility of autonomous vehicles operating in high-performance environments in a reference-free setting.

Assessing the robustness of the algorithm is identified as the next step, as the current work assumes perfect knowledge of the environment. Although the proposed Spatial Envelope MPC is motivated by and evaluated in performance driving scenarios, the same methodology could be applied to other robotic systems such as drones, wheeled robots, and robotic arms. Applications to other problem domains are identified as important future research directions.

REFERENCES

- [1] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi, and R. Mangharam, "Autonomous vehicles on the edge: A survey on autonomous vehicle racing," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 458–488, 2022.
- [2] A. Wischnewski, M. Geisslinger, J. Betz, T. Betz, F. Fent, A. Heilmeier, L. Hermansdorfer, T. Herrmann, S. Huch, P. Karle *et al.*, "Indy autonomous challenge-autonomous race cars at the handling limits," in *International Munich Chassis Symposium*. Springer, 2022, pp. 163–182.
- [3] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 972–983, 2020.
- [4] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes, "Collision avoidance and stabilization for autonomous vehicles in emergency scenarios," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1204–1216, 2016.
- [5] J. Wurts, J. L. Stein, and T. Ersal, "Collision imminent steering at high speeds on curved roads using one-level nonlinear model predictive control," *IEEE Access*, vol. 9, pp. 39 292–39 302, 2021.
- [6] T. P. Weber and J. C. Gerdes, "Modeling and control for dynamic drifting trajectories," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 2, pp. 3731–3741, 2024.
- [7] J. Y. Goh and J. C. Gerdes, "Simultaneous stabilization and tracking of basic automobile drifting trajectories," in *IEEE Intelligent Vehicles Symposium*, 2016, pp. 597–602.
- [8] J. Y. Goh, M. Thompson, J. Dallas, and A. Balachandran, "Beyond the stable handling limits: nonlinear model predictive control for highly transient autonomous drifting," *Vehicle System Dynamics*, pp. 1–24, 2024.
- [9] J. Dallas, M. P. Cole, P. Jayakumar, and T. Ersal, "Terrain adaptive trajectory planning and tracking on deformable terrains," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 11 255–11 268, 2021.
- [10] C. Shen, S. Yu, B. I. Epureanu, and T. Ersal, "An efficient global trajectory planner for highly dynamical nonholonomic autonomous vehicles on 3D terrains," *IEEE Transactions on Robotics*, 2023.
- [11] S. Yu, C. Shen, and T. Ersal, "Nonlinear model predictive planning and control for high-speed autonomous vehicles on 3D terrains," *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 4th–417, 2021.
- [12] C. E. Beal and J. C. Gerdes, "Model predictive control for vehicle stabilization at the limits of handling," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1258–1269, 2012.
- [13] M. Brown, J. Funke, S. Erlien, and J. C. Gerdes, "Safe driving envelopes for path tracking in autonomous vehicles," *Control Engineering Practice*, vol. 61, pp. 307–316, 2017.
- [14] J. Wurts, J. L. Stein, and T. Ersal, "Collision imminent steering at high speed using nonlinear model predictive control," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8278–8289, 2020.
- [15] P. Stano, U. Montanaro, D. Tavernini, M. Tufo, G. Fiengo, L. Novella, and A. Sorniotti, "Model predictive path tracking control for automated road vehicles: A review," *Annual Reviews in Control*, vol. 55, pp. 194–236, 2023.
- [16] S. Erke, D. Bin, N. Yiming, Z. Qi, X. Liang, and Z. Dawei, "An improved A-Star based path planning algorithm for autonomous land vehicles," *International Journal of Advanced Robotic Systems*, vol. 17, no. 5, 2020.
- [17] D. Osmankovic, A. Tahirovic, and G. Magnani, "All terrain vehicle path planning based on D* Lite and MPC based planning paradigm in discrete space," in *IEEE International Conference on Advanced Intelligent Mechatronics*, 2017, pp. 334–339.
- [18] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," in *Proceedings of the AAAI Spring Symposium*, 2008.
- [19] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.
- [20] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Research Report 9811*, 1998.
- [21] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [22] D. J. Webb and J. Van Den Berg, "Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 5054–5061.
- [23] J. H. Jeon, R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, P. Tsiotras, and K. Iagnemma, "Optimal motion planning with the half-car dynamical model for autonomous high-speed driving," in *American Control Conference*, 2013, pp. 188–193.
- [24] 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 *International IEEE Conference on Intelligent Transportation Systems*, 2013, pp. 378–383.
- [25] E. Alcalá, V. Puig, J. Quevedo, and U. Rosolia, "Autonomous racing using linear parameter varying-model predictive control (LPV-MPC)," *Control Engineering Practice*, vol. 95, p. 104270, 2020.
- [26] J. Y. Goh, T. Goel, and J. Christian Gerdes, "Toward automated vehicle control beyond the stability limits: drifting along a general path," *Journal of Dynamic Systems, Measurement, and Control*, vol. 142, no. 2, p. 021004, 2020.
- [27] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "Combined speed and steering control in high-speed autonomous ground vehicles for obstacle avoidance using model predictive control," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 8746–8763, 2017.
- [28] C. Hu, L. Zhao, L. Cao, P. Tjan, and N. Wang, "Steering control based on model predictive control for obstacle avoidance of unmanned ground vehicle," *Measurement and Control*, vol. 53, no. 3-4, pp. 501–518, 2020.
- [29] H. Lim, Y. Kang, C. Kim, J. Kim, and B.-J. You, "Nonlinear model predictive controller design with obstacle avoidance for a mobile robot," in *IEEE International Conference on Mechatronic and Embedded Systems and Applications*, 2008, pp. 494–499.
- [30] J. Fan, N. Murgovski, J. Liang, and A. Elawad, "Exact obstacle avoidance for autonomous vehicles in polygonal domains," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2024.
- [31] S. Li, Z. Li, Z. Yu, B. Zhang, and N. Zhang, "Dynamic trajectory planning and tracking for autonomous vehicle with obstacle avoidance based on model predictive control," *IEEE Access*, vol. 7, pp. 132 074–132 086, 2019.

- [32] H. Febbo, J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "Moving obstacle avoidance for large, high-speed autonomous ground vehicles," in *American Control Conference*, 2017, pp. 5568–5573.
- [33] B. Gutjahr, L. Gröll, and M. Werling, "Lateral vehicle trajectory optimization using constrained linear time-varying MPC," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1586–1595, 2016.
- [34] S. Xu and H. Peng, "Design, analysis, and experiments of preview path tracking control for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 48–58, 2019.
- [35] J. Hu, S. Xiong, J. Zha, and C. Fu, "Lane detection and trajectory tracking control of autonomous vehicle based on model predictive control," *International Journal of Automotive Technology*, vol. 21, pp. 285–295, 2020.
- [36] M. Brown and J. C. Gerdes, "Coordinating tire forces to avoid obstacles using nonlinear model predictive control," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 1, pp. 21–31, 2019.
- [37] M. Thompson, J. Dallas, J. Y. Goh, and A. Balachandran, "Adaptive nonlinear model predictive control: maximizing tire force and obstacle avoidance in autonomous vehicles," *IEEE Transactions on Field Robotics*, 2024.
- [38] J. Lee, B. Kim, J. Seo, K. Yi, J. Yoon, and B. Ko, "Automated driving control in safe driving envelope based on probabilistic prediction of surrounding vehicle behaviors," *International Journal of Passenger Cars-Electronic and Electrical Systems*, vol. 8, no. 2015-01-0314, pp. 207–218, 2015.
- [39] B. Jiang, X. Li, Y. Zeng, and D. Liu, "A novel maneuver-based driving envelope generation approach for driving safety assessment," *Applied Sciences*, vol. 11, no. 10, p. 4702, 2021.
- [40] S. Yu, C. Shen, and T. Ersal, "Autonomous driving using linear model predictive control with a koopman operator based bilinear vehicle model," *IFAC-PapersOnLine*, vol. 55, no. 24, pp. 254–259, 2022.
- [41] J. Bernhard, P. Hart, A. Sahu, C. Schöller, and M. G. Cancimane, "Risk-based safety envelopes for autonomous vehicles under perception uncertainty," in *IEEE Intelligent Vehicles Symposium*, 2022, pp. 104–111.
- [42] K. Daniels, V. Milenkovic, and D. Roth, "Finding the largest area axis-parallel rectangle in a polygon," *Computational Geometry*, vol. 7, no. 1, pp. 125–148, Jan. 1997.
- [43] R. Molano, P. G. Rodríguez, A. Caro, and M. L. Durán, "Finding the largest area rectangle of arbitrary orientation in a closed contour," *Applied Mathematics and Computation*, vol. 218, no. 19, pp. 9866–9874, Jun. 2012.
- [44] H. Liu, J. Zhou, X. Wu, and P. Yuan, "Optimization algorithm for rectangle packing problem based on varied-factor genetic algorithm and lowest front-line strategy," in *IEEE Congress on Evolutionary Computation*, 2014, pp. 352–357.
- [45] R. E. Korf, M. D. Moffitt, and M. E. Pollack, "Optimal rectangle packing," *Annals of Operations Research*, vol. 179, pp. 261–295, 2010.
- [46] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, and U. Müller, "Explaining how a deep neural network trained with end-to-end learning steers a car," *arXiv:1704.07911*, 2017.
- [47] T. Weiss and M. Behl, "Deepracing: A framework for autonomous racing," in *Design, Automation & Test in Europe Conference & Exhibition*, 2020, pp. 1163–1168.
- [48] J. Niu, Y. Hu, B. Jin, Y. Han, and X. Li, "Two-stage safe reinforcement learning for high-speed autonomous racing," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2022, pp. 3934–3941.
- [49] A. Remonda, S. Krebs, E. Veas, G. Luzhnica, and R. Kern, "Formula RL: Deep reinforcement learning for autonomous racing using telemetry data," *arXiv:2104.11106*, 2021.
- [50] Y. Song, H. Lin, E. Kaufmann, P. Dürr, and D. Scaramuzza, "Autonomous overtaking in gran turismo sport using curriculum reinforcement learning," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 9403–9409.
- [51] V. A. Laurense, *Integrated motion planning and control for automated vehicles up to the limits of handling*. Stanford University, 2019.
- [52] C. Nuthong and T. Charoenpong, "Lane detection using smoothing spline," in *International Congress on Image and Signal Processing*, vol. 2, 2010, pp. 989–993.
- [53] G. Kreisselmeier and R. Steinhauser, "Systematic control design by optimizing a vector performance index," in *Computer Aided Design of Control Systems*. Elsevier, 1980, pp. 113–117.
- [54] F. Nielsen and K. Sun, "Guaranteed bounds on information-theoretic measures of univariate mixtures using piecewise log-sum-exp inequalities," *Entropy*, vol. 18, no. 12, p. 442, 2016.
- [55] T. G. Molnar and A. D. Ames, "Composing control barrier functions for complex safety specifications," *IEEE Control Systems Letters*, vol. 7, pp. 3615–3620, 2023.
- [56] I. Karino, J. Dallas, and Y. M. J. Goh, "Shared control for giving ordinary drivers expert level drifting skills," in *2023 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2023, pp. 1461–1467.
- [57] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.
- [58] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.
- [59] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.
- [60] N. Rajeev, "Pratheeksudi natural frequency, ride frequency and their influence in suspension system design," *J. Eng. Res. Appl.*, vol. 9, pp. 60–64, 2019.
- [61] J. Talbot, M. Brown, and J. C. Gerdes, "Shared control up to the limits of vehicle handling," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 2977–2987, 2023.
- [62] F. Braghin, F. Cheli, S. Melzi, and E. Sabbioni, "Race driver model," *Computers & Structures*, vol. 86, no. 13–14, pp. 1503–1516, 2008.
- [63] S. Yu, C. Shen, J. Dallas, B. I. Epureanu, P. Jayakumar, and T. Ersal, "A real-time terrain-adaptive local trajectory planner for high-speed autonomous off-road navigation on deformable terrains," *IEEE Transactions on Intelligent Transportation Systems*, 2024.



Siyuan Yu received his B.S.E in Electrical and Computer Engineering from Shanghai Jiao Tong University, China, in 2020 and the B.S.E and M.S.E in Mechanical Engineering from University of Michigan, Ann Arbor in 2020 and 2022, respectively. He is currently pursuing the Ph.D. degree in Mechanical Engineering at the University of Michigan, Ann Arbor. His research interests include modeling, system identification, motion planning and control, with respect to vehicle systems.



Congkai Shen earned his Bachelor of Science in Electrical and Computer Engineering from Shanghai Jiao Tong University, China, in 2020. Subsequently, he completed his Bachelor of Science and Master of Science in Mechanical Engineering at the University of Michigan, Ann Arbor, in 2020 and 2022, respectively. Currently, he is pursuing the Ph.D. degree in Mechanical Engineering at the University of Michigan, Ann Arbor. His research focuses on modeling, system identification, motion planning, and control, in the context of vehicle systems.



Yufei Xi is currently pursuing B.S.E in Mechanical Engineering from Shanghai Jiao Tong University, and B.S.E in Computer Engineering from University of Michigan, Ann Arbor. His research interests include modeling, motion planning and control of dynamic systems.



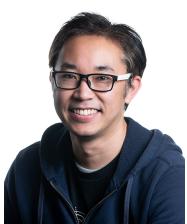
James Dallas received the B.S. degree from the Pennsylvania State University, University Park, PA USA, and the M.S. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI USA, in 2017, 2018, and 2021, respectively, all in Mechanical Engineering. He is currently a Research Scientist with the Toyota Research Institute. His research interests include modeling, system identification, system dynamics and control, shared control, and adaptive and robust optimal control, with applications to vehicle systems.



Michael Thompson received a B.S. degree in aerospace engineering from the University of Notre Dame and a M.S. degree in aerospace engineering from Stanford University. He is currently a member of the Extreme Performance Intelligent Control group at Toyota Research Institute. His research interests include vehicle dynamics modeling and control algorithms for high performance autonomous vehicles



John Subosits received his B.S.E. degree in mechanical and aerospace engineering from Princeton University and M.S. and Ph.D. degrees in mechanical engineering from Stanford University. Currently, he leads the Extreme Performance Intelligent Control group at Toyota Research Institute (TRI). His research interests include algorithms for vehicle control that match the performance, robustness, and adaptability of the best human (racing) drivers.



Hiroshi Yasuda received a B.E. from the Tokyo University of Science, Japan, in 2003, and M.E. and Ph.D. in engineering from the Tokyo Institute of Technology, Japan, in 2005 and 2008. He is currently a Staff Researcher and HMI tech lead at the Toyota Research Institute in Los Altos, California. His research interests include HMIs for advanced safety systems, and augmented/mixed reality for vehicles.



Tulga Ersal received the B.S.E. degree from the Istanbul Technical University, Istanbul, Turkey, in 2001, and the M.S. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI USA, in 2003 and 2007, respectively, all in mechanical engineering. He is currently an Associate Professor in the Department of Mechanical Engineering, University of Michigan, Ann Arbor. His research interests include modeling, simulation, and control of dynamic systems, with applications to vehicle and energy systems.