2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Spatially-Aware Adaptive Trajectory Optimization with Controller-Guided Feedback for Autonomous Racing

Alexander Wachter[1], Alexander Willert[1], Marc-Philip Ecker[1,2] and Christian Hartl-Nesic[1]

***Abstract*— We present a closed-loop framework for autonomous raceline optimization that combines NURBS-based trajectory representation, CMA-ES global trajectory optimization, and controller-guided spatial feedback. Instead of treating tracking errors as transient disturbances, our method exploits them as informative signals of local track characteristics via a Kalman-inspired spatial update. This enables the construction of an adaptive, acceleration-based constraint map that iteratively refines trajectories toward near-optimal performance under spatially varying track and vehicle behavior. In simulation, our approach achieves a 17.38 % lap time reduction compared to a controller parametrized with maximum static acceleration. On real hardware, tested with different tire compounds ranging from low to high friction, we obtain a 7.60 % lap time improvement without explicitly parametrizing friction. This demonstrates robustness to changing grip conditions in real-world scenarios.**

I. Introduction

High-performance autonomous racing poses a unique optimization challenge: Vehicles repeatedly traverse identical track layouts and this structured repetition can be exploited for systematic trajectory improvement. Unlike general autonomous driving on public roads where conditions vary unpredictably, racing circuits exhibit consistent spatial patterns that can be systematically learned and leveraged. Current methods employ a decoupled two-stage paradigm: offline trajectory generation followed by online tracking control [1], [2]. Hence, conventional approaches do not exploit the repetitive nature. Recent advances focus predominantly on improving controllers to handle challenging trajectories using learning-based model-predictive control (MPC) [3], [4], adaptive parameter tuning [5], and vehicle model adaptation [6], [7].

These controller-centric approaches interpret tracking errors primarily as control deficiencies and mitigate them using sophisticated strategies. As a result, they focus on local adaptations and treat each lap independently, missing the opportunity to exploit the repetitive structure of racing circuits. Furthermore, they often require long prediction horizons to handle local disturbances as effectively as a global trajectory planner.

When a controller consistently deviates from the reference trajectory in certain track regions, it signals that the planned trajectory is the limiting factor, rather than the controller trying to track it. Instead of forcing the controller to accommodate difficult trajectories, we propose to systematically modify the trajectory globally based on feedback from controller performance. In this way, we also shift computation from the online controller to the offline trajectory design by incorporating feedback of the controller performance into trajectory planning. This not only reduces the online computational load but also enables the controller to operate with longer horizons or additional optimization objectives.

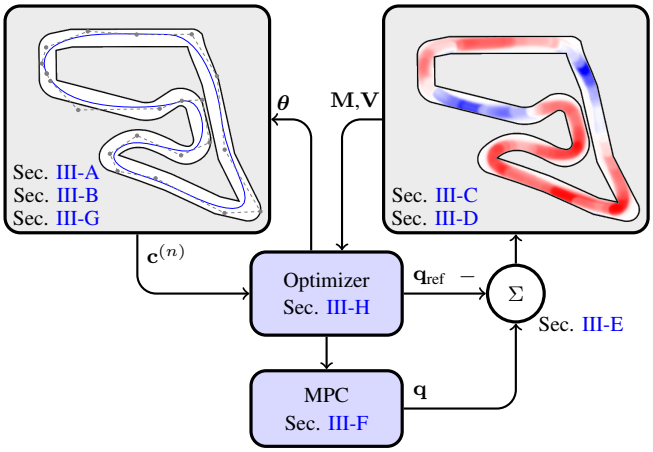

Fig. 1. Closed-loop raceline optimization framework using NURBS trajectories, CMA-ES optimization, and MPC tracking feedback.

In this work, we present a closed-loop raceline optimization framework that leverages controller tracking performance to improve the trajectory. Our approach constructs a spatial constraint map from the execution feedback, identifying regions where trajectory adjustments reduce the control error and lap time simultaneously, see Fig. 1. By combining Non-Uniform Rational B-Spline (NURBS)-based trajectory representation with a Covariance Matrix Adaptation Evolution Strategy (CMA-ES) optimization, we enable continuous adaptation of both trajectory shape and timing across multiple laps, while enforcing vehicle dynamics and achieving a smooth representation of our trajectory.

Our main contributions in this work are:

- **Trajectory optimization based on execution feedback**: We adapt trajectory shape and timing based on controller tracking performance, moving from local controller-centric adaptation to global trajectory-centric optimization.
- **Spatial constraint learning**: Leveraging tracking errors to learn a spatial performance map guides trajectory

[1] Alexander Wachter, Alexander Willert, Marc-Philip Ecker and Christian Hartl-Nesic are with the Automation & Control Institute (ACIN), TU Wien, Vienna, Austria, {wachter,willert, ecker,hartl}@acin.tuwien.ac.at

[2]Marc-Philip Ecker is with the AIT, Austrian Institute of Technology GmbH, Vienna, Austria, ecker@ait.ac.at

* This project is funded by the FFG (FO999896399). www.ffg.at

optimization without explicit environmental sensing.

- **Closed-loop refinement architecture**: The integration of a NURBS-based representation with CMA-ES optimization and adaptive spatial constraints demonstrates consistent lap time improvements via trajectory adaptation.
- **Open-source implementation**: We provide an open-source implementation with integrated GUI for ease of use, available at github.com/TU-Wien-ACIN-CDS/SpatialTrajoptRacing.

II. Related Work

The field of motion planning for autonomous vehicle has progressed from basic path planning approaches to advanced, spatially-aware trajectory optimization methods, with notable developments specialized in autonomous racing.

A. Motion Planning and Trajectory Optimization

The core challenge in autonomous racing is centered on the Minimum Lap Time Problem (MLTP), formulated as an optimal control problem [8], [9]. Early geometric methods focused on curvature minimization [10], while current approaches employ two-stage pipelines combining global offline optimization with local online planning [11]. Point-mass models with acceleration constraints [12], [13] offer computational speed, while single-track models with tire dynamics [14], [15] provide enhanced realism.

B. Spatially-Aware Dynamics and Adaptive Methods

Most approaches assume spatially invariant vehicle dynamics, ignoring location-specific surface variability. The GripMap framework addresses this potential by spatially resolving dynamic constraints in Frenet coordinates, achieving $5.2\,\%$ lap time improvement [16]. Christ et al. [17] incorporate locally varying tire-road friction via friction maps, which is computationally demanding for online applications. Model-adaptive approaches address time-varying friction without spatial resolution. Nagy et al. [7] employ *ensemble Gaussian processes* for multi-surface adaptation, while Kalaria et al. [5] use *extreme learning machines* for online tire-slip approximation. However, these focus on temporal changes rather than position-dependent friction models leveraging repeated track traversal.

C. Learning-Based Approaches and Optimization Methods

Population-based optimization methods like CMA-ES [18] prove effective for non-convex trajectory optimization where gradient information is uncertain. Most of the works that solve the MLTP problem produce static trajectories without iteratively learning from execution. Recent learning-based methods have introduced significant improvements by leveraging execution feedback. *Iterative learning control* (ILC) exploits repetitive lap-by-lap driving to improve feedforward inputs [19]. *Learning MPC* (LMPC) extends this concept by using past lap data to refine trajectories [20], [4]. Deep-kernel Gaussian processes [6] provide enhanced vehicle dynamics modeling. Sampling-based planning [2] enables dynamic adaptation for three-dimensional race tracks.

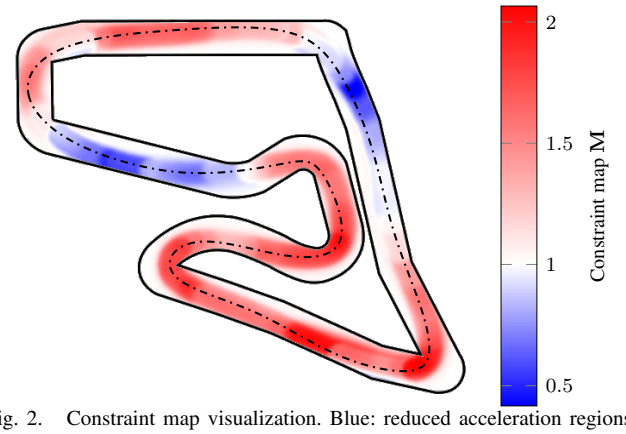

Fig. 2. Constraint map visualization. Blue: reduced acceleration regions; Red: increased acceleration regions.

III. Method

We present a closed-loop learning framework for trajectory optimization in autonomous racing that integrates smooth trajectory representation, adaptive constraint modeling, and real-time control. In contrast to conventional two-step offline raceline optimization, the proposed framework formulates trajectory design as an iterative process that continuously adapts via execution feedback.

The framework comprises four main components, see Fig. 1. A **NURBS-based trajectory representation** (Section III-A) restricts the search space to feasible trajectories, reducing optimization complexity. An **adaptive constraint map** (Section III-C) models spatially-varying acceleration limit, which are refined via controller feedback. **Global trajectory optimization** (Section III-H), implemented with CMA-ES, computes time-optimal trajectories based on the representation and constraint map. During execution, a **tracking controller** (Section III-F) provides feedback in terms of tracking errors. **Blame region computation** (Section III-D) localizes trajectory segments associated with the tracking errors, while **error feedback** (Section III-E) updates the constraint map accordingly.

The design of our approach tightly couples spatial constraint adaptation and trajectory optimization, i.e., updated acceleration limits are immediately reflected in the trajectory lap time. The result is a continuous learning cycle in which execution feedback directly informs subsequent trajectory generation, leading to progressive performance improvements.

A. NURBS-based Trajectory Representation

The foundation of our adaptive framework lies in representing the racing trajectories by Non-Uniform Rational B-Splines (NURBS), which establish a mapping from a normalized domain $u \in [0, 1]$ to the raceline trajectory. This spline representation enables analytical computation of derivatives to arbitrary order, only depending on the degree of the curve.

The spatial NURBS curve representation is mapped to a temporal trajectory with the parameterization

$$u(t) = \frac{t}{T}, \quad 0 \leq t \leq T \ , \tag{1}$$

where T denotes the lap time. This parameterization transforms the geometric path $\mathbf{c}(u)$ defined in the normalized domain $u \in [0,1]$ into a physical trajectory $\mathbf{q}(t) = \mathbf{c}(u(t))$ executed by the vehicle. Consequently, the vehicle's position at any time t is given by evaluating the NURBS curve at the corresponding parameter value $u = t/T$.

For a NURBS curve of degree p defined over $m+1$ knots $\mathbf{u} = [u_0, \ldots, u_m]^{\mathrm{T}}$ with $n+1$ control points $\mathbf{p}_i \in \mathbb{R}^2$, $i = 0, \ldots, n$, where $n = m - p - 1$, the resulting trajectory has C^{p-1} continuity and is mathematically described by

$$\mathbf{c}(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i \mathbf{p}_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \ , \quad 0 \le u \le 1 \ , \tag{2}$$

where $w_i \in \mathbb{R}^+$ represent positive weighting coefficients and $N_{i,p}(u)$ denote the B-spline basis functions of degree p evaluated over the knot vector $\mathbf{u}$ given by

$$\mathbf{u} = [\underbrace{0, \ldots, 0}_{p+1}, u_{p+1}, \ldots, u_{m-p-1}, \underbrace{1, \ldots, 1}_{p+1}]^{\mathrm{T}} \ . \tag{3}$$

Note that the knot multiplicities of $p+1$ at the boundaries allow for an analytical specification of the derivative constraints as well as the positions. In our racing context, we adopt cubic NURBS ($p = 3$) to guarantee C^2 continuity, thereby ensuring smooth curvature transitions essential for feasibility in vehicle dynamics and control stability.

Physical derivatives of the trajectory $\mathbf{q}(t) = \mathbf{c}(t/T)$ follow from application of the chain rule, yielding

$$\mathbf{q}^{(k)}(t) = \frac{1}{T^k} \mathbf{c}^{(k)} \left(\frac{t}{T} \right) \ , \quad k \ge 1 \ , \tag{4}$$

where $\mathbf{c}^{(k)} = \frac{\partial^k \mathbf{c}}{\partial u^k}$ represents the k-th parametric derivative. This formulation not only provides computationally efficient access to velocity, acceleration, jerk, and higher-order kinematic quantities, but also allow for a dynamic limit satisfaction by utilizing the lap time T.

B. Time-Optimal Parameterization with Spatial Adaptation

Using the NURBS trajectory introduced in Section III-A, the velocity and acceleration along the path are expressed using (4) as functions of the lap time T. This formulation establishes the minimal feasible lap time while enforcing the vehicle's dynamic limits.

The car velocity *along the path* is given by

$$v(t) = \frac{1}{T} \left\| \mathbf{c}^{(1)}(u) \right\|_2 \ , \tag{5}$$

showing the scaling with $\frac{1}{T}$. By setting $v(t) = v_{\max}$ and solving for T, we obtain the minimum feasible lap time imposed by the velocity limit.

Similarly, the decomposition into longitudinal ($\parallel$) and lateral ($\perp$) acceleration components yields

$$a_{\parallel}(t) = \frac{\mathbf{c}^{(1)}(u) \cdot \mathbf{c}^{(2)}(u)}{T^2 \left\| \mathbf{c}^{(1)}(u) \right\|_2^2} \ , \quad \text{and} \tag{6}$$

$$a_{\perp}(t) = \frac{\mathbf{c}^{(1)}(u) \times \mathbf{c}^{(2)}(u)}{T^2 \left\| \mathbf{c}^{(1)}(u) \right\|_2^2} \ , \tag{7}$$

where $\cdot$ and $\times$ denote the dot and cross product, respectively. Equations (5)–(7) explicitly express that velocity and acceleration scale with $\frac{1}{T}$ and $\frac{1}{T^2}$, respectively, as derived in (4). Analogous to the velocity (5), setting $|a_{\parallel}(t)| = a_{\parallel,\max}$ and $|a_{\perp}(t)| = a_{\perp,\max}$ yields the corresponding lap times constrained by the longitudinal and lateral acceleration limits.

Collecting these conditions, the minimum feasible lap time under constant constraints $v_{\max}$, $a_{\parallel,\max}$, and $a_{\perp,\max}$ is

$$T = \max_{u \in [0,1]} \left\{ \frac{v(u)}{v_{\max}}, \sqrt{\frac{|a_{\parallel}(u)|}{a_{\parallel,\max}}}, \sqrt{\frac{|a_{\perp}(u)|}{a_{\perp,\max}}} \right\} \ . \tag{8}$$

This formulation is generalized by allowing the acceleration limits to vary along the path, i.e.,

$$T = \max_{u \in [0,1]} \left\{ \frac{v(u)}{v_{\max}}, \sqrt{\frac{|a_{\parallel}(u)|}{a_{\parallel,\max}(\mathbf{q}(u))}}, \sqrt{\frac{|a_{\perp}(u)|}{a_{\perp,\max}(\mathbf{q}(u))}} \right\} \ . \tag{9}$$

The terms $a_{\parallel,\max}(\mathbf{q}(u)))$ and $a_{\perp,\max}(\mathbf{q}(u)))$ are spatially-varying functions that depend on the current position $\mathbf{q}(u)$. These are provided by our adaptive constraint map, which is introduced in the next section. The extension (9) allows to capture location-specific phenomena directly in offline trajectory planning, such as friction variations, surface irregularities, or banking changes. Each of these disturbances affect the lap time T positively or negatively.

C. Adaptive Constraint Map Framework

The spatially-varying acceleration limits $a_{\parallel,\max}(\mathbf{q}(u))$ and $a_{\perp,\max}(\mathbf{q}(u))$ introduced in the previous section are described by an adaptive constraint map $\mathbf{M} \in \mathbb{R}^{X \times Y}$ with X and Y denoting the number of grid cells horizontally and vertically. This map provides a grid-based representation of the track, where each cell value $[\mathbf{M}]_{x,y} = M_{x,y}$, $x = 1, \ldots, X$, $y = 1, \ldots, Y$, acts as a local scaling factor that modifies the nominal longitudinal and lateral acceleration capacities in the form

$$a_{\parallel,\max}(x,y) = M_{x,y} a_{\parallel,\text{nominal}} \tag{10}$$

$$a_{\perp,\max}(x,y) = M_{x,y} a_{\perp,\text{nominal}} \ . \tag{11}$$

Therefore, heterogeneous track characteristics, such as friction variations, surface irregularities, or banking changes, are consistently expressed as variations in effective acceleration capability. A shared scaling factor $M_{x,y}$ is applied to both longitudinal and lateral acceleration limits, assuming isotropic surface effects. Although separate maps could capture anisotropic friction, a single map reduces complexity and suffices for the repetitive racing scenario considered. This map is updated based on closed-loop execution data, which will be introduced in the subsequent subsections. By continuously updating the scaling factors using the trajectory-tracking feedback, the system adaptively refines local constraints, which in turn impact the computed global trajectory and its lap time, see Section III-B.

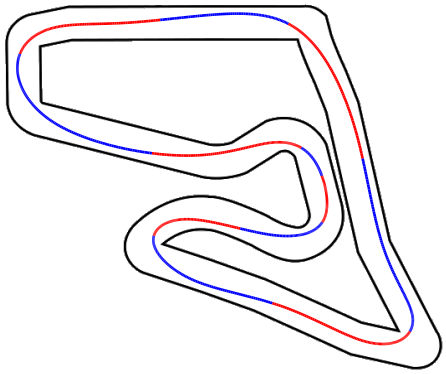

Fig. 3. Track segmentation into acceleration (red) and deceleration (blue) zones for blame region identification. Since the trajectory does not contain any neutral phases, these are not represented in the figure.

D. Blame Region Computation Through Derivative Analysis

If a vehicle encounters an area of reduced friction on the track, the resulting loss of control leads to tracking errors observed later in the trajectory. This temporal and spatial separation between the cause (e.g., a low-friction area) and its impact (e.g., subsequent tracking errors) requires a systematic method to identify the origin of control degradation. Consequently, the error feedback mechanism aims to determine the causes of deviation from the planned trajectory.

Tracking errors are often linked to acceleration changes exceeding the traction limits. The longitudinal acceleration $a_{\parallel}(u)$ derived from the NURBS trajectory parameterization is used to detect sign transitions using

$$\mathcal{S} = \operatorname{sign}(a_{\parallel}) \ , \tag{12}$$

$$\mathcal{Z} = \{\, i \ \mid \ \mathcal{S}[i+1] \neq \mathcal{S}[i] \,\} \ . \tag{13}$$

These zero crossings partition the track into zones of accelerating (positive sign), decelerating (negative sign), and neutral (zero) phases, as illustrated in Fig. 3. This zoning allows errors to be traced back along the trajectory causally linked to the error. To attribute a tracking error to its originating cause, our method associates the error with the most recent acceleration transition along the trajectory. Specifically, the zero-crossing immediately preceding the point of minimum distance is selected. This transition index, denoted $i_{\text{transition}}$, is computed as

$$i_{\text{transition}} = \max\big(\{\, z \in \mathcal{Z} \ \mid \ z < i_{\min} \,\} \cup \{\mathcal{Z}[-1]\}\big) \ . \tag{14}$$

The index $i_{\min}$ is defined via the closest point on the planned path $\mathbf{q}_{\text{ref}}(u)$, where the minimum distance to the actual vehicle position $\mathbf{q}$ occurs, i.e.,

$$\hat{e} = \min_{u \in [0,1]} \|\mathbf{q} - \mathbf{q}_{\text{ref}}(u)\|_2 \ , \tag{15}$$

which is used to quantify tracking errors. The point $i_{\text{transition}}$ marks the boundary of the zone where the error likely originated, enabling feedback of the error to the relevant dynamic phase.

The *blame region* is defined as the trajectory section between $i_{\text{transition}}$ and $i_{\min}$. This region establishes a direct link between observed tracking errors and distinct trajectory characteristics, such as excessive acceleration demands, within a consistent acceleration phase. By restricting the analysis to such a region, our method realizes targeted modifications of the trajectory where they are most effective.

E. Error Feedback and Constraint Learning

The error feedback mechanism is formulated in a general way. Let $\hat{e}$ denote a general error signal obtained from the execution performance. This signal is modulated to promote either conservative adaptation or progressive improvement, depending on whether the observed error is below or above a given threshold e_{th}. Thus, the modulated error signal e is defined as

$$e = \begin{cases} w^{+}\hat{e}, & \text{if } \hat{e} < e_{\text{th}} \\ w^{-}\hat{e}, & \text{if } \hat{e} \geq e_{\text{th}} \ . \end{cases} \tag{16}$$

This signal is fed back to the identified blame region, affecting all map cells $M_{x,y}$ within a radius of the relevant trajectory points.

The constraint maps (10) and (11) are updated using a method similar to a Kalman filter [21], where each spatial cell (x, y), $x = 1, \ldots, X$, $y = 1, \ldots, Y$, is treated as an independent state with associated uncertainty $[\mathbf{V}]_{x,y} = V_{x,y}$, cf. Fig. 1. The update equations are formulated as

$$K_{x,y} = \frac{V_{x,y}}{V_{x,y} + R} \ , \tag{17}$$

$$M^{+}_{x,y} = M^{-}_{x,y} + K_{x,y}\, e \ , \tag{18}$$

$$V^{+}_{x,y} = (1 - K_{x,y}) V^{-}_{x,y} + Q \ , \tag{19}$$

where R is the measurement noise variance, Q is the process noise variance, and $[\mathbf{K}]_{x,y} = K_{x,y}$ represents the gain. The superscripts '$-$' and '$+$' indicate values before and after the update, respectively.

F. Model Predictive Control Implementation

Given a reference trajectory $\mathbf{q}_{\text{ref}}(t)$, the system is controlled using a trajectory-tracking MPC. Unlike a classical MPC, which jointly optimizes path and timing, a trajectory-tracking MPC focuses solely on following the given trajectory, leveraging the embedded temporal profile [22], [23]. This decoupling simplifies the optimization process and reduces online computational effort [24], making it more suitable for real-time applications in autonomous systems.

The MPC used in this work utilizes a kinematic single-track vehicle model $\mathbf{f}(\mathbf{x}, \mathbf{u})$ with the state $\mathbf{x} = [x, y, \theta, \delta, v]^{\mathrm{T}}$ (comprising the position x and y, heading θ, steering angle δ, and velocity v), and the control input $\mathbf{u} = [a, \dot{\delta}]^{\mathrm{T}}$ (containing the longitudinal acceleration a and the steering rate $\dot{\delta}$) [25]. At each time step, the MPC computes control actions that minimize the deviation from the reference trajectory while respecting vehicle and actuator constraints. The resulting *tracking errors* (15) provide feedback signals for adaptive trajectory adjustments in subsequent iterations.

G. Trajectory Design with Closure Constraints

Our approach embeds closure constraints directly into the NURBS trajectory representation, ensuring that the optimization procedure only produces physically realizable racing trajectories. Since racing circuits are inherently closed, the trajectory must satisfy continuity conditions at the junction between the end and start of the lap.

For cubic NURBS ($p = 3$), we enforce C^2 continuity at the closure point by analytically constraining the last three control points based on the first three. The detailed derivation of these analytical constraints is provided in the source code of our implementation. These constraints guarantee smooth transitions in position, velocity, and acceleration at the lap boundary, preventing discontinuities that would violate vehicle dynamics. Incorporating the closure constraints reduces the optimization degrees of freedom (DoF) by three control points per spatial dimension (a total of six DoF for 2-D trajectories).

Given these constraints, the trajectory optimization operates over a reduced parameter space $\boldsymbol{\theta}$ containing only the free parameters, reading as

$$\boldsymbol{\theta} = (\mathbf{p}_{\text{free}}, \mathbf{w}_{\text{free}}, \mathbf{u}_{\text{free}}) , \tag{20}$$

where $\mathbf{p}_{\text{free}} \in \mathbb{R}^{2(n-2)}$ are the control points not analytically determined by the closure constraints (i.e., $\mathbf{p}_0, \ldots, \mathbf{p}_{n-3}$), $\mathbf{w}_{\text{free}} \in \mathbb{R}^{n-2}$ are the corresponding adjustable weight parameters that control the curve locally, and $\mathbf{u}_{\text{free}} \in \mathbb{R}^{m-2(p+1)}$ are the interior knot positions that define the parameterization distribution along the curve. This reduced parameterization allows the optimizer to systematically explore the space of feasible trajectories while automatically satisfying continuity requirements.

H. Global Trajectory Optimization

The NURBS representation detailed in Section III-A, the adaptive local constraints detailed in Section III-B, and the closure constraints in Section III-G are solved as a global trajectory optimization problem. The optimization utilizes CMA-ES [18] to search the parameter space $\boldsymbol{\theta}$ from (20). The main advantages of our approach are that the lap time is implicitly solved and the optimizer can systematically only sample feasible trajectories. The objective function for this optimization problem is formulated as

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \; T(\boldsymbol{\theta}) + \lambda_{\text{dist}} \Phi_{\text{distance}} + \lambda_{\text{curv}} \Phi_{\text{curvature}} , \tag{21}$$

which combines the trajectory duration $T(\cdot)$ with geometric penalties for geometric constraint violations. Segments that lie outside the track boundaries are strongly penalized via λ_{dist}, which increases proportionally with the squared distance from the track centerline, guiding the optimizer to generate fast trajectories that remain within the racing limits. In addition, excessive curvature is penalized via $\lambda_{\text{curv}} \Phi_{\text{curvature}}$, ensuring that the local curvature $\kappa(u)$ does not exceed the admissible maximum $\kappa_{\max}$, which corresponds to respecting the minimal turning radius of the vehicle. The curvature penalty term is chosen as

$$\Phi_{\text{curvature}} = \int_0^1 \max\big(0, \kappa(u) - \kappa_{\max}\big)^2 \mathrm{d}u , \tag{22}$$

where $\kappa(u)$ denotes the curvature at path parameter u along the trajectory. This formulation penalizes only those segments where the curvature exceeds the vehicle's maximum admissible curvature $\kappa_{\max}$.

The time scaling computation (6)–(9) incorporates dynamic constraints using the adaptive constraint map. When acceleration limits reflect the learned spatial variations, the optimization process accounts for regions with reduced traction while exploiting areas with higher available grip. This coupling between trajectory optimization and constraint learning establishes a feedback mechanism in which performance of the trajectory execution influences the subsequent trajectory generation in a global manner. Additionally, an exploration component may be added, leveraging the uncertainty map $\mathbf{V}$ in the constraint estimates to guide the trajectory towards regions with potentially higher friction.

Fig. 4. F1Tenth experimental platform with Asus NUC and Hokuyo LiDAR.

IV. Experimental Setup and Results

We evaluate our approach both in simulation and real hardware using a F1Tenth racecar [26], see Fig. 4. The evaluation assesses the trajectory quality, optimization performance, and the adaptation to varying surface conditions. We compare our static as well as adaptive approach with two baselines, i.e., (i) a minimum-curvature trajectory (*Min Curvature*) [27], (ii) a raceline optimized using an approximate homeomorphism (*Homeomorphism*) [28].

First, all baseline approaches are evaluated in simulation on a variety of static racetracks [29] to assess the quality of the generated trajectories and the trajectory representation. Second, we measure lap times with and without adaptive approach. Third, on a racetrack with low-friction regions we run a detailed analysis to quantify how quickly our approach is able to adapt. Fourth, we conduct real-world experiments on a track with varying surface properties. In this hardware experiment, global friction is varied using different wheel sets, enabling us to evaluate how effectively our method

TABLE I

LAP TIMES IN S FOR DIFFERENT TRAJECTORIES AND RACETRACKS.

Map	Ours Static	Ours Adaptive	Min Curvature Static	Homeomorphism Static
F1Aut	20.02	**16.54**	22.12	21.83
Wall1	16.82	**15.71**	17.32	16.92
Levine	11.08	**10.42**	11.39	12.18
Operngasse	7.49	**6.24**	8.37	7.58

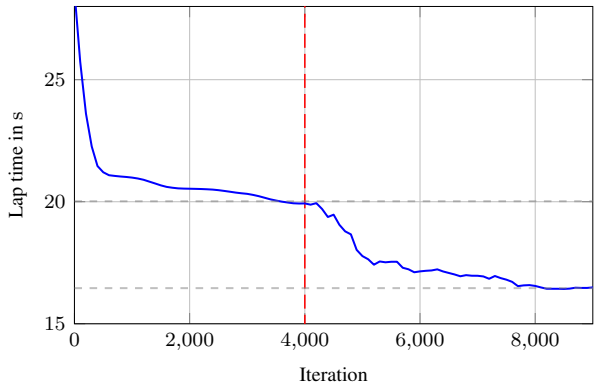

Fig. 5. Optimizer convergence on F1Aut track. Red line: Error feedback activation. Convergence is achieved within 8 laps, with 500 iterations/lap.

adapts to varying friction limits along the trajectory. Running concurrently with trajectory execution, the optimizer enables online adaptation and achieves approximately 30 iterations per second on a Ryzen 9 7900X3D. The trajectory is continuously updated in the background, with smooth transitions to newly optimized references to prevent abrupt changes in the control signal and degradation of lap time or vehicle performance. Vehicle parameters follow the standard F1Tenth platform specification [26]. Key settings are the CMA-ES population size of 45, map resolution of $0.05\,\mathrm{m}$, process noise $Q = 0.01$, measurement noise $R = 0.5$, MPC horizon $N = 20$, and sampling time $\Delta t = 0.05\,\mathrm{s}$.

A. Performance Comparison

We evaluate the impact of our method on lap performance across four distinct track configurations with varying complexity and length. Table I presents the lap times for both the baseline optimizations and our approach with and without adaptation.

The results demonstrate consistent performance improvements across all tested circuits. Our adaptation method improves the lap time from $16.82\,\mathrm{s}$ to $15.71\,\mathrm{s}$ on the Wall1, which corresponds to a relative improvement of $6.60\,\%$. The most significant improvements are observed on the long F1Aut track with high-speed corners, where our approach reduces the lap time from $20.02\,\mathrm{s}$ to $16.54\,\mathrm{s}$, corresponding to an improvement of $17.38\,\%$. The smaller Operngasse and Levine tracks show improvements of $16.69\,\%$ and $5.96\,\%$ respectively, indicating that the benefits scale effectively across different track characteristics. Compared to the *Min Curvature* method, which yields lap times of $22.12\,\mathrm{s}$ (F1Aut), $17.32\,\mathrm{s}$ (Wall1), $11.39\,\mathrm{s}$ (Levine), $8.37\,\mathrm{s}$ (Operngasse), respectively, our approach consistently achieves faster laps. In contrast, the lap times corresponding to the *Homeomorphism* approach exhibit up to $9.93\,\%$ performance loss against our static approach. The performance gains of our approach listed in Table I are achieved while maintaining consistent tracking error throughout the trajectory, ensuring that the improved lap times do not compromise vehicle control and safety.

B. Impact of Adaptive Constraint Map

The impact of the adaptive constraint map described in Sections III-C to III-E is shown in Fig. 5, which depicts the convergence behavior in simulation for the F1Aut track.

The first 4000 optimizer iterations are completed without providing error feedback to our optimizer. The car converges to a lap time of $20.02\,\mathrm{s}$. As soon as we enable the feedback indicated by the vertical dashed redline, the optimizer converges after 8 laps (roughly 500 iterations/lap) to a lap time of $16.54\,\mathrm{s}$, a relative improvement of $17.38\,\%$.

C. Convergence Analysis with Local Friction Variations

To evaluate the optimizer's responsiveness to localized surface variations, we introduce two regions with $80.0\,\%$ less friction compared to the nominal track surface. Fig. 6 illustrates these regions as well as the evolution of the optimized trajectory over three laps, in which the spatial convergence occurs. Initially unaware of friction variations, the vehicle traverses the first low-friction region using the nominal trajectory in lap 1. After feeding back the tracking error between the red-dotted reference trajectory $\mathbf{q}_{\mathrm{ref}}$ and the executed blue trajectory $\mathbf{q}$, the blame region (shown in blue) is updated. The optimizer then adapts both the trajectory shape and its timing, enabling the vehicle to successfully navigate the second previously unknown low-friction area during lap 2, where the same procedure is repeated. By lap 3, the optimizer converges to an optimal path that effectively navigates the reduced-grip corner, indicated by the blue line. This demonstrates quick adaptation to localized surface variations of our method.

D. Real-World Adaptive Experiments

We validated our approach using the real-time system depicted in Fig. 4 and we documented the experiments as a video at www.acin.tuwien.ac.at/7rac. To demonstrate the impact of the proposed method, we tested global friction variations by employing different tire configurations on the vehicle. Three scenarios were evaluated: a low-friction configuration with four off-road tires (*Low*), a medium-friction configuration with slick front tires and off-road rear tires (*Medium*), and a high-friction configuration with four slick tires (*High*). The baseline performance without friction parameterization yielded a conservative lap time of $7.53\,\mathrm{s}$ on a circular track with varying width, as shown in Fig. 7.

Upon activation of the error-feedback mechanism, consistent performance improvements were observed across all tire configurations. In order to achieve convergence, the error feedback threshold was set based on the position measurement accuracy. Therefore, a balance between position measurement noise and maximum admissible deviation from the reference trajectory was chosen for e_{th}, with convergence being achieved after approximately 10 laps. The system achieved a minimum lap time of $5.29\,\mathrm{s}$ under high-friction

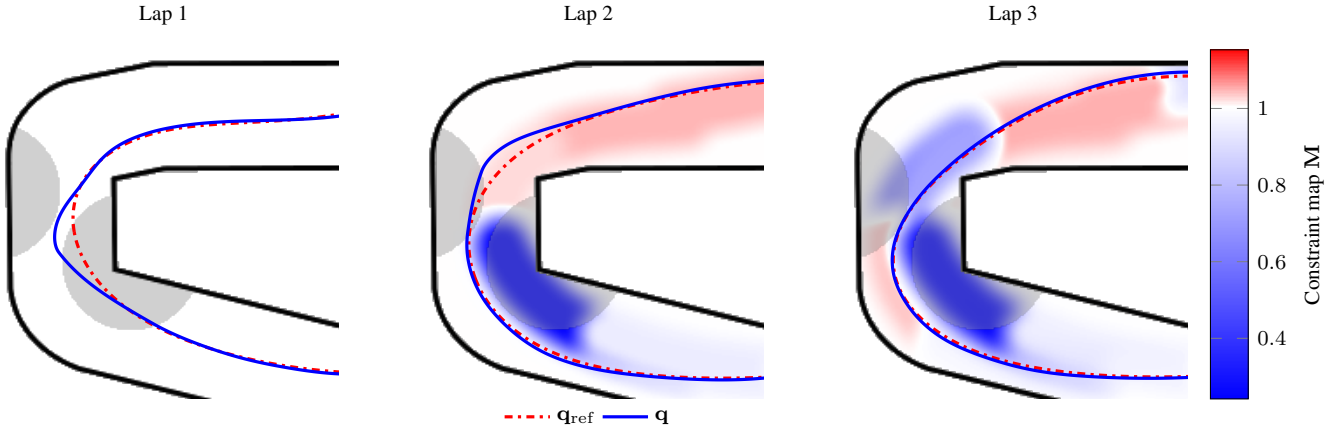

Fig. 6. Illustration of the two artificially introduced low-friction regions (highlighted in gray) and the corresponding evolution of the optimized trajectory across three laps, showing the nominal planned trajectory (red-dotted line), the executed trajectory (blue line), and updated constraint map (shaded blue and red regions).

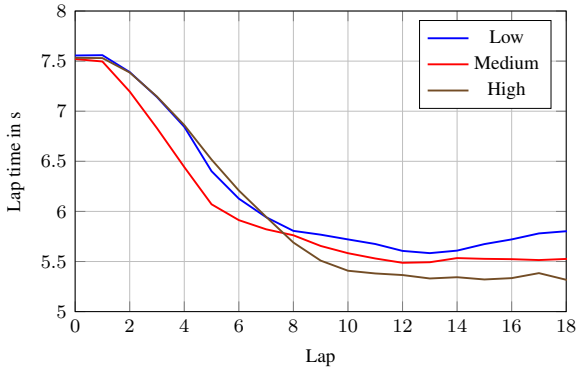

Fig. 7. Lap times for three tire configurations (Low/Medium/High friction). Error feedback activated at lap 0.

conditions, while the medium and low friction configurations converged to lap times of $5.56\,\mathrm{s}$ and $5.73\,\mathrm{s}$, respectively. This corresponds to an adequate adaptation to the different friction conditions, resulting in a $7.60\,\%$ lap time improvement without explicit friction parametrization or environment sensing. The resulting trajectory shapes after convergence are illustrated in Fig. 8. Due to the circuit layout, the vehicle performance was primarily constrained by the lateral acceleration $a_\perp$. Fig. 8 also shows the exploration of the local acceleration limits while maximizing their utilization. For higher friction conditions, the algorithm generates trajectories with increased average velocity and slightly greater path length, whereas lower friction configurations result in reduced speed and a more circular trajectory shape. Note that when acceleration limits are set to maximum or mean acceleration values, the controller fails to accurately follow the generated trajectory, resulting in performance degradation or, in worst-case scenarios, vehicle instability resulting in a crash. Notably, the system does not require manual tuning of the constraint parameters. Instead, it automatically discovers the feasible limits for each wheel configuration.

V. Conclusion

The closed-loop raceline optimization framework proposed in this work systematically leverages execution feedback to global trajectory improvements for autonomous racing applications, shifting from controller-centric adaptation to trajectory-centric optimization. Our proposed approach integrates NURBS trajectory parameterization with CMA-ES optimization and learning of adaptive spatial constraints. This allows continuous refinement of both trajectory shape and timing using the MPC tracking error as feedback. The spatial-constraint learning mechanism translates tracking errors into location-specific constraint maps without explicit environmental sensing, while performing causal attribution of errors to specific trajectory segments using a so-called *blame region* computation.

Experimental validation demonstrates consistent lap time improvements ranging from $5.96\,\%$ to $17.38\,\%$ across diverse track configurations in simulation and real-world F1Tenth experiments. We achieve convergence after approximately 10 laps and performance improvements across varying friction conditions. The proposed method shows significant performance enhancement compared to static trajectory approaches, highlighting its potential in applications for autonomous vehicles where structured repetition enables systematic learning.Future work will investigate controller dependency by systematic evaluation with different underlying control architectures, extend beyond positional error metrics to incorporate temporal dependencies, and replace the Kalman-inspired update mechanism with learning-based approaches that automatically identify origin regions of error.

References

[1] J. L. Vázquez, M. Brühlmeier, A. Liniger, A. Rupenyan, and J. Lygeros, "Optimization-Based Hierarchical Motion Planning for Autonomous Racing," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2397–2403.

[2] L. Ögretmen, M. Rowold, A. Langmann, and B. Lohmann, "Sampling-Based Motion Planning with Online Racing Line Generation for Autonomous Driving on Three-Dimensional Race Tracks," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2024, pp. 811–818.

[3] H. Xue, E. L. Zhu, J. M. Dolan, and F. Borrelli, "Learning Model Predictive Control with Error Dynamics Regression for Autonomous Racing," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 13 250–13 256.

[4] D. R. Gomes, M. A. Botto, and P. U. Lima, "Learning-based Model Predictive Control for an Autonomous Formula Student Racing Car," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 12 556–12 562.

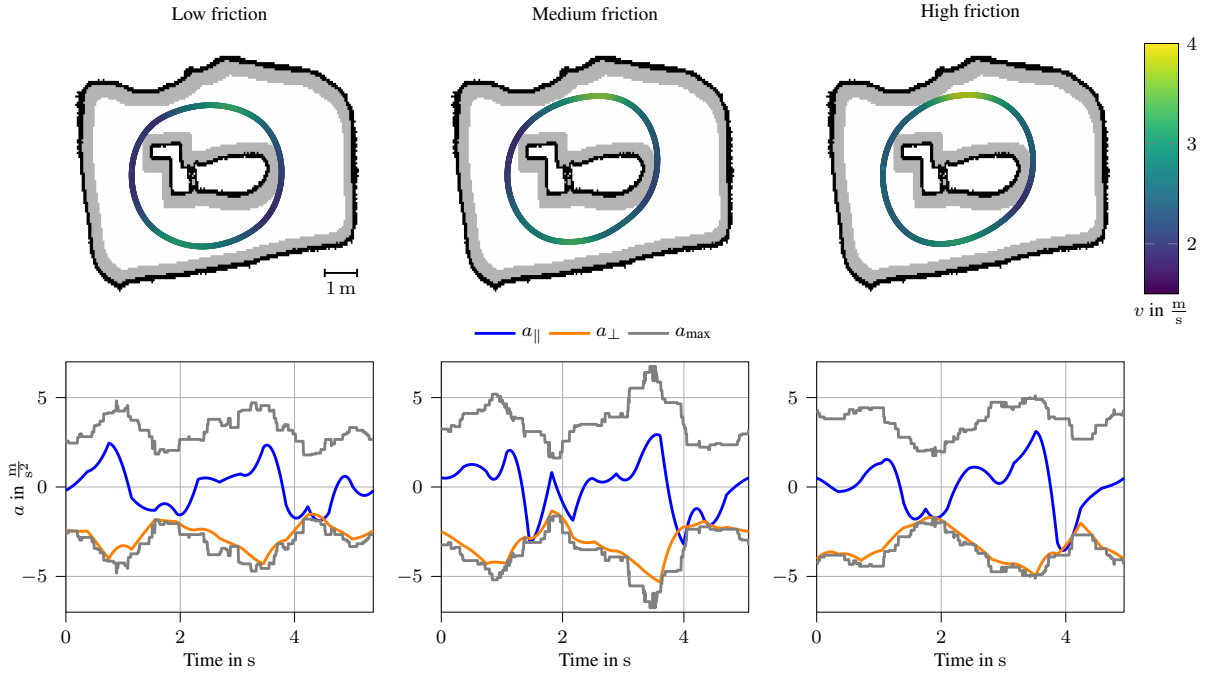

Fig. 8. Optimized trajectories under different global friction conditions showing the lateral acceleration $a_{\perp}$, the longitudinal acceleration $a_{\parallel}$, and the acceleration constraints $a_{\max}$ from the constraint map. Higher-friction setups produce trajectories with increased average velocity and slightly longer path lengths, while lower-friction setups result in reduced speeds and more circular trajectories.

[5] D. Kalaria, Q. Lin, and J. M. Dolan, "Adaptive Planning and Control with Time-Varying Tire Models for Autonomous Racing Using Extreme Learning Machine," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 10 443–10 449.

[6] J. Ning and M. Behl, "Scalable Deep Kernel Gaussian Process for Vehicle Dynamics in Autonomous Racing," in *Proc. Conference on Robot Learning*. PMLR, 2023, pp. 761–773.

[7] T. Nagy, A. Amine, T. X. Nghiem, U. Rosolia, Z. Zang, and R. Mangharam, "Ensemble Gaussian Processes for Adaptive Autonomous Driving on Multi-friction Surfaces," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 494–500, 2023.

[8] N. Dal Bianco, E. Bertolazzi, F. Biral, and M. Massaro, "Comparison of direct and indirect methods for minimum lap time optimal control problems," *Vehicle System Dynamics*, vol. 57, no. 5, pp. 665–696, 2019.

[9] M. Massaro and D. Limebeer, "Minimum-lap-time optimisation and simulation," *Vehicle System Dynamics*, vol. 59, no. 7, pp. 1069–1113, 2021.

[10] D. Brayshaw and M. Harrison, "A quasi steady state approach to race car lap simulation in order to understand the effects of racing line and centre of gravity location," *Proc. of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 219, no. 6, pp. 725–739, 2005.

[11] J. Betz, *et al.*, "TUM autonomous motorsport: An autonomous racing software for the Indy Autonomous Challenge," *Journal of Field Robotics*, vol. 40, no. 4, pp. 783–809, 2023.

[12] M. Veneri and M. Massaro, "A free-trajectory quasi-steady-state optimal-control method for minimum lap-time of race vehicles," *Vehicle System Dynamics*, vol. 58, no. 6, pp. 933–954, 2020.

[13] M. Rowold, L. Ögretmen, U. Kasolowsky, and B. Lohmann, "Online Time-Optimal Trajectory Planning on Three-Dimensional Race Tracks," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2023, pp. 1–8.

[14] G. Perantoni and D. J. Limebeer, "Optimal control for a formula one car with variable parameters," *Vehicle System Dynamics*, vol. 52, no. 5, pp. 653–678, 2014.

[15] N. Dal Bianco, R. Lot, and M. Gadola, "Minimum time optimal control simulation of a GP2 race car," *Proc. of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 232, no. 9, pp. 1180–1195, 2018.

[16] F. Werner, A.-K. Schwehn, M. Lienkamp, and J. Betz, "Gripmap: An efficient, spatially resolved constraint framework for offline and online trajectory planning in autonomous racing," in *Intelligent Vehicles Symposium*, 2025, pp. 103–110.

[17] F. Christ, A. Wischnewski, A. Heilmeier, and B. Lohmann, "Time-optimal trajectory planning for a race car considering variable tyre-road friction coefficients," *Vehicle System Dynamics*, vol. 59, no. 4, pp. 588–612, 2021.

[18] N. Hansen, "The CMA evolution strategy: A tutorial," *arXiv preprint arXiv:1604.00772*, 2016.

[19] N. R. Kapania and J. C. Gerdes, "Path Tracking of Highly Dynamic Autonomous Vehicle Trajectories via Iterative Learning Control," in *Proc. IEEE American control conference (ACC)*, 2015, pp. 2753–2758.

[20] U. Rosolia and F. Borrelli, "Learning How to Autonomously Race a Car: A Predictive Control Approach," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2713–2719, 2019.

[21] D. Simon, *Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches*. New Jersey, USA: John Wiley & Sons, 2006.

[22] S. Yu, M. Hirche, Y. Huang, H. Chen, and F. Allgöwer, "Model predictive control for autonomous ground vehicles: a review," *Autonomous Intelligent Systems*, vol. 1, no. 1, p. 4, 2021.

[23] Q. Shi, J. Zhao, A. El Kamel, and I. Lopez-Juarez, "MPC Based Vehicular Trajectory Planning in Structured Environment," *IEEE Access*, vol. 9, pp. 21 998–22 013, 2021.

[24] J. Zhang, Z. Wang, L. Li, K. Yang, and Y. Lu, "Trajectory tracking control of autonomous vehicles based on event-triggered model predictive control," *IET Intelligent Transport Systems*, vol. 18, no. S1, pp. 2856–2868, 2024.

[25] D. Schramm, M. Hiller, and R. Bardini, *Vehicle Dynamics: Modeling and Simulation*. Berlin Heidelberg, Germany: Springer, 2014.

[26] M. O'Kelly, H. Zheng, D. Karthik, and R. Mangharam, "F1TENTH: An Open-source Evaluation Environment for Continuous Control and Reinforcement Learning," *Proc. of Machine Learning Research*, vol. 123, pp. 77–89, 2020.

[27] A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp, and B. Lohmann, "Minimum curvature trajectory planning and control for an autonomous race car," *Vehicle System Dynamics*, vol. 58, no. 10, pp. 1497–1527, 2020.

[28] J. Klapálek, A. Novák, M. Sojka, and Z. Hanzálek, "Car Racing Line Optimization with Genetic Algorithm using Approximate Homeomorphism," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 601–607.

[29] I. Charles, H. Maghsoumi, and Y. Fallah, "Advancing autonomous racing: A comprehensive survey of the roboracer (f1tenth) platform," in *2025 6th International Conference on Artificial Intelligence, Robotics and Control (AIRC)*, 2025, pp. 207–213.