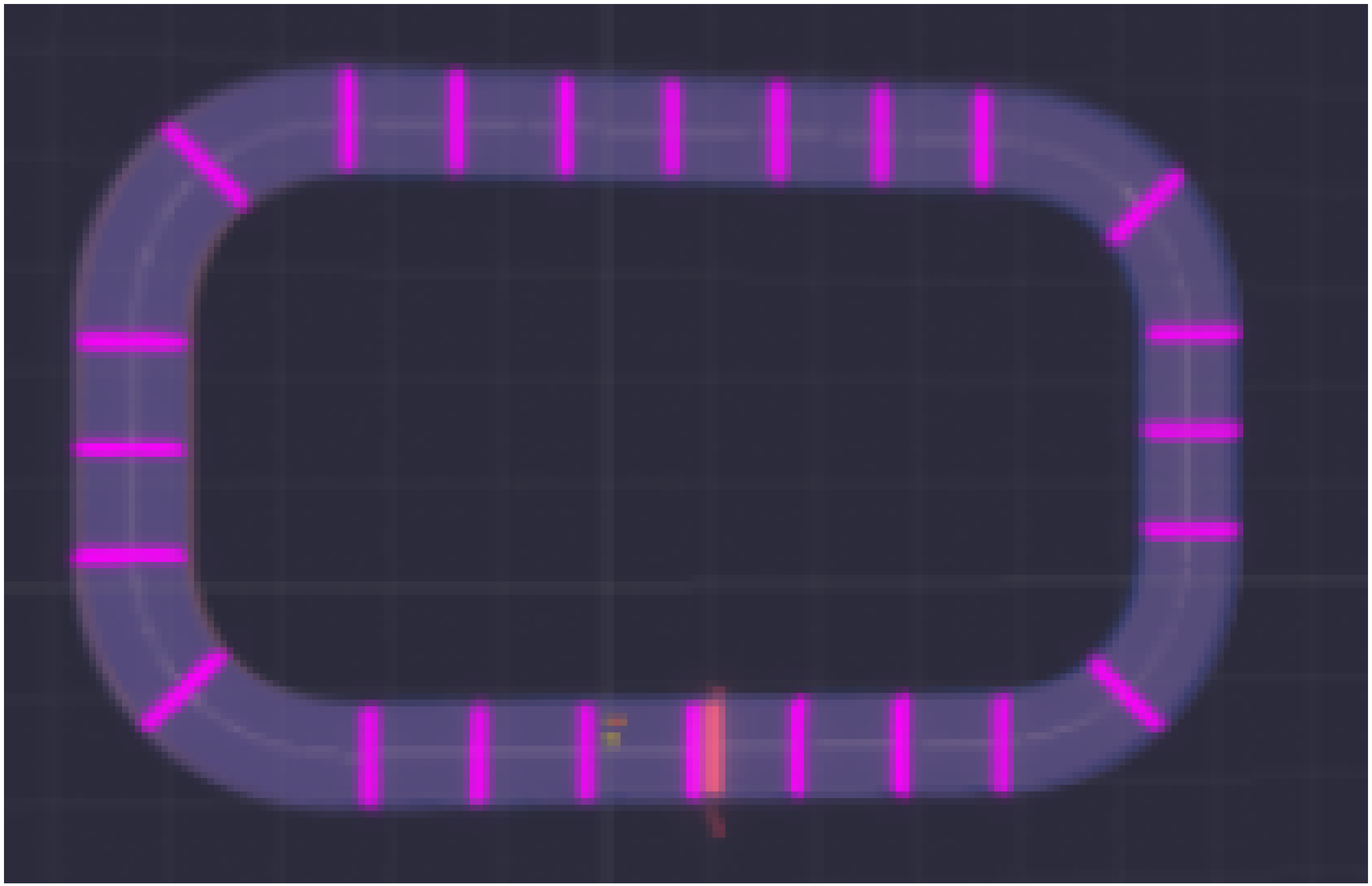
Hierarchical Control for Multi-Agent Autonomous Racing

Rishabh Saumil Thakkar, Aryaman Singh Samyal, David Fridovich-Keil, Zhe Xu, Ufuk Topcu

2022 The University of Texas at Austin

Racing Game

checkpoints



Abstract

Successful human drivers are required to both outperform opponents and adhere to the rules of racing.

Prior approaches in autonomous racing usually over-simplify the latter by only considering collision avoidance. The resulting behavior is an agent that is **not optimal**, or agent that maybe **quick but is unsafe or unfair**.

They develop a **hierarchical control** scheme for optimal long-term plans, and closely adheres to the safety and fairness rules.

Hierarchical control

Divide computation to 2 parts

High-level planner: forms a discrete approximation of the general formulation of the game.

Low-level planner: solves a simplified, continuous state/action dynamic game.

General MA racing game formulation

Compute a player’s distance to **center line** of track. The limit of the distance is track width.

Define **collision avoidance** rules of the game. Each player have indicator function that evaluates if each other players are behind the player.

There are some prior work which tries to stay on the lane and avoid collision if the two is close. They thought it is not enough to make is like real-life racing.

Key rule for fair and safe driving

1. Greater emphasis on collision avoidance for a vehicle that is following another.
   (leading player can make decision without needing to consider an aggressive move that risks collision.)
2. The player **lane switching is limited** while on a straight section.
   (no zig-zagging)

Difficulty in real-time computation

It is possible to make the function for these two rules, but it uses spatially an temporally dependent constraints. It means they are discrete and **cannot be easily differentiated**. Therefore, most optimization algorithm would not apply in real time.

Hierarchical Control Design

approximate long-term optimal choice.

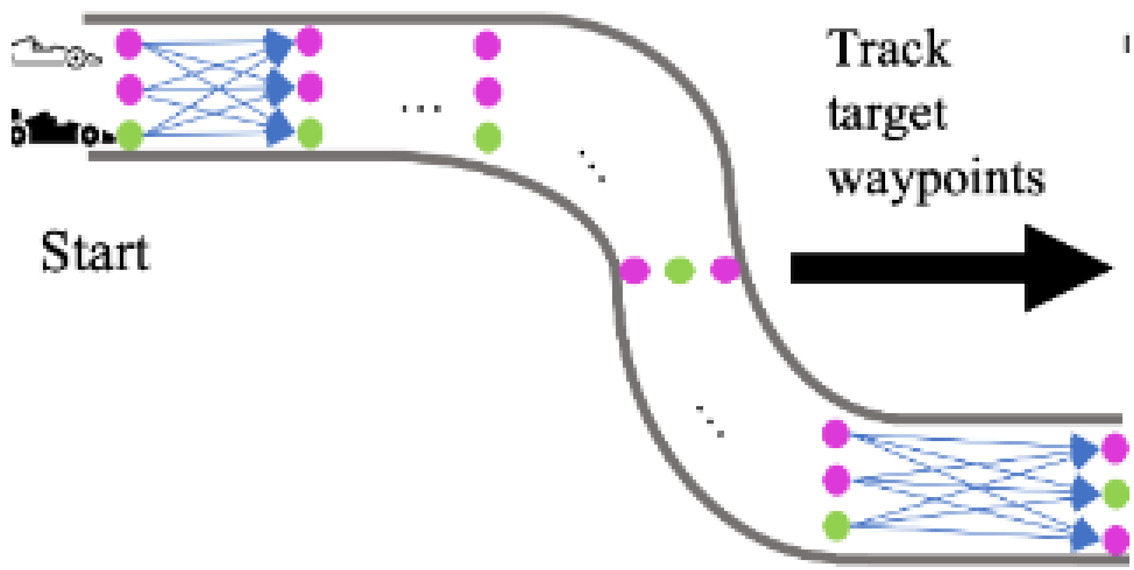
high-level planner: transforms the general formulation into a game with **discrete states and actions** where all of the discrete rules are naturally encoded.

low-level planner :solves a **simplified version** of the racing game and rules.
greater emphasis on tracking a series of way-points,
smaller emphasis on the original game-theoretic objective.

The low-level planner is solved by an optimization method in real-time.

High-level Planner

g Speed	Tire Wear	Waypoints
0~2m/s	0~5%	left side of checkpoint
2~4m/s	5~10%	right side of checkpoint
4~6m/s	10~15%	middle



Have a variable of **lane ID**, and **recent lane changes**.

The player’s actions are defined by Lane ID, next checkpoint, target speed.

Estimate the time from current checkpoint to next checkpoint. If two players reach the same lane at same time, the actions can be dismissed.

The earlier player to reached to the checkpoint will be compute the action first. The player behind others are responsible for collision avoidance.

MCTS is used to solve this.

Low-level Planner

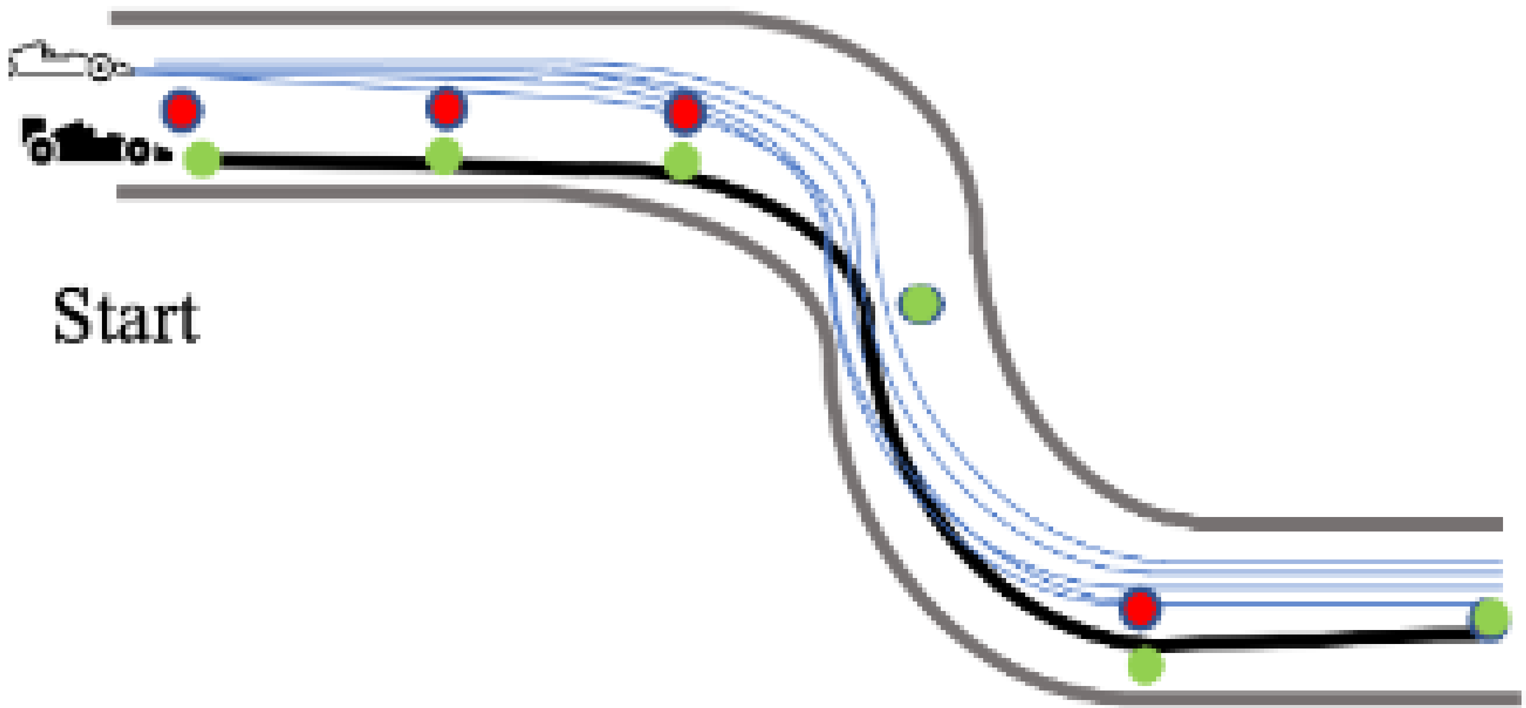
Producing the control inputs.

It get long-term plan from high-level planner, so can formulate a reduced version of the original game.

Objective

1. maximize the difference of checkpoint index between its own and opponents’ at the end of steps.
2. minimize the tracking error. (be close to high-level waypoints)

Trace waypoints from high-level planner



low-level: MA reinforcement

Reward

- Passing as many checkpoints in limited time
  - additional reward for being close to target lane and velocity
- Minimizing the time between two checkpoints

Penalty

- Swerving too frequently on straights
- Going off track or hitting a wall
- Colliding with other players

Frequency

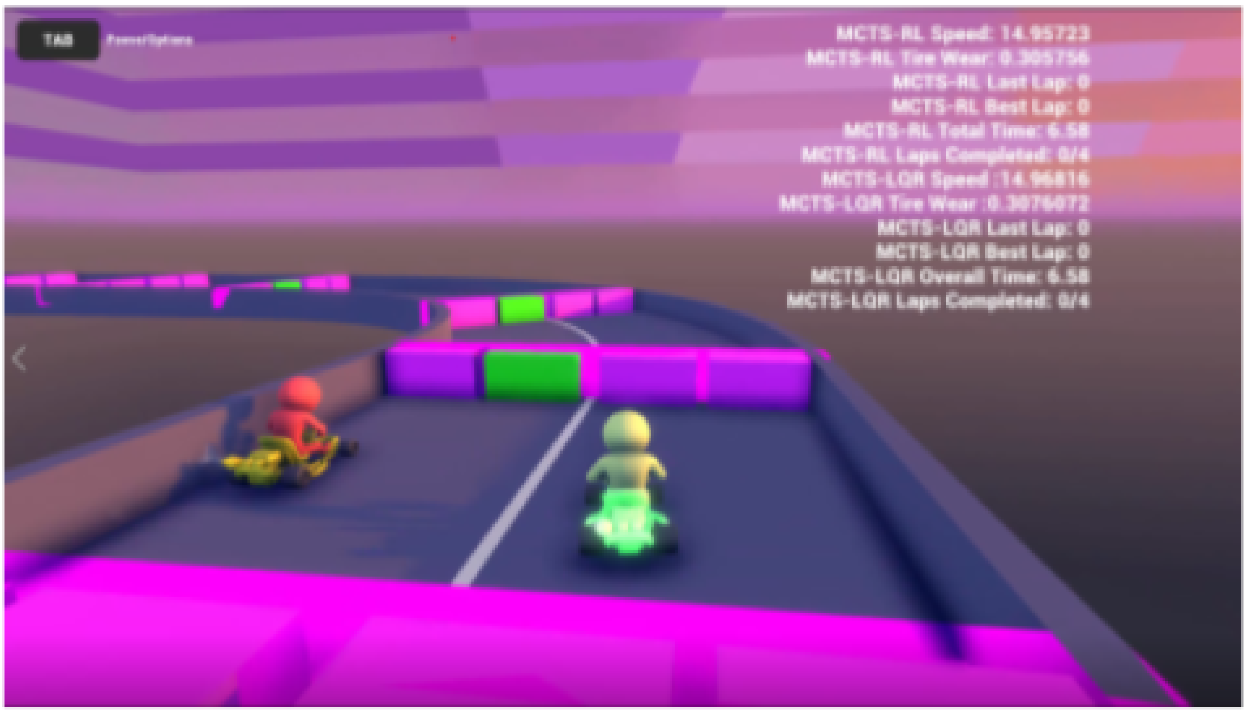
high-level: 1-2 Hz

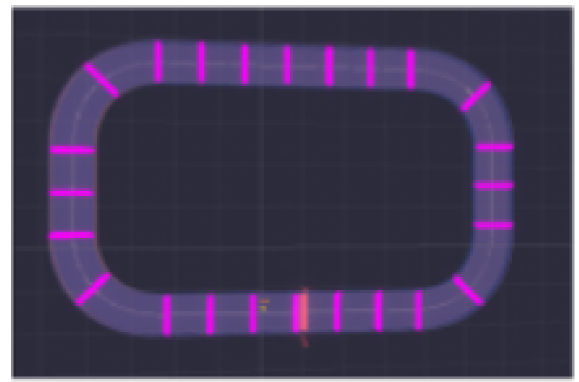
low-level: 25-50 Hz (Real-time)

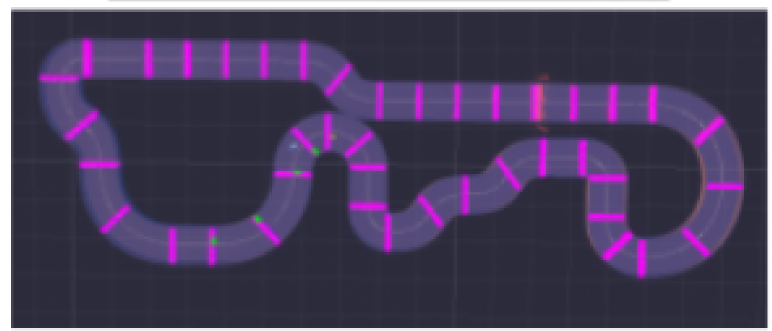
Experiments

Race environment in Unity.

Simple oval track, Complex track
1 vs 1 with prior MA racing agents







Results

The proposed hierarchical variants outperformed their respective baselines.

youtube

- race win rate
- safety score (calculated from collisions)

Conclusion

The hierarchical controller outperformed other MA autonomous racing.

It can compute long-term trajectories real-time.

Their driving was resemble to expert human drivers.

It can be extended to other MA systems.