Iterative Best Response of Zero-Sum Racing Game

ME-429 Final Presentation

Clément Suttor, Yo-Shiun Cheng

May 28, 2025

Motivation

- Problem: self-driving cars must share road with one another and with human drivers. When two cars race down the same track, they need to go fast and avoid crashes.
- Why does it matter: prevent collisions, keep traffic flow smoothly, adaptation to real world scenarios.
- Why game theory: each player plans moves by guessing what the other will do next. This "anticipation " avoids head-on surprises.

Problem Statement

We decided to design a game for competing autonomous racing cars:

- **Zero-sum**: $\mathcal{J}_2 = -\mathcal{J}_1$
- **Multi-stage**, because decisions are made for $k = 0, \cdots, N$-1.
- **Dynamic**, since state updates couple decisions over time.
- **Feedback**, as both players re-solve at each k based on the current x_k.

We model each car with a bicycle model in Frenet frame by tracking its longitudinal distance (s), lateral offset (d), heading error (e_ψ), and velocity (v).

Past Work

- Authors in [1] show a nonlinear receding horizon game-theoretic planner for autonomous cars in competitive scenarios.
- Can we obtain competitive behavior with a simplified framework?
- Authors in [2] show that policy optimization converges to NE in ZS LQ games.
- Does it still apply for constrained optimization in a ZS LQ dynamic game?

[1] Mingyu Wang et al. “Game Theoretic Planning for Self-Driving Cars in Competitive Scenarios”.
[2] Kaiqing Zhang, Zhuoran Yang, and Tamer Basar. “Policy optimization provably converges to Nash equilibria in zero-sum linear quadratic games”.

Frenet Coordinate Transformation

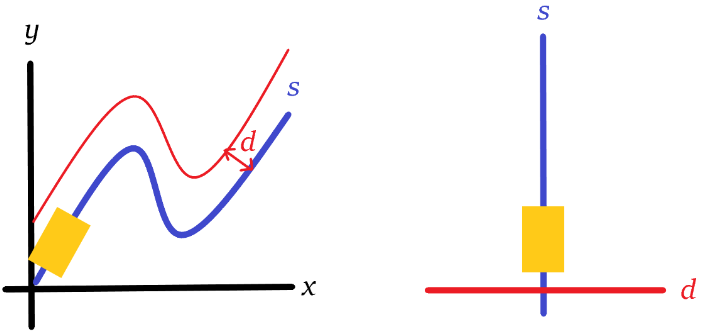


Figure: Visualization of the Frenet Frame Transformation

Simulation Setup

1. Initialize cars on the track
2. At every timestep of the simulation:
   - Compute an approximation of the saddle point strategy using the iterative best-response algorithm.
   - Each player only plays the first action.
   - Repeat until the goal is reached.

Saddle-point Estimation with Iterative Best Response Map

Algorithm 1 Iterative Best–Response at time step k

Require: current state $x_{f,k}$

Initialize

$$U_2^{(0)} \leftarrow \text{zero}, \quad U_1^{(0)} \leftarrow \text{zero}.$$

for $i = 1, \ldots, i_{\max}$ **do**

  Player 1 (minimizer) update:

$$U_1^{(i)} = \arg\min_{U_1} J_1\big(U_1, U_2^{(i-1)}\big) \quad \text{s.t. dynamics \& constraints}$$

  Warm-start QP solver with $U_1^{(i-1)}$

  Player 2 (maximizer) update:

$$U_2^{(i)} = \arg\max_{U_2} J_1\big(U_1^{(i)}, U_2\big) = \arg\min_{U_2} -J_1\big(U_1^{(i)}, U_2\big) \quad \text{s.t. dynamics \& constraints}$$

  Warm-start QP solver with $U_2^{(i-1)}$

  if $\|U_1^{(i)} - U_1^{(i-1)}\|_\infty < \epsilon$ **and** $\|U_2^{(i)} - U_2^{(i-1)}\|_\infty < \epsilon$ **then**

    break

  end if

end for

Return $U_1^* = U_1^{(i)}, \ U_2^* = U_2^{(i)}$

Quadratic Program Formulation

$$\min_{u_1 \in \mathcal{U}_1} \mathcal{J}_1(x_1, u_1) = \sum_{k=0}^{N-1} J_k(x_1^k, u_1^k) + J_{\text{terminal}}(x_1^N)$$

with stage cost

$$J_k = J_{\text{goal}}(s_1, s_2) \; + \; J_{\text{opt}}(d_1, d_2) \; + \; J_{\text{next}}(s_1, s_2) \; + \; J_u(u_1, u_2).$$

Subject to dynamics constraints:

$$x_{k+1} = Ax_1^k + Bu_1^k, \quad x_1^0 = x_1(0)$$

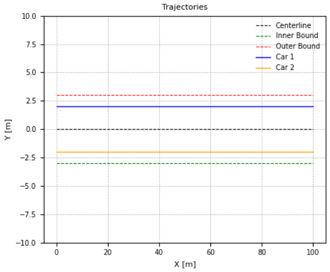
State and Input Constraints:

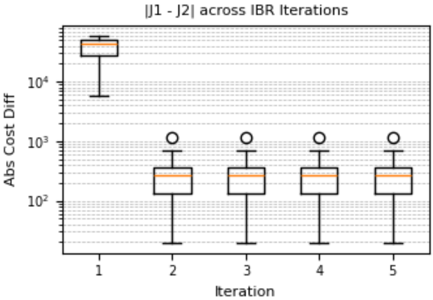
$$\begin{aligned} 0 \leq v_1^k \leq v_{\max}, \\ -a_{\max} \leq a_1^k \leq a_{\max}, \\ -\delta_{\max} \leq \delta_1^k \leq \delta_{\max}, \end{aligned}$$

Track Boundary Constraint:

$$-\frac{\text{track\_width}}{2} \leq d_k \leq \frac{\text{track\_width}}{2}$$

Simulation Results





- Optimal behavior for the straight line track
- Mean and median cost differences are 296.9 and 269.4 once converged
- Given that the average absolute cost is on the order of ~ 3000, this implies a relative cost gap $|\bar{V} - \underline{V}|$ of less than 10%.

Take-Away Messages

- The controller computes the expected optimal actions for a straight line track.
- The best-response algorithm converges to a good approximation of a saddle-point equilibrium.
- The QPs need to be made more robust, dynamics need to be simplified/revised and other game-theory approaches could be implemented using the same framework e.g. a leader-follower zero-sum game approach.

Backup slide

Continuous-time bicycle dynamics in the Frenet frame:

$$\dot{s} = \frac{v\cos(e_\psi)}{1 - \kappa_r(s)\,d}, \tag{1}$$

$$\dot{d} = v\sin(e_\psi), \tag{2}$$

$$\dot{e}_\psi = \frac{v}{L}\tan\delta \;-\; \frac{\kappa_r(s)\,v\cos(e_\psi)}{1 - \kappa_r(s)\,d}, \tag{3}$$

$$\dot{v} = a. \tag{4}$$

Linearization:

$$x_{k+1} \approx A\,x_k \;+\; B\,u_k,$$

where

$$A = \frac{\partial g}{\partial x}\bigg|_{(x_0,u_0)}, \quad B = \frac{\partial g}{\partial u}\bigg|_{(x_0,u_0)}.$$

Backup Slide

We split the per-stage cost into four terms:

$$J_{\mathrm{goal}}(s_1, s_2) = \alpha_{\mathrm{goal}} \left(s_{\mathrm{goal}} - s_1\right)^2 - \alpha_{\mathrm{goal}} \left(s_{\mathrm{goal}} - s_2\right)^2, \tag{5}$$

$$J_{\mathrm{opt}}(d_1, d_2) = \alpha_{\mathrm{opt}} \left(d_{\mathrm{opt}_1} - d_1\right)^2 - \alpha_{\mathrm{opt}} \left(d_{\mathrm{opt}_2} - d_2\right)^2, \tag{6}$$

$$J_{\mathrm{next}}(s_1, s_2) = \alpha_{\mathrm{next}} \left(s_{next} - s_1\right)^2 - \alpha_{\mathrm{next}} \left(s_{next} - s_2\right)^2, \tag{7}$$

$$J_u(u_1, u_2) = \alpha_u u_1^2 + \alpha_u u_2^2, \tag{8}$$