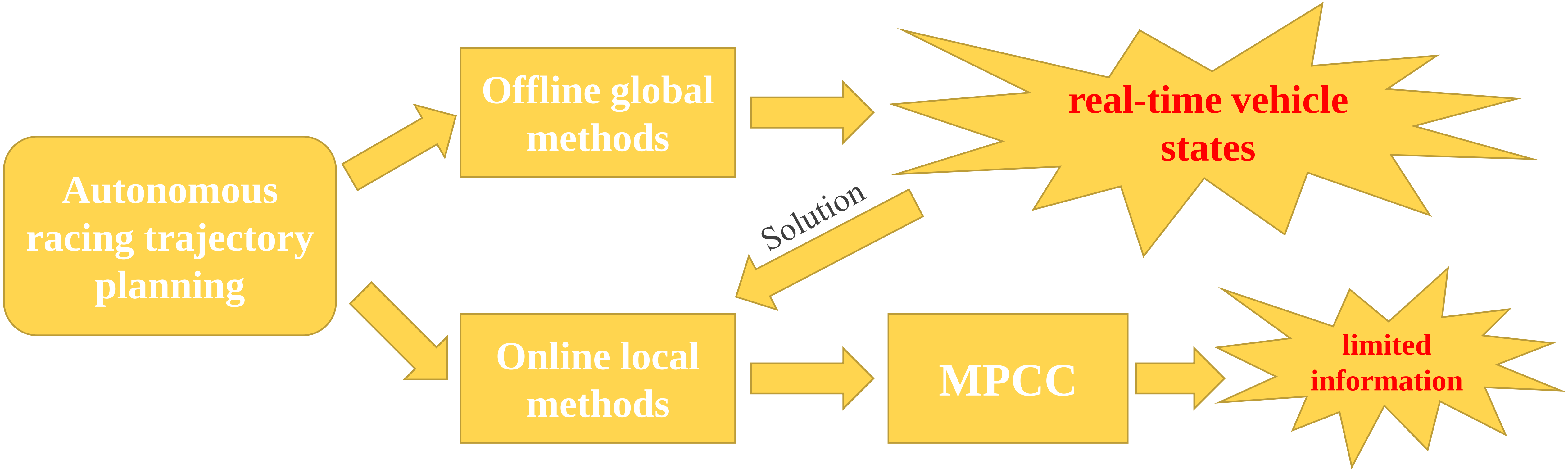
Reduce Lap Time for Autonomous Racing with Curvature-Integrated MPCC Local Trajectory Planning Method

Zhouheng Li, Lei Xie, Cheng Hu, Hongye Su

Introduction



Liniger A, Domahidi A, Morari M. **Optimization-based autonomous racing of 1: 43 scale RC cars**[J]. Optimal Control Applications and Methods, 2015, 36(5): 628-647.

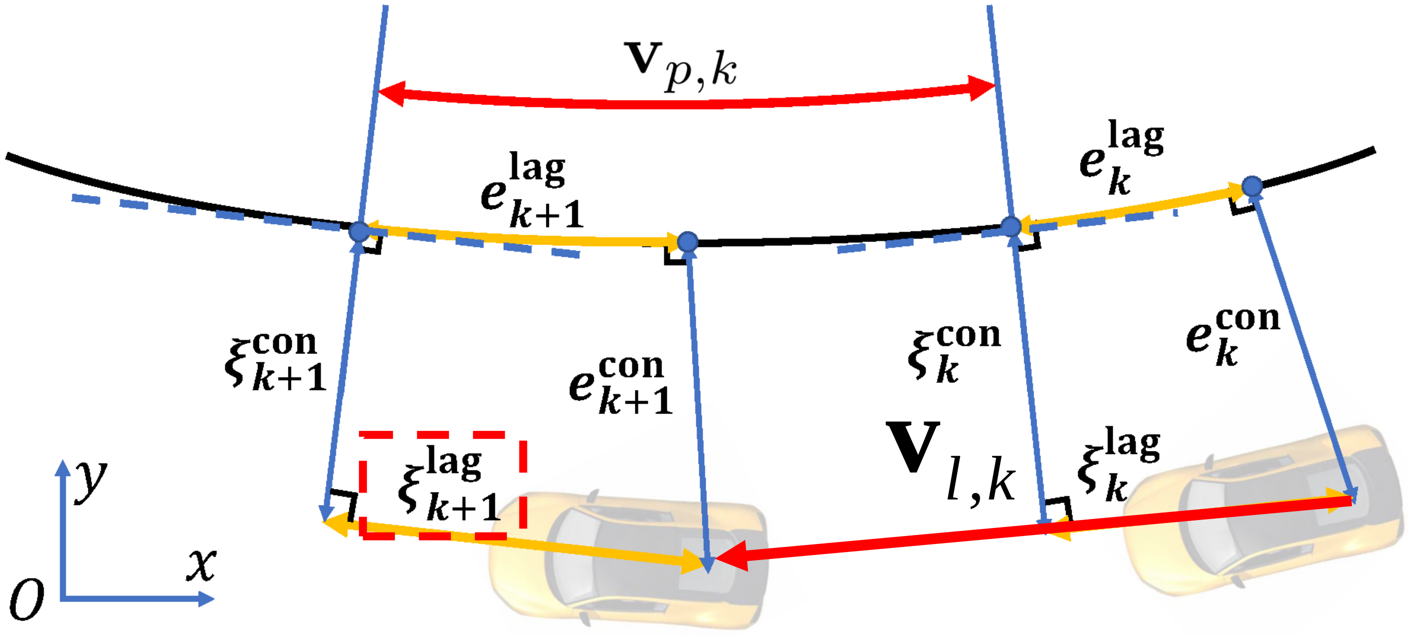
Heilmeier A, Wischnewski A, Hermansdorfer L, et al. **Minimum curvature trajectory planning and control for an autonomous race car**[J]. Vehicle System Dynamics, 2019.

Motivation

$$\mathbf{v}_{p,k} \approx \mathbf{v}_{l,k}$$

$$J_{\mathrm{MPCC}} = \sum_{k=1}^{N_p} \left(\|\xi_k\|_Q^2 - \boxed{\gamma \cdot \mathbf{v}_{p,k} \cdot T_s} \right) + \sum_{k=1}^{N_c-1} \|\Delta u_k\|_{R_1}^2 + \sum_{k=1}^{N_c} \|u_k - u_{\mathrm{ref}}\|_{R_2}^2$$

$$
\begin{aligned}
\min_{u} \quad & J_{\mathrm{MPCC}} \\
\text{s.t.} \quad & \zeta(0) = \zeta_{\mathrm{cur}} && \text{(4a)} \\
& \zeta(k+1) = f_d(\zeta(k), u(k)) && \text{(4b)} \\
& \zeta_{\min} \leq \zeta(k) \leq \zeta_{\max} && \text{(4c)} \\
& u_{\min} \leq u(k) \leq u_{\max} && \text{(4d)}
\end{aligned}
$$



Contribution

How to incorporate **global racetrack information** in MPCC's autonomous racing framework?

Contribution

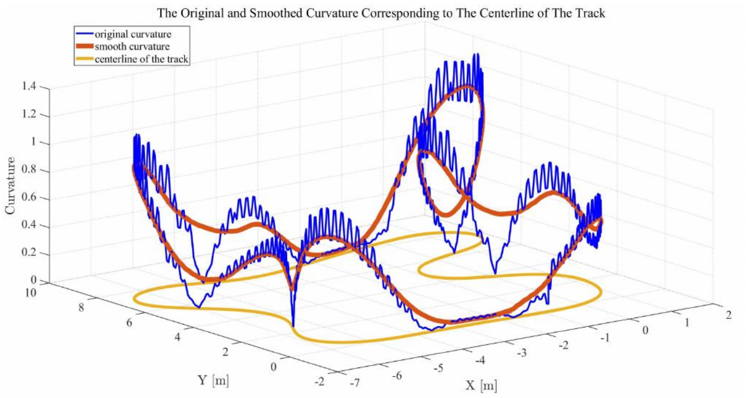
Our solution: Curvature-Integrated MPCC Local Trajectory Planning (**CiMPCC**)

- A practical method for **integrating curvature** into optimization problems;
- A novel mapping method between **curvature and velocity**.

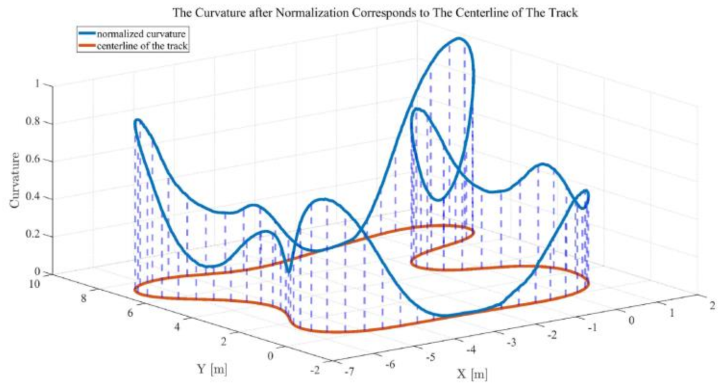
Curvature smoothing → Normalization of curvature → Curvature velocity mapping → optimization problems

Method

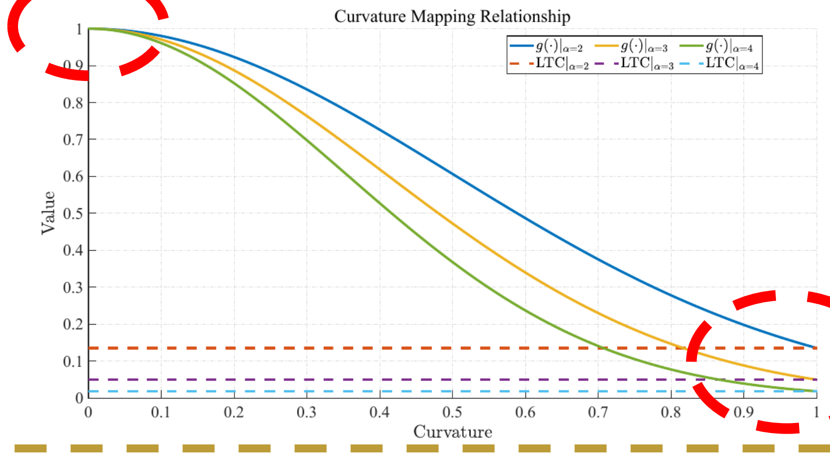
$$\boldsymbol{\kappa}_i = \frac{\|\Delta x_i \cdot \Delta^2 x_i - \Delta^2 x_i \cdot \Delta y_i\|}{(\Delta x_i^2 + \Delta y_i^2)^{\frac{3}{2}}}$$



$$\mathbf{K}_i = \frac{1}{\mathbf{w}} \sum_{m=i-\frac{\mathbf{w}-1}{2}}^{i+\frac{\mathbf{w}-1}{2}} \boldsymbol{\kappa}_m \qquad \mathbf{K}_i^{\mathrm{n}} = \frac{\mathbf{K}_i - \mathbf{K}_{\min}}{\mathbf{K}_{\max} - \mathbf{K}_{\min}}$$



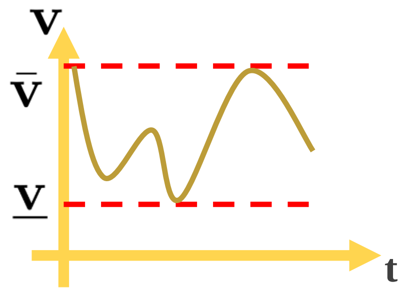
$$g(\mathbf{K}_i^{\mathrm{n}}) = e^{-\boldsymbol{\alpha}\cdot(\mathbf{K}_i^{\mathrm{n}})^2}, \boldsymbol{\alpha} > 0, i \in [1, N]$$



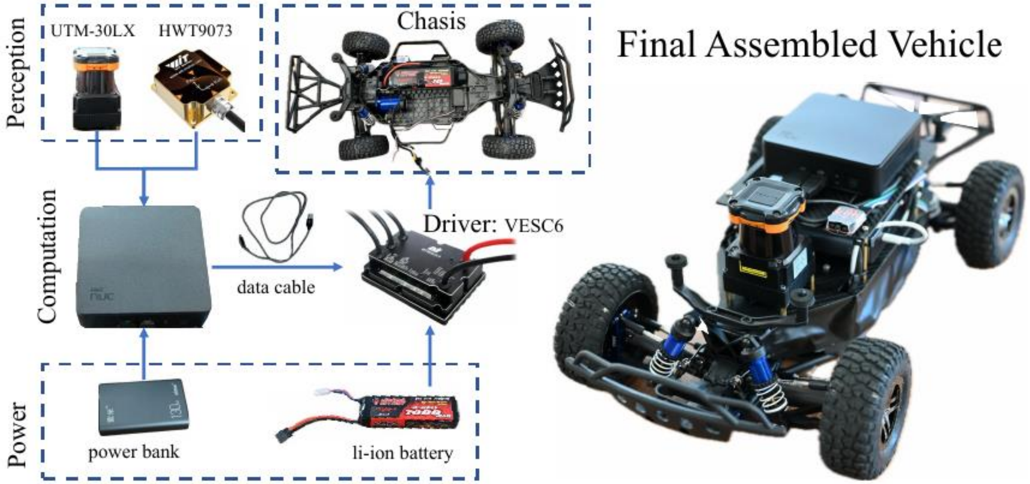
$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_l & \mathbf{v}_p \end{bmatrix}$$

$$J_{\mathrm{Ci}} = \sum_{k=1}^{N_p} (1-\boldsymbol{\beta}) \cdot \|\mathbf{v}_k - \underline{\mathbf{v}}\|_{R_3}^2 + \boldsymbol{\beta} \cdot \|\mathbf{v}_k - \bar{\mathbf{v}}\|_{R_3}^2, \boldsymbol{\beta} = g(\mathbf{K}_{\mathrm{cur}}^{\mathrm{n}})$$

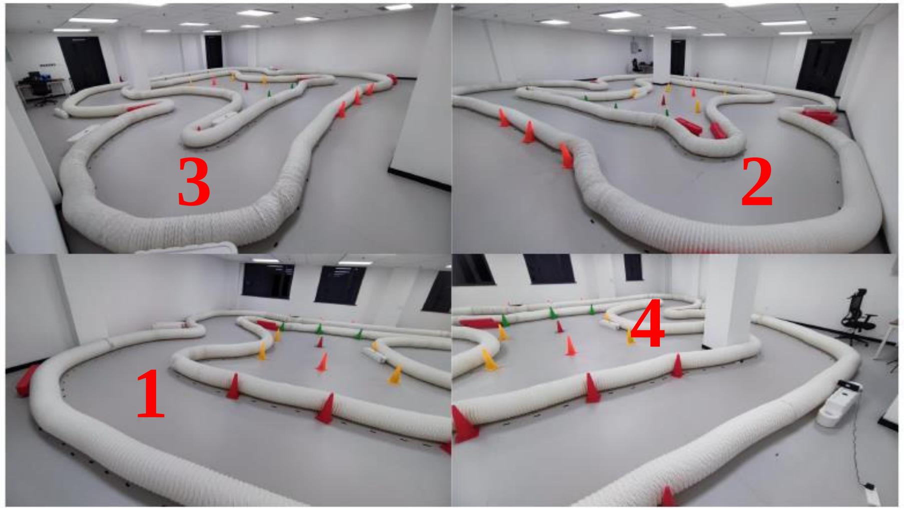
$$\begin{aligned} \min_u \quad & \boxed{J_{\mathrm{Ci}}} + J_{\mathrm{MPCC}} \\ \text{s.t.} \quad & \zeta(0) = \zeta_{\mathrm{cur}} \\ & \zeta(k+1) = f_d(\zeta(k), u(k)) \\ & \zeta_{\min} \le \zeta(k) \le \zeta_{\max} \\ & u_{\min} \le u(k) \le u_{\max} \end{aligned}$$



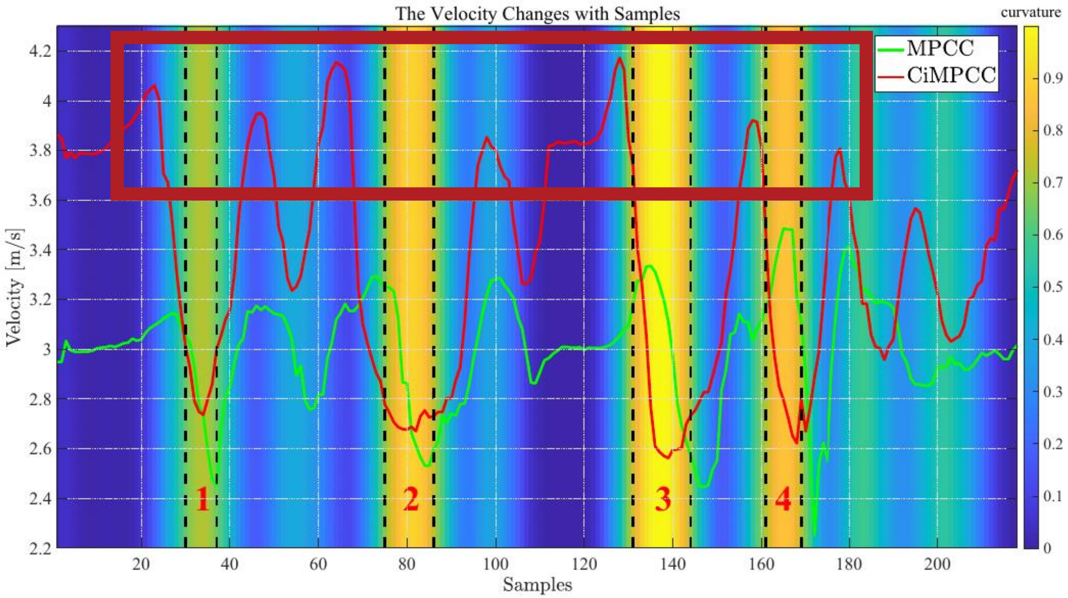
Experimental result

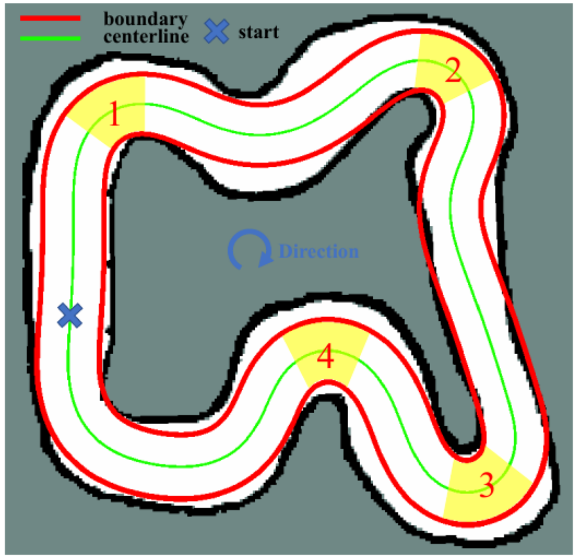


Driving, Drifting, and Racing Automation (DDRA)



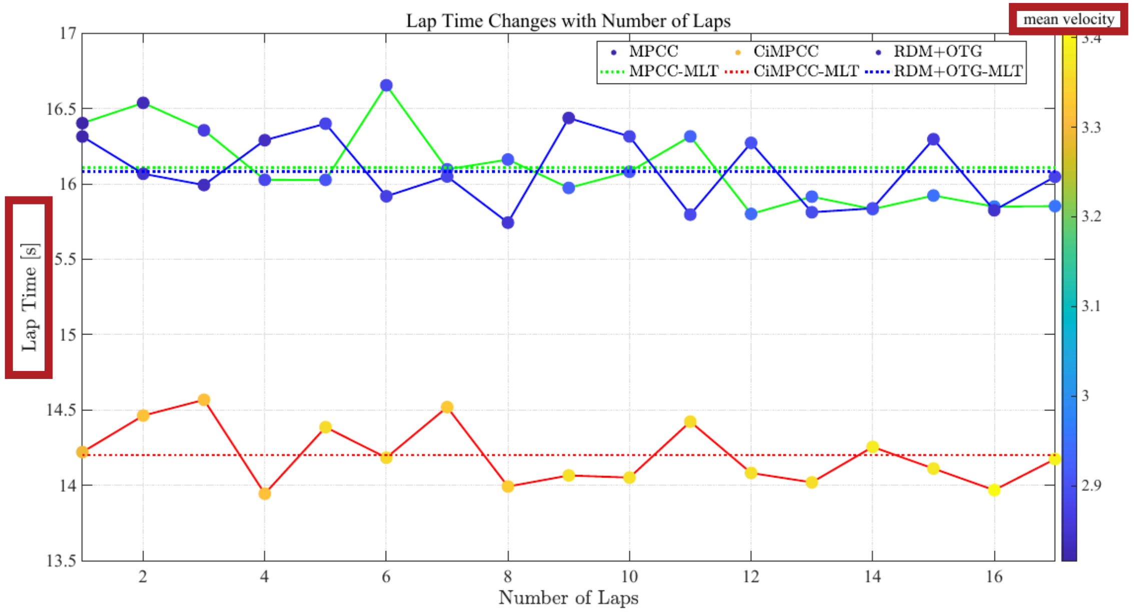
Based on the comparison and analysis of the control laws, it can be concluded that CiMPCC is effective for modeling the normalized smooth curvature of the racetrack centerline.

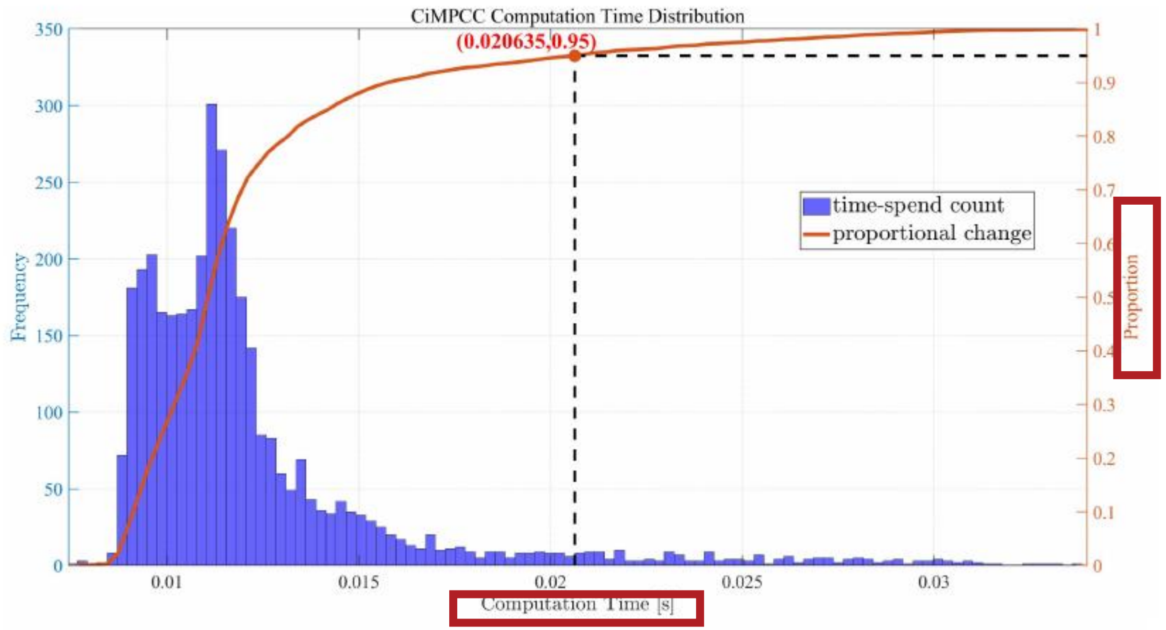




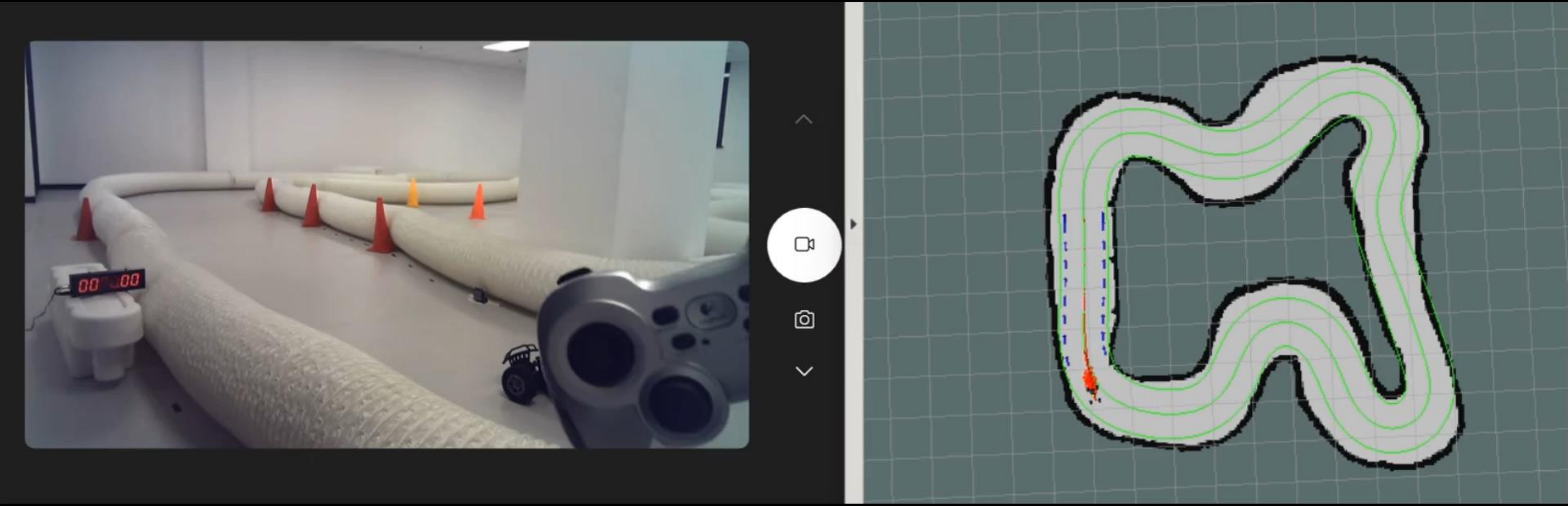
Experimental result

- Subsequently, we continuously performed the DDRA for a total of **seventeen laps** of autonomous racing.
- According to the percentage of performance improvement it can be seen that CiMPCC outperforms the traditional MPCC and RDM+OTG (based on path velocity decomposition) methods in terms of **mean velocity and lap time**. This also shows the stability of the CiMPCC method.
- This shows that the computation efficiency of CiMPCC is sufficient for **real-time computation**.





Conclusion



- The main innovation lies in the method of integrating the racetrack centerline into the optimization problem.
- The experimental results from long-term autonomous racing demonstrate that the CiMPCC method decreases the mean lap time by **11.8%** compared to the traditional MPCC method.

Future work

Velocity Prediction MPCC (VPMPCC)

Automatically tuning parameters using BO

Additional **velocity decisions**

Mean velocity achieved **93.18%** of limits

Trajectory filter for BO training

Fast planning of **high-quality trajectories**

Mean trajectory planning time is **7.04 ms** with **transferable parameters**

Objective Function adapted to **Racing (OFR)**

Improve **training efficiency** of BO

Imporve training efficiency by **42.86%**

ACCEPT MY ENDLESS GRATITUDE