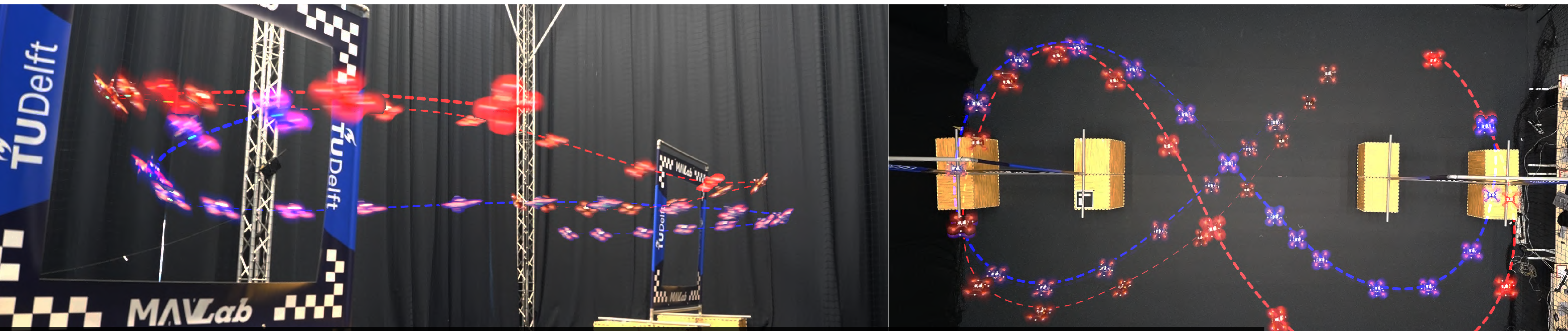
Interaction-aware autonomous drone racing

Andrei-Carlo Papuc, Lasse Peters, Sihao Sun, Javier Alonso Mora

video, references



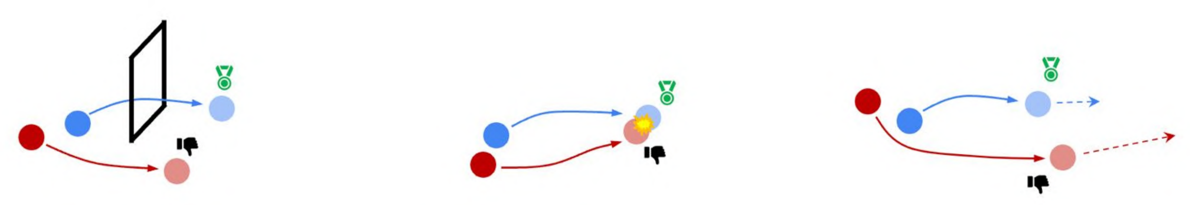
Motivation

While much of the recent research in autonomous racing has focused on optimizing single-agent performance [3, 11], such as minimizing lap times, real-world racing scenarios often involve multiple competitors, each with their own strategies and goals. This creates a dynamic, multi-agent environment where decision-making is influenced by the actions of other participants.

Main contribution

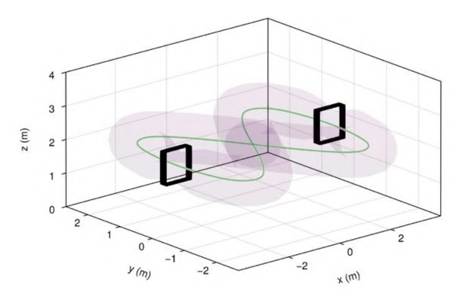
We develop interaction-aware decision-making strategies for autonomous drone racing, leveraging game-theoretic methods to enhance competitive performance.

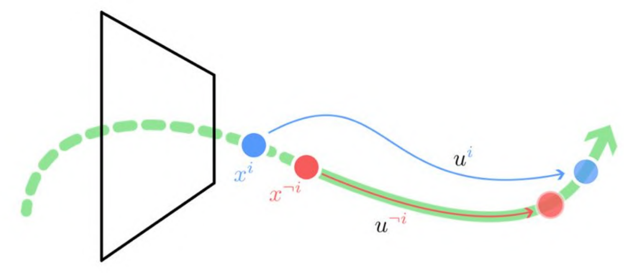
Methodology



Racing Rules

1. Players shall pass through all the gates and not deviate from the track more than 1.5 m
2. The attacker (player behind) is responsible for collision avoidance
3. Players shall adhere to the maximum speed requirements associated with their roles
4. The winner is determined based on time spent as a defender





Racing Assumptions

1. Focus is on planning and control, not on perception
2. Full global knowledge of the race track
3. Full knowledge of own and opponents' states
4. No communication between players

$$J^i(\boldsymbol{x}_0^i, \boldsymbol{x}_0^{-i}, \boldsymbol{u}^i, \boldsymbol{u}^{-i}) = \sum_{k=0}^{K-1} \|e^c(\theta_k^i)\|^2_{q_c(\boldsymbol{p}^d(\theta_k^i))} + \|e^l(\theta_k^i)\|^2_{q_l} \quad (1a)$$

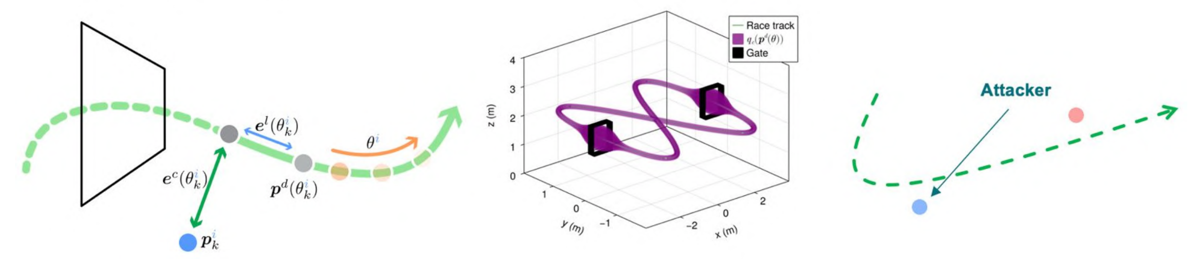
$$+\, \mu(\theta_k^{-i} - \theta_k^i) \quad (1b)$$

$$+\, \mathbb{1}\{\text{attacker} = i\}\, q_{\text{col}}\ \text{col}(\boldsymbol{p}_k^i, \boldsymbol{p}_k^{-i}) \quad (1c)$$

Progress

$$\boldsymbol{x}^i = [\boldsymbol{p}\ \boldsymbol{v}\ \boldsymbol{a}\ \theta\ v_\theta]^T$$

$$\boldsymbol{u}^i = [\boldsymbol{j}\ \Delta v_\theta]^T$$



Objective Components

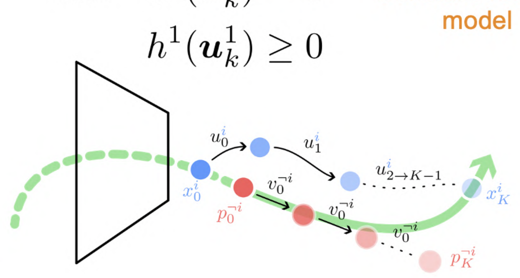
Model Predictive Control (MPC)

$$\boldsymbol{u}^{1*} \in \arg\min_{\boldsymbol{u}^1} \sum_{k=0}^{K-1} J^1(\boldsymbol{x}_0^1, \boldsymbol{u}_k^1, \boldsymbol{p}_k^2)$$

$$\boxed{\boldsymbol{p}_k^2 = \boldsymbol{v}_0^2 k \Delta t}$$

Constant velocity model

$$\text{s.t.}\quad c^1(\boldsymbol{u}_k^1) = 0, \quad h^1(\boldsymbol{u}_k^1) \geq 0$$



Goal

Address the limitations of existing MPC-based approaches, which often model opponents as dynamic obstacles without accounting for their strategic behavior.

Model Predictive Game (MPG)

$$\boldsymbol{u}^{1*} \in \arg\min_{\boldsymbol{u}^1} \sum_{k=0}^{K-1} J^1(\boldsymbol{x}_0^1, \boldsymbol{x}_0^2, \boldsymbol{u}_k^1, \boldsymbol{u}_k^2)$$

$$\text{s.t.}\quad c^1(\boldsymbol{u}_k^1) = 0, \quad h^1(\boldsymbol{u}_k^1) \geq 0$$

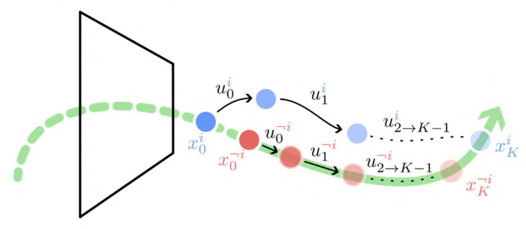
$$\boldsymbol{u}^{2*} \in \arg\min_{\boldsymbol{u}^2} \sum_{k=0}^{K-1} J^2(\boldsymbol{x}_0^1, \boldsymbol{x}_0^2, \boldsymbol{u}_k^1, \boldsymbol{u}_k^2)$$

$$\text{s.t.}\quad c^2(\boldsymbol{u}_k^2) = 0, \quad h^2(\boldsymbol{u}_k^2) \geq 0$$

Each player independently solves the following coupled optimization problem to find a

Nash Equilibrium

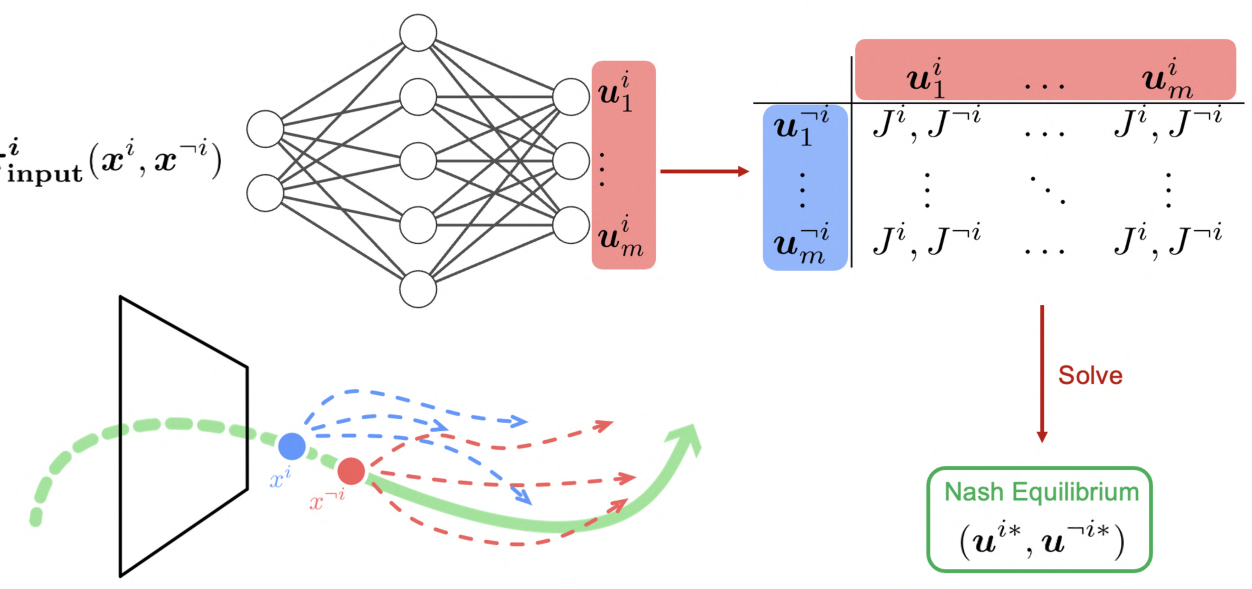
$$(\boldsymbol{u}^{i*}, \boldsymbol{u}^{-i*})$$



To compute a local NE for this problem, we leverage MCPTrajectoryGameSolver.jl [24]. This toolchain is specifically designed for formulating dynamic trajectory games by transcribing the equilibrium conditions into a mixed complementarity problem.

Lifted Model Predictive Game (LMPG)

$$\pi^i_{\text{input}}(\boldsymbol{x}^i, \boldsymbol{x}^{-i})$$



$$(\boldsymbol{u}^{i*}, \boldsymbol{u}^{-i*})$$

We explore lifted game formulations to accelerate online computation, building on the approach proposed by Peters et al. [26], and introduce a specialized training procedure tailored for racing applications.

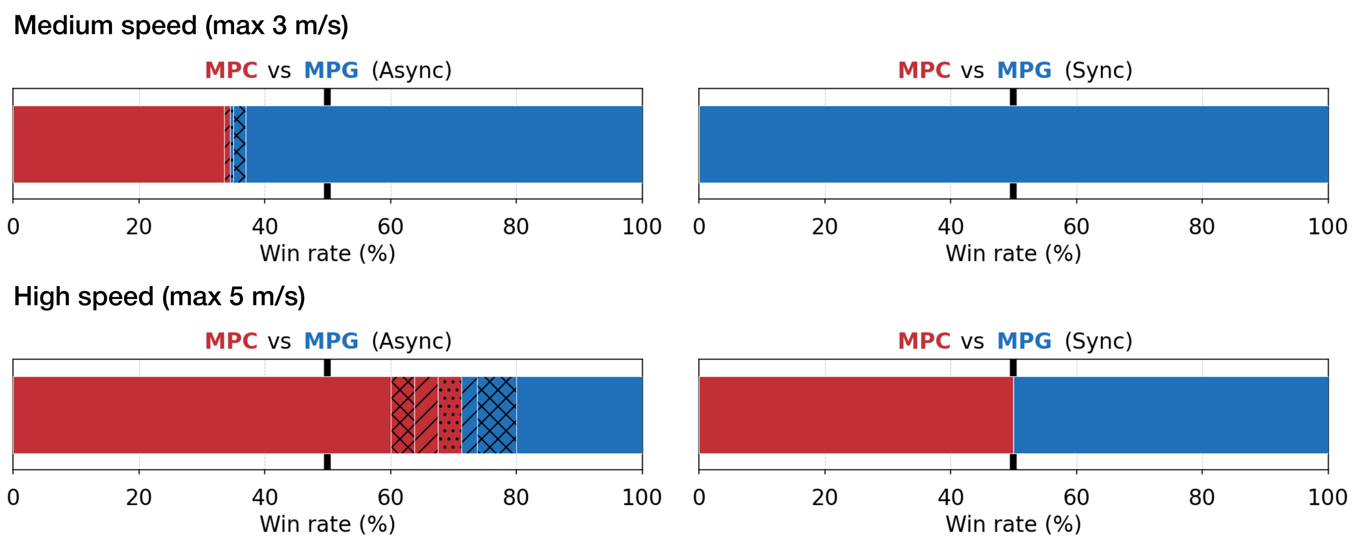
Results

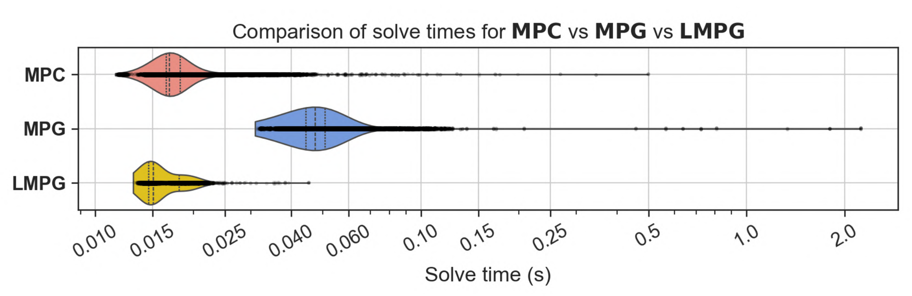
Asynchronous racing

Agents compute and execute strategies at their own independent rates, without waiting for the others

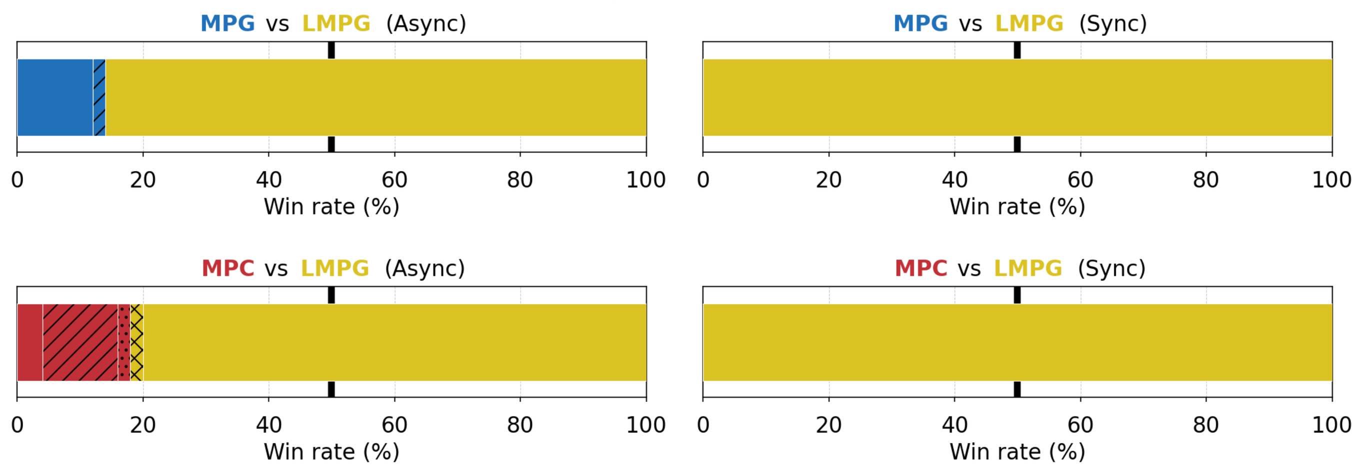
Synchronous racing

Physics waits for agents to compute a strategy and execute them in sync





High solve times of MPG lead to competitive disadvantage at higher speeds



Main findings

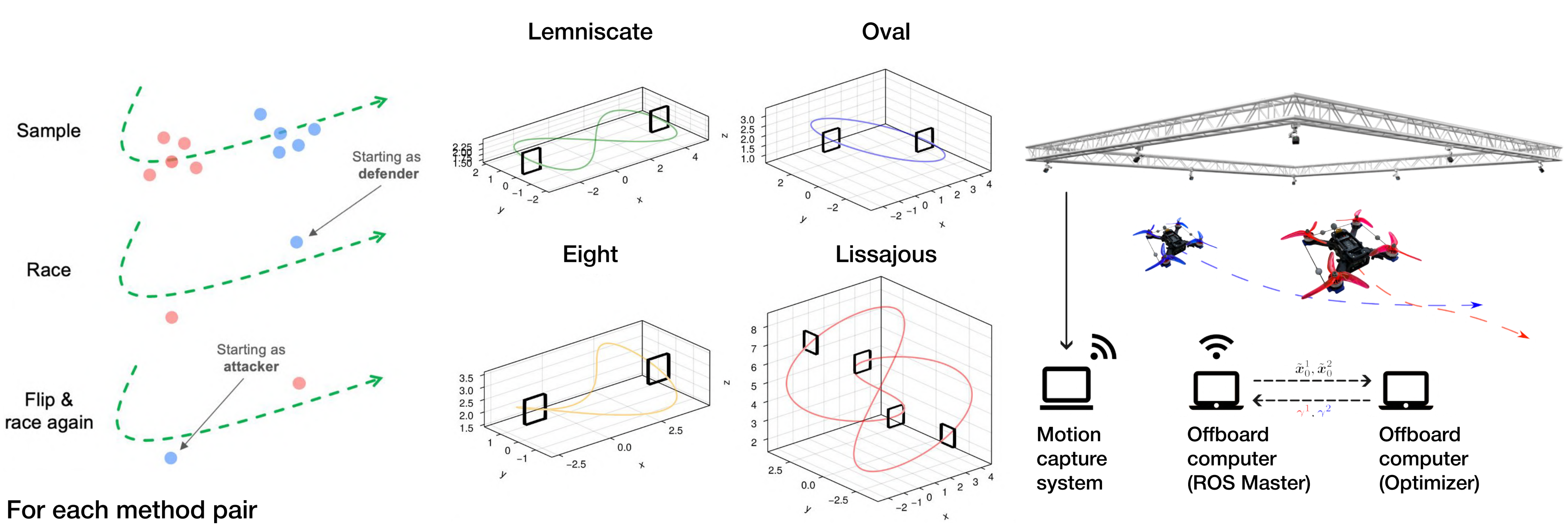
1. MPG consistently maintains a competitive advantage over MPC in synchronous mode. This is evident across different speed configurations where MPG executes strategic overtakes and maintains a dominant racing position

2. Induced delays and decentralized play reduces racing performance, particularly affecting MPG at higher speeds, which suffers from increased computational overhead.

3. By accelerating MPG via learning, we are able to achieve solve times comparable to MPC while maintaining its competitive edge in both synchronous and asynchronous modes.

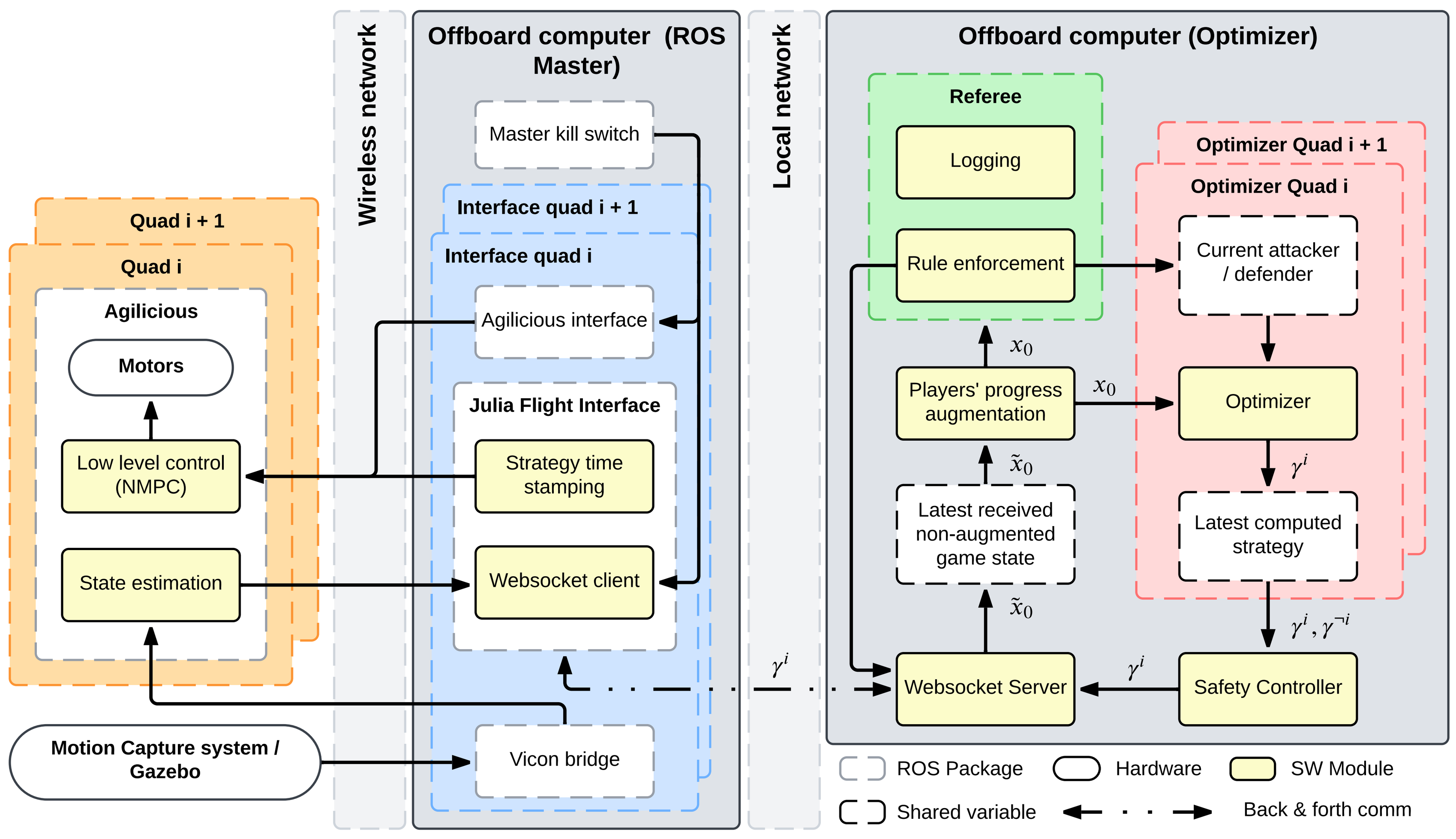
Experimental setup

All vs all tournament on four tracks of varying complexity, number of gates and size



For each method pair

- uniformly sample number of initial states around starting positions
- first race once starting as an attacker, then race again starting as a defender; each method experiences both starting positions.



AUTONOMOUS MULTI-ROBOTS LAB
TU Delft
Cognitive Robotics