Journal in ARES: Controller design for autonomous racing vehicle - VIP 47922

Zhishan Wang

***Abstract*—This journal is mainly used as a record and summary for the 2026 spring semester. Firstly, in this journal, we will mainly review the content and achievements of the previous semester. Then, we will describe the theoretical goals and basic implementation methods for this semester. Afterwards, we will conduct literature reading to get a general understanding of the current situation and background knowledge within the industry. Next, we will carry out research and design for this semester. At the end of this journal, we will record and summarize professional activities. We will also document some details and interesting events of the meetings within our group, as well as write down the key events.**

Date: 2/23/2026

I. THE CONTENT COMPLETED IN THE FALL SEMESTER OF 2025

A. *Brief recap*

Our design objective is to create a controller for high-speed competitive autonomous driving racing cars, ensuring that it meets the requirements of the racing competition and is competitive. We hope that the controller we design will enable our racing cars to operate stably, safely, and efficiently.

Firstly, we carried out the design and description of the physical model. Starting from the most basic bicycle model, we used Newton's second law to describe the basic dynamics structure of the racing car. At the same time, we also imposed restrictions and defined the state of the racing car and the capabilities of the controller. We aimed to design a closed-loop real-time control system to control the errors, so that the racing car could fulfill the path planning instructions and requirements.

We used Matlab to build our own simulation system. In this system, we could adjust the heading error and the design of the basic controller to conduct simulations. We attempted to obtain the changes in the error and thereby analyze the performance of the controller under different race conditions (weather, curves or straight sections, etc.). Eventually, we also obtained the simulation conclusion. Our controller can effectively stabilize the racing car and meets the speed requirements.

B. *State definition*

We define the vehicle state in the global frame as

$$x = [\,X,\ Y,\ \psi,\ v_x,\ v_y,\ r\,]^\top,$$

where (X, Y) is the global position, ψ is yaw angle, v_x and v_y are the longitudinal and lateral velocities in the vehicle frame, and $r = \dot{\psi}$ is the yaw rate. The control input is

$$u = [\,\delta,\ a_x\,]^\top,$$

with δ the front steering angle and a_x the commanded longitudinal acceleration (or a throttle/brake command mapped to acceleration). The reference is generated from the track centerline (or a planned path), typically including desired heading ψ_{ref} and desired speed v_{ref}.

C. *Model design and description (planar bicycle model)*

Starting from Newton's second law, we model the racecar using a planar bicycle model. The kinematics are

$$\dot{X} = v_x \cos\psi - v_y \sin\psi, \qquad \dot{Y} = v_x \sin\psi + v_y \cos\psi, \qquad \dot{\psi} = r.$$

The dynamics are

$$m(\dot{v}_x - r v_y) = F_{x,f} + F_{x,r} - F_{\text{res}}(v_x), \qquad m(\dot{v}_y + r v_x) = F_{y,f} + F_{y,r},$$

$$I_z \dot{r} = \ell_f F_{y,f} - \ell_r F_{y,r},$$

where m is vehicle mass, I_z is yaw inertia, and ℓ_f, ℓ_r are axle distances. Slip angles are

$$\alpha_f = \delta - \tan^{-1}\Big(\frac{v_y + \ell_f r}{v_x}\Big), \qquad \alpha_r = -\tan^{-1}\Big(\frac{v_y - \ell_r r}{v_x}\Big),$$

with $F_{y,f} = f(\alpha_f)$ and $F_{y,r} = f(\alpha_r)$ using a linear/nonlinear tire model. Longitudinal resistance is included:

$$F_{\text{res}}(v_x) = c_r + c_d v_x^2,$$

capturing rolling resistance and aerodynamic drag. The model is discretized for controller-in-the-loop simulation.

D. *MATLAB simulation system (closed-loop pipeline)*

We implemented a MATLAB-based simulation system for repeatable closed-loop testing across track segments (straights/curves) and scenario settings. The pipeline is

track/trajectory reference → error computation → controller → vehicle m

To ensure fair comparisons, we standardized model parameters, time step and integration, actuator constraints (saturation/rate limits), and test scenarios, enabling rapid tuning and regression testing.

Purdue University Electrical and Computer Engineering; VIP ARES; Institute for Control, Optimization and Networks(ICON)

E. Error observation and sensor perception

The controller operates on tracking errors derived from simulated "sensor perception" of vehicle states. We compute

$$e_\psi = \psi_{\text{ref}} - \psi, \qquad e_v = v_{\text{ref}} - v_x,$$

and the lateral error e_y as the signed cross-track distance to the reference centerline. In the baseline setting, these errors (or the states required to compute them) are available directly in simulation. For robustness-oriented designs, observer-based estimation (e.g., ESO in ADRC) is used to infer total disturbances online and reduce reliance on an accurate plant model.

F. Controller description (PID baseline and ADRC family)

Baseline PID/PI. We use a PID controller for lateral steering and a PI controller for longitudinal speed tracking:

$$\delta = K_p e_y + K_i \int e_y \, dt + K_d \dot{e}_y, \qquad a_x = K_{pv} e_v + K_{iv} \int e_v \, dt.$$

This baseline is stable and interpretable, serving as a reference.

ADRC (NLADRC/LADRC). To improve robustness, we implemented ADRC-based controllers which treat unknown dynamics and disturbances as a lumped "total disturbance" and estimate it online via an extended state observer (ESO). A tracking differentiator (TD) is used to smooth the reference, while state error feedback (SEF) generates control action using observer estimates. We explored NLADRC first and then moved toward LADRC to reduce tuning complexity while maintaining disturbance rejection capability.

G. Results description and analysis

We evaluated controllers using a multi-objective performance measure:

$$J = w_t \frac{T_{\text{lap}}}{T_{\text{ref}}} \; + \; w_s \frac{\text{Viol}}{\text{Viol}_{\text{ref}}} \; + \; w_p \frac{\text{RMS}(\Delta u)}{\text{RMS}(\Delta u)_{\text{ref}}}.$$

The PID/PI baseline provides a reliable reference. ADRC-based designs improve the speed–safety trade-off by reducing violation rates while maintaining (or improving) lap time, and typically yield smoother steering due to observer-based disturbance compensation and reference smoothing. Pareto-style comparisons (speed vs. safety) indicate that ADRC shifts the achievable operating region toward faster and safer behavior, forming a strong foundation for future validation beyond pure simulation.

II. THE GENERAL GOALS FOR THE SPRING SEMESTER OF 2026

In Spring 2026, our work will transition from single-vehicle controller benchmarking to a system-level autonomous racing framework in competitive multi-vehicle scenarios. The key motivation is that real racing performance is not only determined by tracking quality on an empty track, but also by interaction-aware decision-making, multi-agent safety constraints, and the robustness of the closed-loop system under uncertainty. Building upon the Fall 2025 closed-loop simulation pipeline (vehicle model + PID baseline + ADRC variants + evaluation metrics), we set the following general goals for this semester.

A. Competitive racing strategies in multi-vehicle scenarios

A primary goal is to develop and evaluate racing strategies that explicitly account for other vehicles on the track. Instead of optimizing a single car's lap time in isolation, we will consider strategic behaviors such as overtaking, defending, lane/line selection, and collision-avoidance maneuvers. The strategy layer will output high-level intents (e.g., target speed profile, desired racing line or lane, passing decision), which will serve as references for the low-level controller. We will define scenario families (e.g., two-car chase, pack racing, corner-entry conflicts) and use consistent performance criteria, such as lap time under interaction, successful pass rate, collision/near-miss rate, and rule/track-boundary compliance, to compare strategies.

B. Multi-agent autonomy and control: controller design and analysis under interaction

The second goal is to extend controller design and analysis to the multi-agent setting. In multi-vehicle racing, each agent's dynamics are coupled through interaction constraints (spacing, collision avoidance, shared track boundaries) and through partial information (limited sensing and prediction uncertainty). We will study both centralized and decentralized control architectures: centralized approaches can coordinate multiple vehicles through shared planning, while decentralized approaches use local observations and limited communication, which is closer to realistic deployment. Our focus is to understand how interaction-aware references and constraints propagate into low-level control performance, and how to guarantee safety (no collision, bounded constraint violation) while maintaining competitive speed. This component will include comparative baselines (simple rule-based interaction policies) and more structured designs (e.g., constraint-based coordination or game-inspired decision policies), evaluated under identical simulation conditions.

C. Optimization of the control system model (parameter tuning and design trade-offs)

We will optimize the control system at the model-and-controller level to improve performance and reduce manual tuning. This includes refining model parameters (e.g., tire/drag terms) that most affect high-speed cornering behavior, as well as systematic tuning of controller parameters (PID gains, ADRC bandwidth/observer gains, constraints on actuation). The optimization objective will remain multi-objective: speed, safety, and smoothness, consistent with our previous evaluation framework. Concretely, we will treat tuning as a search/optimization problem over controller and model parameters, and compare optimized settings against manually tuned baselines. The key output is a more principled and reproducible design process, where gains and model choices are justified through measurable improvements rather than ad-hoc adjustments.

D. Resilience of the closed-loop racing control system (system-level robustness)

A central system-level goal is to evaluate the resilience of the closed-loop controller in the presence of uncertainty and disturbances. Resilience here means the ability to maintain safe, stable, and acceptable performance when assumptions are violated—such as changes in tire-road friction, sensor noise/latency, actuation limits, external disturbances (wind/drag mismatch), or modeling errors. We will design stress-test suites that systematically vary these factors and report degradation curves (performance vs. disturbance level). We will also examine failure modes (e.g., oscillatory steering, loss of lateral stability, aggressive recovery causing violations) and propose mitigation mechanisms, such as adaptive gain scheduling, disturbance estimation improvements, or constraint-aware safety layers.

E. Stability and sensitivity analysis of the control system

Finally, we will conduct stability and sensitivity analyses to understand why the controller succeeds or fails and to provide theoretical support for design decisions. This includes local stability analysis of the tracking dynamics around operating points (e.g., high-speed cornering) and sensitivity of closed-loop performance to parameter variations (controller gains, friction, mass/inertia, time step). When full nonlinear analysis is intractable, we will use linearization-based tools and frequency-domain reasoning to characterize robustness margins, and connect these results to observed simulation behavior. The expected outcome is not only improved performance but also a clearer understanding of the closed-loop system's stability limits and the key parameters that dominate racing behavior.

Overall, these goals shift the project from "controller implementation on a single car" to a more complete autonomous racing system: strategy → perception/error estimation → multi-agent decision/control → robustness and stability verification. This framework will guide our weekly development, experiments, and documentation throughout Spring 2026.

III. LITERATURE REVIEW: THE CORRESPONDING ARTICLE IS LISTED IN THE TABLE AT THE END

A. How this review supports our design

This literature review is organized as a design-driven evidence base rather than a collection of summaries. The goal is to extract reusable modeling patterns, control/decision architectures, and evaluation protocols that can be directly integrated into our autonomous racing stack. Specifically, each paper is mapped to one (or multiple) of our Spring 2026 goals: (i) multi-vehicle racing strategies, (ii) multi-agent autonomy and control, (iii) optimization of the control-system model and tuning, (iv) resilience of the closed-loop racing system, and (v) stability and sensitivity analysis. For every paper, we record the core formulation (system model, objective, constraints, information structure), the key assumptions, and actionable take-away messages that translate into concrete implementation and testing steps in our MATLAB simulation environment.

B. Multi-vehicle racing strategy and interaction modeling

In competitive racing, performance is fundamentally interaction-dependent: overtaking, defending, and line selection are not simply "path tracking problems," but strategic decisions coupled through shared track geometry and collision-avoidance constraints. Literature in this category provides the high-level decision layer that outputs interaction-aware references for the low-level controller (e.g., target line, speed profile, passing intent). Our focus is to identify: (i) what objectives and incentives are used (time gain, risk, rule compliance), (ii) how opponents are modeled (predictive, reactive, game-theoretic, or rule-based), and (iii) how strategies translate into reference trajectories that remain feasible for the controller under actuation limits.

C. Multi-agent autonomy and control (controller design under coupling)

Multi-vehicle scenarios introduce coupling through safety constraints (collision avoidance, minimum spacing), shared resources (track boundaries), and partial observability (limited sensing and prediction uncertainty). Literature in this group provides control architectures that explicitly account for coupling, including centralized coordination, decentralized policies, and distributed optimization. We are particularly interested in how interaction constraints are encoded (hard constraints vs. penalty terms), how information is shared (communication vs. purely local sensing), and how safety is guaranteed while maintaining competitive speed. This group directly supports our objective to move from single-agent control benchmarking to multi-agent closed-loop performance analysis.

D. Optimization-based controller design and model tuning

Optimization methods appear at two levels in autonomous racing: planning (trajectory optimization / MPC) and controller/model tuning (gain selection, parameter identification, trade-off exploration). The literature here provides structured ways to reduce ad-hoc manual tuning by casting controller design as a repeatable optimization problem. We focus on methods that (i) define multi-objective costs aligned with racing trade-offs (speed–safety–smoothness), (ii) handle constraints and computation limits, and (iii) provide systematic parameter tuning procedures for controllers such as PID, ADRC/LADRC, or MPC variants. A key outcome of this group is a reproducible workflow for selecting model and controller parameters with measurable justification.

E. Resilience and robustness of closed-loop racing systems

Autonomous racing is highly sensitive to uncertainty: tire-road friction changes, aerodynamic drag mismatch, sensor noise/delay, and actuator saturation can cause performance degradation or even instability at high speed. Literature in this group focuses on robustness and resilience—defined as the ability of the closed-loop system to remain safe and stable, and to degrade gracefully under disturbances or model mismatch.

We extract disturbance models, stress-testing protocols, and robustness quantification tools (e.g., performance vs. uncertainty curves). This group supports system-level analysis and guides the design of failure-mode tests and mitigation strategies (observer-based compensation, safety layers, adaptive tuning).

F. Stability and sensitivity analysis

Beyond achieving good simulation performance, we need principled explanations of when the system is stable and why failures occur. Literature in stability and sensitivity analysis offers tools to characterize operating envelopes (e.g., high-speed cornering), robustness margins, and parameter sensitivities. We focus on approaches that connect theory to measurable outcomes: linearization around operating points, frequency-domain robustness reasoning, gain/parameter sweeps, and sensitivity derivatives of performance metrics with respect to model/controller parameters. The goal is to produce stability/sensitivity evidence that supports design decisions and informs safe, reliable deployment in more realistic environments.

G. Synthesis (to be completed after reviewing papers)

After reviewing papers across the five groups, we will synthesize cross-paper insights into a concrete design roadmap: what decision/control architecture we adopt, what baselines we implement, what metrics we report, and what stress-test suites we use to validate resilience and stability.

H. Paper-to-category mapping and design inspirations (Spring 2026)

Following the functional-layer decomposition adopted in this journal, we map each reference to one of the five categories: (i) multi-vehicle racing strategy/decision-making, (ii) multi-agent autonomy & control, (iii) optimization-based control, (iv) resilience & robustness, and (v) stability & sensitivity analysis. Rather than summarizing papers independently, we extract what each work contributes to our Spring 2026 goals: multi-vehicle racing strategy, interaction-aware control, optimization-based model/controller refinement, system-level resilience, and stability/sensitivity verification.

I. Synthesis: implications for our Spring 2026 design roadmap

Across the literature, three recurring implications stand out. First, multi-vehicle racing should be treated as a hierarchical autonomy problem: a strategy layer selects interaction-aware intents (line/speed/pass/defend) while low-level controllers track feasible references under actuation limits. Second, the dominant difficulty in multi-agent racing is often the information pattern (local sensing, intermittent communication, adversarial/outlier behavior), motivating explicit graph modeling and hybrid-style estimation/fusion. Third, resilience and stability should be evaluated via stress-test suites (dropout, noise/delay, model mismatch, and adversarial behaviors) and quantified using degradation curves of multi-objective metrics (speed–safety–smoothness), rather than only best-case performance. These insights will guide our implementation baselines and evaluation protocols throughout Spring 2026.

J. Summary of literature insights and how they guide our next steps

The reviewed literature collectively suggests that competitive autonomous racing should be formulated as a *system-level* multi-agent autonomy problem rather than a single-vehicle tracking task. Works on racing strategy (e.g., open-loop planning and mechanism-fusion approaches) motivate a hierarchical design in which a strategy layer selects interaction-aware intents (pass/defend/follow and line/speed choices), while the low-level controller ensures feasibility and smooth execution under actuation limits. Surveys on multi-agent control emphasize that the *information pattern*—local sensing, intermittent communication, and coupling constraints—often dominates performance, which justifies explicitly modeling the agent interaction graph and designing estimation/fusion mechanisms that remain stable under intermittent data.

In addition, the resilience-focused references broaden our evaluation perspective: robustness should be characterized not only by nominal lap time but also by *graceful degradation* under disturbances, model mismatch, measurement dropouts, and adversarial/outlier behaviors. Finally, stability, sensitivity, and verification-oriented works highlight the importance of connecting empirical performance to principled analysis, including operating-envelope characterization, parameter sensitivity studies, and (when applicable) certifiable bounds with transparent conservativeness–computation trade-offs. These insights directly shape our Spring 2026 roadmap: we will implement interpretable multi-vehicle strategy baselines, incorporate intermittent-information and attack/fault stress tests in simulation, and augment controller development with optimization-based tuning and stability/sensitivity verification.

IV. PROGRESS IN 2026 SPRING(REAL-TIME UPDATE)

A. Date: 2/15/2026 Quanser simulation

In the ECE48300 course, we learned the basic methods for software simulation and hardware testing using Quanser. Professor Pare was also willing to provide us with basic equipment for support.

We employed a basic PD controller for the control. Currently, the car can operate stably at a reference speed of 0.3 meters per second.

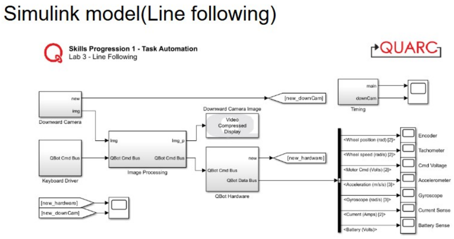

Fig. 1. The Quanser simulation model in Simulink

WHY Quanser?
Our Answer:

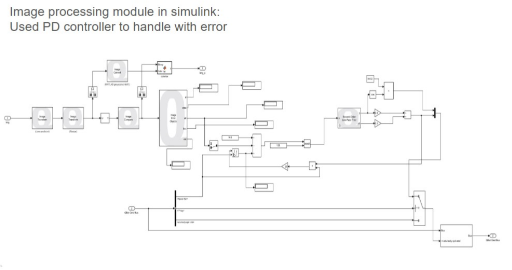

Fig. 2. Image processing model: PD controller

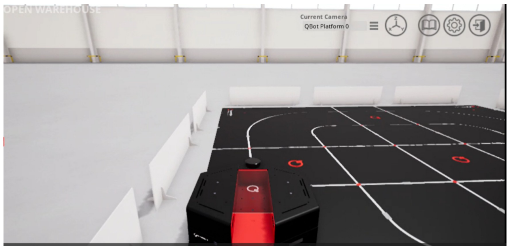

Fig. 3. Virtual simulation in Quanser: Line Following

We chose Quanser because it can maximize the certainty of ”creating and conducting on-site verification” within the project period: Professor Pare is willing to provide Quanser hardware, enabling us not only to conduct virtual simulation but also to carry out HIL/real-machine closed-loop tests; at the same time, Quanser has a mature ecosystem, unified interfaces, and is easy to get started, which can significantly reduce the cost of driver integration and debugging, allowing us to focus on the design and comparison verification of control algorithms and hardware demonstrations. Compared to higher-fidelity large-scale simulation platforms (such as CARLA/Isaac), its drawback is that the ecosystem is relatively closed and the ability to model complex environments is weak, but it better matches our goal of mainly focusing on control verification and hardware presentation.

B. Date: 2/24/Multi vehicles racing strategy and consensus(Math Model)

C. Multi-vehicle competitive racing: model analysis

This subsection analyzes the multi-vehicle racing setting and clarifies how multi-agent interaction enters our controller design. The key perspective is that, unlike single-vehicle tracking, competitive racing couples agents through (i) *shared constraints* (track boundaries and collision avoidance) and (ii) *information constraints* (local sensing and intermittent communication). Therefore, the closed-loop controller of each vehicle should be designed with explicit dependence on opponent/teammate states (or their estimates), and with a clear separation between *interaction-level modeling* and *low-level tracking control*. [**?**]

1) State definition in the Frenet frame: Consider N vehicles indexed by $i \in \{1, 2, \dots, N\}$. For vehicle i, we define the state in the track-aligned Frenet frame as

$$x_i = [\, s_i,\ d_i,\ v_i,\ e_{\psi,i} \,]^\top \in \mathbb{R}^4, \qquad u_i = [\, a_i,\ \delta_i \,]^\top \in \mathbb{R}^2, \tag{1}$$

where s_i is arc-length progress along the track centerline, d_i is lateral offset, v_i is speed, and $e_{\psi,i}$ is heading error. The inputs a_i and δ_i denote the (equivalent) longitudinal acceleration and steering angle, respectively. This representation is convenient for multi-vehicle racing because (i) track boundary constraints are naturally expressed as bounds on d_i conditioned on s_i, and (ii) collision-avoidance reasoning can be done directly in the (s, d)-plane.

2) Discrete-time dynamics abstraction: With sampling time $T_s > 0$, we write the discrete-time dynamics in the compact form

$$x_i(k+1) = f\big(x_i(k), u_i(k)\big). \tag{2}$$

At this stage of modeling, $f(\cdot)$ serves as an abstraction that can represent either (i) a simplified kinematic model for planning/strategy, or (ii) a higher-fidelity bicycle/plant model for controller-in-the-loop evaluation. This separation is important: interaction reasoning and safety constraints can often be formulated using a simpler model, while the final closed-loop feasibility must be validated against the true low-level dynamics and actuation limits.

3) Track and actuation constraints: Track boundaries are enforced by the s-dependent lateral corridor:

$$d_{\min}\big(s_i(k)\big) \le d_i(k) \le d_{\max}\big(s_i(k)\big). \tag{3}$$

Input and speed limits are modeled as

$$a_{\min} \le a_i(k) \le a_{\max}, \qquad |\delta_i(k)| \le \delta_{\max}, \tag{4}$$

$$0 \le v_i(k) \le v_{\max}. \tag{5}$$

These constraints play two distinct roles in multi-vehicle racing: (i) they define the *feasible set* for each agent individually, and (ii) they interact with collision-avoidance constraints, creating coupling-induced infeasibility when traffic is dense. Hence, multi-vehicle performance depends on both strategic decisions and low-level tracking capability.

4) Multi-vehicle interaction: neighbor sets and safety sets: We stack the multi-vehicle state as

$$X(k) = [\, x_1(k)^\top,\ x_2(k)^\top,\ \dots,\ x_N(k)^\top \,]^\top \in \mathbb{R}^{4N}. \tag{6}$$

For each vehicle i, the relevant-neighbor set is defined by a local “interaction window” in the Frenet coordinates:

$$\mathcal{N}_i(k) = \Big\{ j \neq i \;\Big|\; |s_j(k) - s_i(k)| \le S_{\text{range}},\ \ |d_j(k) - d_i(k)| \le D_{\text{range}} \Big\}. \tag{7}$$

Collision avoidance is enforced through an inflated occupied set for vehicle j in the (s, d)-plane:

$$\mathcal{O}_j(k) = \left\{ (s, d) \in \mathbb{R}^2 \;\middle|\; \left(\frac{s - s_j(k)}{\Delta s_{\text{safe}}} \right)^2 + \left(\frac{d - d_j(k)}{\Delta d_{\text{safe}}} \right)^2 \le 1 \right\}, \tag{8}$$

and vehicle i must satisfy

$$\big(s_i(k), d_i(k)\big) \notin \bigcup_{j \in \mathcal{N}_i(k)} \mathcal{O}_j(k). \tag{9}$$

This construction makes the coupling explicit: even if each agent satisfies (3)–(5), feasibility additionally depends on the joint configuration of neighbors through (9). Practically, $(\Delta s_{\text{safe}}, \Delta d_{\text{safe}})$ controls conservativeness: increasing them improves safety margin but may reduce overtaking opportunities and speed.

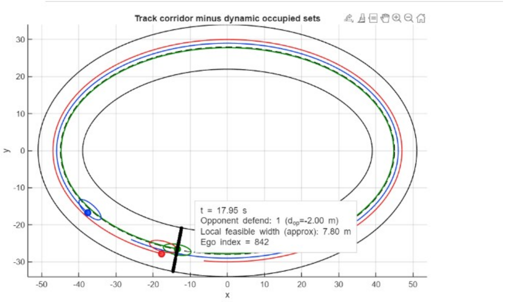

Fig. 4. Multi vehiecel racing

5) Information graph and coordination via consensus: To model information exchange (communication) or local sensing, we introduce a time-varying directed graph

$$\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k)), \qquad \mathcal{V} = \{1, 2, \dots, N\}, \tag{10}$$

where an edge $(j, i) \in \mathcal{E}(k)$ indicates vehicle i can access a coordination variable of vehicle j at time k. The neighbor set is

$$\mathcal{N}_i(k) = \{\, j \in \mathcal{V} \mid (j, i) \in \mathcal{E}(k) \,\} \cup \{i\}. \tag{11}$$

Importantly, racing does *not* require physical states $\{x_i\}$ to agree. Instead, we define a coordination variable $y_i(k) \in \mathbb{R}^p$ to encode team-level intent or shared estimates (e.g., attack/defend/yield score, risk budget, target gap/spacing). The coordination objective is agreement among teammates $i, j \in \mathcal{T} \subseteq \mathcal{V}$:

$$\lim_{k\to\infty} \|y_i(k) - y_j(k)\| = 0, \qquad \forall i, j \in \mathcal{T}. \tag{12}$$

A standard linear consensus update is

$$y_i(k+1) = \sum_{j \in \mathcal{N}_i(k)} w_{ij}(k)\, y_j(k), \tag{13}$$

where $w_{ij}(k) \ge 0$, $\sum_{j\in\mathcal{N}_i(k)} w_{ij}(k) = 1$, and $w_{ij}(k) = 0$ if $j \notin \mathcal{N}_i(k)$. Stacking $Y(k) = [y_1(k)^\top, \dots, y_N(k)^\top]^\top \in \mathbb{R}^{Np}$ gives

$$Y(k+1) = \big(W(k) \otimes I_p\big) Y(k). \tag{14}$$

For time-invariant W with sufficient connectivity, W^k converges to a rank-one limit and yields weighted consensus; in the doubly-stochastic case, the consensus equals the average. A useful convergence-rate proxy is the magnitude of the second-largest eigenvalue $|\lambda_2(W)|$: smaller values imply faster agreement and thus quicker team-level coordination.

When communication is bandwidth-limited, a gossip update can be used, where only one pair exchanges information at each step. For directed graphs where doubly-stochastic weights are difficult, push-sum (ratio consensus) provides an alternative. These variants are particularly relevant in multi-vehicle racing, where teammate communication may be intermittent while opponent information is mostly local and perception-limited.

6) Implications for controller design in multi-vehicle racing: The above model highlights two coupled design questions for Spring 2026:

1) **Interaction-aware references and constraints.** The low-level controller should track references that remain feasible under (3)–(9). Hence, the strategy/planning layer must output interaction-aware intents (target line, speed profile, pass/defend decisions) that respect safety sets and actuation bounds.
2) **Information-limited coordination and resilience.** Because $\mathcal{G}(k)$ is time-varying and may be intermittent, coordination variables y_i should be designed to be robust to partial information (e.g., gossip or push-sum), and the closed-loop system should be evaluated under communication/perception dropouts. This naturally connects multi-agent coordination to resilience and stability/sensitivity analysis.

Overall, the multi-vehicle model in (1)–(14) provides a unified starting point to study how competition and collaboration reshape both the decision layer and the closed-loop controller behavior in racing.

Date: 2/25/2026

How to combine the consensus situation in multi-vehicle competitions with strategy planning and the algorithm of the controller.

D. Connecting the multi-vehicle model with PID and MPC

The multi-vehicle model defined in the Frenet frame introduces coupling primarily through (i) shared constraints (track boundaries and collision avoidance) and (ii) information constraints (local sensing/communication). This section explains how the model can be integrated with two controller families used in our project: a PID baseline and MPC-style optimization-based control.

1) PID integration: interaction enters through references and safety filtering: Given the Frenet state $x_i = [\,s_i, d_i, v_i, e_{\psi,i}\,]^\top$ and inputs $u_i = [\,a_i, \delta_i\,]^\top$, a strategy/planning layer generates feasible references $(d_{i,\text{ref}}, v_{i,\text{ref}})$ that account for neighbors. The tracking errors are defined as

$$e_{d,i}(k) = d_{i,\text{ref}}(k) - d_i(k), \qquad e_{v,i}(k) = v_{i,\text{ref}}(k) - v_i(k),$$

and we optionally use a combined lateral error

$$e_{\ell,i}(k) = e_{d,i}(k) + \alpha_\psi e_{\psi,i}(k).$$

A discrete-time PID/PI baseline is

$$\delta_i(k) = K_p e_{\ell,i}(k) + K_i \sum_{\tau=0}^{k} e_{\ell,i}(\tau) T_s + K_d \frac{e_{\ell,i}(k) - e_{\ell,i}(k-1)}{T_s}, \qquad a_i(k)$$

Since collision avoidance constraints are difficult to enforce directly in PID, we incorporate a lightweight *safety filter* that minimally modifies the PID output:

$$u_i(k) = \arg\min_{u} \|u - u_i^{\text{PID}}(k)\|^2 \quad \text{s.t.} \quad (s_i^+(u), d_i^+(u)) \notin \bigcup_{j \in \mathcal{N}_i(k)} \mathcal{O}_j(k),$$

where $(s_i^+(u), d_i^+(u))$ is a one-step prediction under a simplified model. This preserves the simplicity of PID while enforcing safety and feasibility.

2) MPC integration: direct handling of constraints and multi-objective trade-offs: MPC naturally incorporates the multi-vehicle model and constraints in a receding-horizon optimization. For each vehicle i, over a horizon H, we solve

$$\min_{U_i} \sum_{\tau=0}^{H-1} \Big(\|x_i(k+\tau|k) - x_{i,\mathrm{ref}}(k+\tau)\|_Q^2 + \|u_i(k+\tau|k)\|_R^2 + \|\Delta u_i(k+\tau|k)\|_S^2 \Big)$$

subject to

$$x_i(k+\tau+1|k) = f\big(x_i(k+\tau|k), u_i(k+\tau|k)\big),$$

$$d_{\min}(s_i) \le d_i \le d_{\max}(s_i), \quad u_i \in \mathcal{U}, \quad v_i \in [0, v_{\max}],$$

$$(s_i(k+\tau|k), d_i(k+\tau|k)) \notin \bigcup_{j \in \mathcal{N}_i(k)} \mathcal{O}_j(k+\tau|k).$$

In practice, neighbor states (s_j, d_j) over the horizon are predicted using constant-velocity models, previous-step planned trajectories, or estimator outputs, turning collision avoidance into a time-varying constraint. This yields a decentralized MPC baseline (each agent solves locally). If additional coordination is needed, a distributed MPC variant can exchange predicted trajectories (or coordination variables) over the information graph $\mathcal{G}(k)$.

3) Role of consensus variables: Consensus is not imposed on the physical states x_i, but on coordination variables y_i (e.g., risk budget, attack/defend intent, target spacing). A standard update

$$y_i(k+1) = \sum_{j \in \mathcal{N}_i(k)} w_{ij}(k) y_j(k)$$

produces team-level agreement under suitable connectivity assumptions. The resulting y_i can parameterize the reference generator (changing $(d_{i,\mathrm{ref}}, v_{i,\mathrm{ref}})$) or the MPC cost/constraints (e.g., more conservative safety margins for defensive modes).

V. EVIDENCE

A. Date: 1/20/2026 Model Improvement

The link to the newest model: Model code
In this model, we add more physical details and consider more nonlinear model factors.

B. Date: 1/30/2026: MPC Exploration

We finish the basic MPC model and know its performance.
The link to the model: MPC Model Demo

C. Date: 2/3/2026 Multi car racing strategy

We discussed the basic strategies for multi-car racing as well as the feasible control strategies.
We also discussed the basic direction of information transmission and the different layers.

The link to the papers I read: Literature Review
The link to the slides of multi agant racing demo: Slides
Link to my own notes:
My notes

D. Date:2/10/2026 Quanser Simulation: PD line following QBOT

We utilized the PD controller to achieve the basic following function of the vehicle, and we also conducted tests and tuning of the parameters.
Link to our Quanser simulation video: Quanser demo video
Link to Quanser Model: Quanser Model

E. Date: 2/12/2026 reading papers

Link to papers:
Multi-Agent Systems: A Survey
Multi-Agent Systems for Power Engineering Applications—Part II: Technologies, Standards, and Tools for Building Multi-agent Systems

Multi-agent systems: which research for which applications

My notes:
My notes

F. Date: 2/20/2026 Multi agent strategy

I have summarized many previous papers and viewpoints, and combined them with the context of racing. I hope to discuss this issue from two aspects. We would like to start by considering the situation with human drivers, and then move on to the situation without human drivers.
Link to my notes:
Multi car racing document

G. Date: 3/5/2026 F1 strategy research

VI. PROFESSIONAL DEVELOPMENT ACTIVITY AND SOME EVENTS

Date : 2/25/2026

A. PD Activity 1:

Track: PD required
Activity: Wellcome to VIP
Organizing Unit: VIP ARES mentors team
Date: 1/15/2026

Take away ideas

- 1. Having understood the new requirements for this semester's paperwork compared to the previous ones, I am now starting to prepare the format and structure of the journal.
- 2.Start thinking about the main ideas for this semester. How to improve and reflect on the content from last semester? What kind of team members should be sought this semester, and what kind of resources support are needed for the research?
- 3.We had a discussion with Terry and decided to make the simulation of the virtual environment and the optimization of the control system the main goals for this semester, and we are actively seeking new members.

B. PD Activity 2: ICON student conference

Track: 2.Communication (Required)
Activity: ICON Student conference Poster presentation
Organizing Unit: ICON
Date: 1/29/2026

Take home ideas

- 1. We had exchanges with many scholars and students who specialize in control, optimization and autonomous systems. They provided us with suggestions and opinions on our design and strategies, which greatly helped us to improve and enhance our work.
- 2. Regarding the design of the ADRC controller, many opinions suggest that it is difficult for undergraduate students to fully master it. Therefore, we will attempt to design an MPC or integrate MPC with ADRC as the controller for our system.
- 3.We realize that for the design of the controller, it is not merely responsible for carrying out the task of path planning. The controller itself can also have the function of planning the driving line. Because the controller is closer to the racing car hardware, it has unique advantages in line planning, and this is one of our future directions.

Audience Choice Award - Silver

This award is hereby presented to

Zhishan Wang

in recognition of the second highest number of audience votes during the Poster Sessions at the ***2026 ICON Student Research Conference.***

January 30, 2026
Date

Shaoshuai Mou
Dr. Shaoshuai Mou
Co-Director of ICON
Elmer Bruhn Associate Professor (AAE)

Shreyas Sundaram
Dr. Shreyas Sundaram
Co-Director of ICON
Marie Gordon Professor (ECE)

ICON INSTITUTE FOR CONTROL, OPTIMIZATION AND NETWORKS

PURDUE UNIVERSITY®

Fig. 5. ICON Award

C. PD Activity 3:Multi-agent control reading group (consensus/formation, distributed MPC, safety constraints)

Track: 3.Technical
Activity: Multi-agent control reading group Poster presentation
Organizing Unit: ICON AAE59000MAAC
Date: 2/13/2026

Take away ideas

- 1.Professor Mou presented some of his viewpoints. I hope that the strategy of multi-vehicle collaborative competition and the construction of the model can be taken as one of the themes for this semester. He gave me many excellent ideas.
- 2. We discussed what layers we should consider in a multi-vehicle competition scenario. We decided to divide them into three levels: the target level, the planning level, and the control level. The coordination and unification of these three levels is the key challenge in this task.
- 3.We also considered how to utilize graph theory to represent the information and communication transmission network in real racing scenarios. We need to take into account the perception and detection capabilities of each vehicle regarding its surrounding environment. Only by clearly addressing this issue can we utilize the detection data of each agent regarding the surrounding conditions to make decisions about our own actions.

REFERENCES

Reference	**Category**	**Key inspiration for our Spring 2026 goals**
Piccinotti (2019/2021), *Open-loop planning for Formula 1 race strategy identification*	Strategy	Template for a strategy layer: compact state features + discrete action set + anytime open-loop planning (receding-horizon). Suggests starting from interpretable action candidates (e.g., pass/defend/follow) before full continuous games or RL.
Yang et al. (2022), *Distributed group cooperation with multi-mechanism fusion in an adversarial environment*	Strategy / MA Control	"Mechanism library" idea: represent behaviors as interpretable objective terms (approach/encircle/avoid-collision) and fuse them via weighting or lightweight optimization. Serves as a practical bridge between hand-crafted rules and high-dimensional learning.
Chen & Ren (2019), *On the Control of Multi-Agent Systems: A Survey*	MA Control	Provides standard MAS modules (consensus/formation/distributed estimation/optimization) and architectures (centralized vs. distributed). Helps formalize our multi-vehicle control problem in graph terms and motivates explicit modeling of coupling constraints and information patterns.
van der Hoek & Wooldridge (2008), *Multi-Agent Systems*	MA Control (conceptual)	Clarifies cognitive vs. strategic-structure models. Supports our choice of a "strategic-structure-first" view for racing (capabilities, constraints, and interaction), while using epistemic framing to describe partial/incomplete information.
Murray (2009), *Optimization-based control*	Optimization	Establishes a unified optimization viewpoint for controller synthesis and tuning. Motivates casting our multi-objective evaluation (speed–safety–smoothness) and constraints (track boundaries, actuation limits, collision avoidance) into a repeatable optimization workflow.
Pirani et al. (2023), *Graph-theoretic approaches for analyzing the resilience of distributed control systems: A tutorial and survey*	Resilience / Stability	Connects resilience to graph structure and adversary models. Motivates system-level resilience testing beyond additive disturbances: intermittent neighbor information, adversarial/outlier behaviors, and performance-degradation curves as a function of information/attack strength.
Qiu et al. (2020), *Resilient model-free adaptive control for cyber-physical systems against jamming attack*	Resilience	Provides an engineering pattern for packet-loss/DoS: predictor–buffer compensation during measurement dropouts. Motivates adding Bernoulli/burst dropout models in simulation and comparing "hold-last" vs. predictive compensation in terms of lap time, violations, and control smoothness.
Li et al. (2016), *On*	MA	Treats intermittent