Threading the Needle - Overtaking Framework for Multi-Agent Autonomous Racing

Suresh Babu, Varundev and Behl, Madhur

Abstract

Multi-agent autonomous racing still remains a largely unsolved research challenge. The high-speed and close proximity situations that arise in multi-agent autonomous racing present an ideal condition to design algorithms which trade off aggressive overtaking maneuvers and minimize the risk of collision with the opponent. In this paper we study a two vehicle autonomous racing setup and present AutoPass - a novel framework for overtaking in a multi-agent setting. AutoPass uses the structure of an automaton to break down the complex task of overtaking into sub-maneuvers that balance overtaking likelihood and risk with safety of the ego vehicle. We present real world implementation of 1/10 scale autonomous racing cars to demonstrate the effectiveness of AutoPass for the overtaking task. Our results indicate that the overtake success ratio for the AutoPass framework is 0.395 or 23 times more likely, compared to a purely reactive system at 0.017, while traditional ROS based path planners (depending on the navigation plugin used) are placed between 0.115 to 0.286.

Introduction

Demonstrating high-speed autonomous racing can be considered a grand challenge for multi-agent robotics, and for autonomous vehicles, and making progress in this arena has the potential to enable breakthroughs in agile and safe autonomy. To succeed at autonomous racing, an autonomous vehicle is required to perform both precise steering and throttle maneuvers in a physically-complex, uncertain environment, and by executing a series of high-frequency decisions. Autonomous racing is also slowly becoming a motorsport featuring head-to-head battle of algorithms. Roborace [1] is the Formula E's sister series, which will feature fully autonomous race cars in the near future. Autonomous racing competitions, such as F1/10 racing [2, 3], Autonomous Formula SAE, and Indy Autonomous Challenge are, both figuratively and literally, getting a lot of traction and becoming proving grounds for testing perception, planing, and control algorithms at high speeds and at the limits of controls.

Most past research in autonomous racing has focused on a single-agent time-trial style of racing, i.e, a single autonomous racecar completes a lap in the shortest amount of time. Time-trial poses a number of challenges in terms of dynamic modeling, on-board perception, localization and mapping, trajectory generation and optimal control. Much less attention has been devoted to the multi-agent style of racing that we address in this paper. In addition to the aforementioned challenges, multi-agent autonomous racing also requires inferring the states of other agents, and opportunistic passing while avoiding collisions. Multi-agent autonomous racing provides the opportunity for testing ground for developing and testing more widely applicable non-cooperative multi-robot planning strategies.

In this paper, we examine a two-agent autonomous racing setup to develop effective strategies for overtaking involving autonomous agents that know each other's goals and constraints. The contributions of the paper are:

1. We present AutoPass - an automaton-based framework for high-speed overtaking in a multi-agent autonomous racing setting. The AutoPass framework distills the overtaking maneuver into canonical sub-maneuvers such as approach, overtake trajectory synthesis, passing, and merge in front of the opponent.
2. We present an energy management system model that accounts for boost energy depletion and recovery during the race - a feature common in many motorsports [4].
3. We demonstrate the effectiveness of our approach and its ability to overtake safely on real F1/10 (one-tenth

scale) autonomous racing testbed [2] as well as on the ROS F1/10 autonomous racing simulator [3].

Our control architecture is modular and can fit into the perception, planning, and control stack of any autonomous racecar.

This paper is organized as follows: In Section 2, we present a detailed review of existing literature relevant to the problem of multi-agent autonomous racing. We then provide an overview of the Boost Energy system model in Section 3. The overtaking problem formulation is described in detail in Section 4, followed by the description of our novel AutoPass framework in Section 5. In Section 6, we evaluate the effectiveness of our proposed AutoPass framework in terms of successful overtakes for both simulated and real F1/10 autonomous racing testbed. Finally, we conclude the paper with a summary of the results and a brief discussion on future work in the conclusion Section 7.

Related Work

Autonomous Overtaking: Research (overtaking has focused mostly on free where the ego vehicle is tasked with s to pass a slower car. There have been on the best way to make this happen. a method for overtaking a car using ar with an emphasis on passenger comfo in [8] demonstrates classical path plar autonomous overtaking. These metho tested for structured autonomous drivi and well-defined passing behaviors, ar autonomous racing. Authors in [9] de overtake strategy from a non-linear m controller, while works demonstrated data-driven approaches to solving auto a simulated environment.

Autonomous Racing: The work done game approach to autonomous racing. graph based trajectory planning for hi limits of control. An example of data- high-speed autonomous racing is dem the purpose high speed autonomous ra overtaking framework, however, we fo described in [15] demonstrates how pu used for high-speed autonomous racing, but it has several drawbacks, such as corner-cutting, which can lead to collisions with the racetrack boundary. The authors in [16] propose using a model predictive controller for a high-speed racing controller that overcomes the corner-cutting problem, but this requires significant off-board computation and is compute and memory intensive. The work shown in [17] demonstrates an online implementation of MPC in an embedded robotic control system that is both fast and reliable for high speed autonomous racing. However, these methods are not capable of autonomous overtaking, except for [16] which can track and avoid dynamic obstacles, but only in a reactive manner.

Boost Energy Recovery and Management: Boost energy is used in real motorsport racing as an aid in overtaking. Examples include the Formula 1 Kinetic Energy Recovery Systems [18] and Motor Generator Units [4], and the Indycar's Push-to-Pass system [19]. There is no related work in incorporating boost energy systems within autonomous racing overtaking approaches. This paper is among the first to address this problem.

Consequentially, in this paper we attempt to address shortcomings of these related efforts by presenting a new framework that uses a high-speed MPC controller that is capable of autonomous overtaking using the F1/10 racecar testbed and also by incorporating a boost energy management system capable of both recovering and using the stored boost energy. The framework is implemented using an automaton that lends itself well to model checking and design by verification methods in the future; such verification is not included in the current work.

Energy Management System

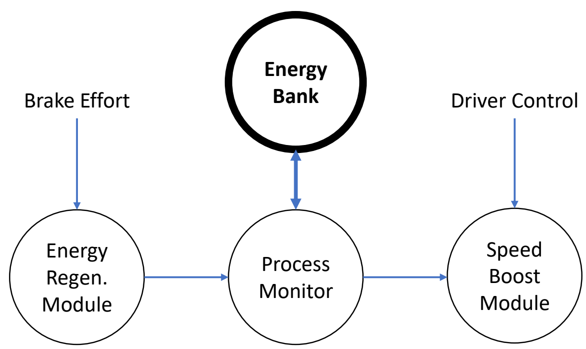


Figure 1: High level architecture of the Energy Management System (EMS).

In motorsport racing, overtaking involves using additional energy (boost) to move past an opponent. Racecars are fitted with dedicated boost energy control systems like the

Variable	Description	Value
Time step	q	$0.1sec$
Energy Bank	T	
Speed drain rate	ΔU	1% of T for $1q$
Max. speed	u_{max}	
Speed boost	u_{boost}	25% of u_{max}
Brake effort	b	0A - 45A
Regeneration range	B	5A - 60A
Regen. capture rate	ΔT	0.1 for $1q$

Table 1: The EMS control variables, with values during experiment. Demand values are between 0 and the stated max; drain rates indicated percentage from T for every q; A is current amperage; v_{max} is continuous max speed.

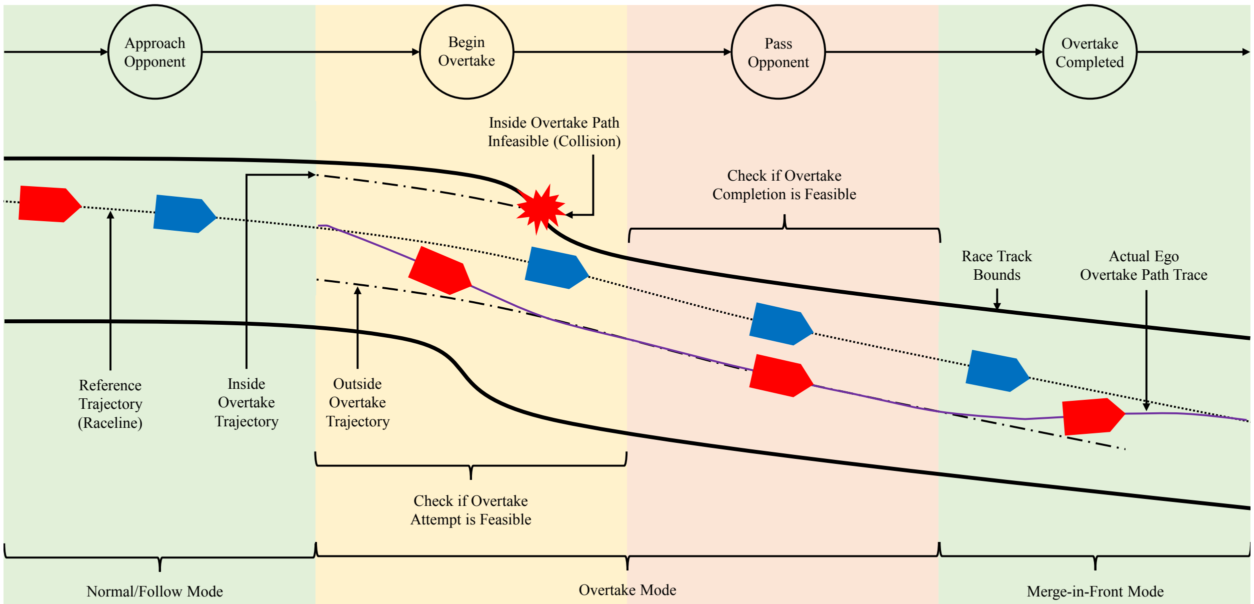


Figure 2: Overtake Stages for a Two Racecar Setup: the ego racecar (a) approaches the leading opponent, (b) chooses a feasible overtake trajectory, (c) passes the opponent, and (d) safely merges in front of the opponent

F1 Motor Generator Unit (MGU) [4] and the Indycar's Push-to-Pass system [19]. In some of the top motorsports, this boost energy has to be recovered by the racecar [4, 18] from its kinetic energy. In a similar manner, we created the Energy Management System (**EMS**) for our F1/10 [2] testbed to recover kinetic energy and provide boost energy to the racecar. The F1/10 testbed used in this paper has an electric drivetrain fitted with a motor controller (VESC [20]) capable of recovering, metering, and storing kinetic energy through regenerative braking. We created a virtual boost control system that allowed the F1/10 racecar to travel at a higher-than-maximum rated speed using the recovered energy for a fixed duration of time. The various parts of the EMS are shown in Figure 1. Energy recovery works by using the back-EMF from the axles to the motor, which is metered and stored in the main traction battery as a virtually separate entity (Energy Bank). When energy boost is necessary, the AEMS provides a dynamically controllable speed boost boost by drawing more power (peak power), which is subtracted from the Energy Bank. Normally the drivetrain is operated under continuous maximum power.

$$
\begin{aligned}
T_q &= T_{q-1} + \Delta T; b \ \epsilon \ B \\
v_q &= v_{q-1} + u_{boost}; T_q = T_{q-1} - \Delta U
\end{aligned}
\tag{1}
$$

The energy regeneration and utilization are governed by the Equation 1 and Table 1 provides a list of variables used by the AEMS. The variable T_q models regenerative braking, while v_q models the dynamic boost energy utilized with proportional energy drain from the EMS.

Problem Formulation

Consider an ego racecar following a global raceline lagging an opponent racecar within its horizon. The objective for the ego racecar is to safely pass the leading opponent. The ego racecar approaches the opponent and generates a feasible overtake trajectory to pass the opponent; the ego racecar estimates its chance of a successful overtake, and if feasible, executes the overtake maneuver in the sequence shown in Figure 2, while utilizing the available boost energy for overtaking as needed. A summary of the four stages of an overtake maneuver is as follows:

- Approach Opponent: The ego racecar approaches the leading opponent and generates multiple overtake trajectories to pass the opponent; the ego racecar estimates its chance of a successful overtake.
- Begin Overtake: The ego racecar chooses a valid overtake trajectory (if one exists) and verifies whether it has enough boost energy to pass the opponent, if so, it starts the overtake maneuver.
- Pass Opponent: The ego racecar passes the opponent and takes the lead while continuing to estimate if the overtake maneuver remains feasible on the current overtake trajectory.
- Complete Overtake: The ego racecar clears enough distance in front of the opponent and attempts to take the lead position on the global raceline trajectory to continue the race. Otherwise the ego racecar abandons the overtake.

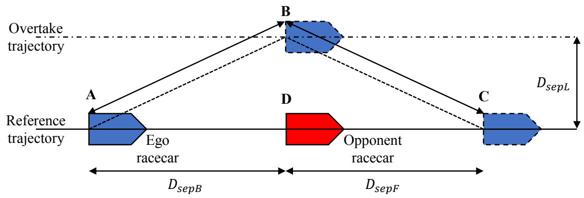


Figure 3: Approximated overtake geometry

Figure 3 provides an approximate geometry of this overtake maneuver. For the purpose of simplification, the ego racecar must not violate (a) the minimum rear separation distance D_{sepB} and minimum frontal separation distance from the opponent (D_{sepF}), and (b) the minimum separation from the fender of the opponent during overtake (D_{sepL}). The ego racecar must take the approximate path **ABC** (x_{tot}) around the opponent, with the point **B** being on an overtake trajectory. Table 2 provides a brief description of notations used in this paper.

Notation	Description
x_{tot}	Total overtake path length
D_{sepL}	Lateral separation between racecars
D_{sepF}	Minimum front separation between racecars
D_{sepB}	Minimum rear separation between racecars
D_{Fmin}	Minimum follow distance
D_{Fmax}	Maximum follow distance

Table 2: List of notations used in AutoPass

The objective of the ego racecar is to safely pass the leading opponent. Assume the ego racecar is travelling at velocity (u_{ego}) and the opponent at (u_{opp}) with a differential velocity of the ego w.r.t. the opponent at (δu). The ego racecar must travel along the path ABC (see Figure 3), which is the total overtake path length (x_{tot}) calculated using Equation 2. Here, we assume D_{sepB} and D_{sepF} are the same. Additionally, in the right triangle **ABD** from Figure 3, **AB** is the hypotenuse, **BD** is the height which is D_{sepL} and the base **AD** is the sum of the initial vehicular separation (D_{sepB}) and the differential progress made by the opponent while the ego travels along the hypotenuse until it reaches the apex at **B**. We calculate the hypotenuse **BC** of the right triangle **BDC** in the similar method. Since D_{sepB} and D_{sepF} are similar, the right triangles **ABD** and **BDC** are the same as long as the opponent's velocity does not change during the overtake maneuver, thus the total overtake path (x_{tot}) is the sum of the hypotenuse AB and AC. Equation 2 explains this mathematically. Equation 2 is formulated in the moving frame of the opponent.

$$
\begin{aligned}
\delta u &= u_{ego} - u_{opp} \\
AB &= (AD^2 + BD^2)^{1/2} \\
&= ((D_{sepB} + \delta u * t)^2 + D_{sepL}^2)^{1/2} \\
BC &= (DC^2 + AD^2)^{1/2} \\
&= ((D_{sepF} + \delta u * t)^2 + D_{sepL}^2)^{1/2} \\
x_{tot} &= AB + BC \\
&= 2 * ((D_{sepB} + \delta u * t)^2 + D_{sepL}^2)^{1/2}
\end{aligned}
\tag{2}
$$

From Equation 2, ($\delta u * t$) is the differential progress made by the ego racecar w.r.t. to the opponent during the time period t. A positive (δu) is necessary for a successful overtake i.e., the velocity of the ego racecar greater than the velocity of the opponent. It is also assumed that both the ego racecar and the opponent have non-zero velocities.

The problem is to find an efficient and feasible way to use the boost energy during the overtake maneuver such that the likelihood of the completion of the maneuver can be computed ahead of time. For this, we compute the minimum value for $u_{ego} + u_{boost}$ (boosted speed) that would satisfy Equation 3. The boosted speed is the sum of the normal speed of the racecar and the temporary increase over the maximum continuous speed provided by the energy management system (EMS). The boosted speed ($u_{ego} + u_{boost}$) is capped at ($u_{max} + u_{boost}$). The energy management system in section 3 and Table 1 provides more information about these variables and how they are controlled by the AutoPass framework.

$$
\begin{aligned}
t &= 2 * \sqrt{\frac{D_{sepL}^2}{(u_{ego} + u_{boost})^2 - u_{opp}^2}} \\
t &<= T
\end{aligned}
\tag{3}
$$

The value t is the boost energy budget (in terms of time) assigned to the overtake maneuver from the Energy Bank (T). This means that the ego racecar has the time target t to reach the end of the overtake path. Figures 2 and 3 along with Table 1 provide more information about the variables.

We have made the following assumptions in this paper: (a) the ego racecar is only tasked with overtaking an opponent and is not expected to defend its position if the opponent initiates an overtake, (b) the ego racecar is capable of accurately estimating the vehicle state of the opponent (incl. position, velocity, etc.), (c) the ego racecar is aware of the opponent's racing strategy, and (d) we currently assume a two racecar setup i.e., one ego and one opponent racecar.

AutoPass Framework

To fully implement the above sequence and to account for scenarios where an overtake may not be possible, current overtake becomes infeasible, or opponent attempts to block an overtake attempt, the ego racecar must continuously track the opponents state and accurately predict the opponent's future state. The entire framework's architecture is shown in Figure 4. The framework works simultaneously to (a) estimate ego and opponent states, (b) make decisions based on an overarching racing strategy, and (c) implement the chosen strategy on the ego racecar's race controller. The three modules are connected to an overtake state machine (a finite state automaton) to precisely execute the correct racing behaviour before, during, and after an overtake attempt. The state machine (AutoPass automaton) is described in detail in the next sections.

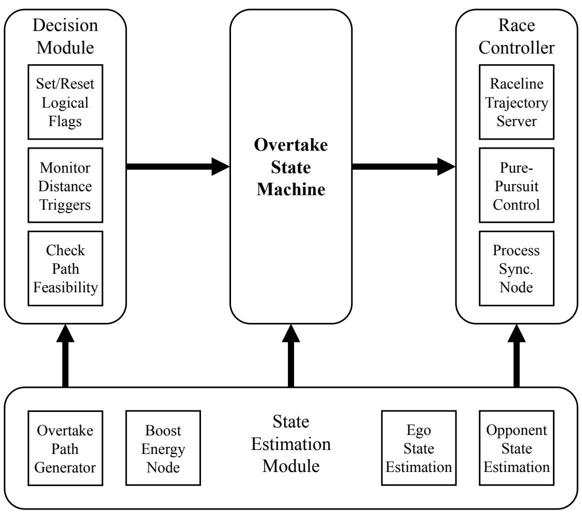


Figure 4: Overtake Control Architecture, showing the various functional modules and their interdependent relationships

State Estimation Module The observable states of the ego racecar and the opponent are estimated using a fast approximate particle filter [21]. These states include position in racetrack and absolute velocities, which helps extrapolating into the future to calculate future states as well. The ego racecar's internal state estimation includes the B.E.M. and the overtake path generation. In our setup, the ego racecar produces several parallel trajectories to the main raceline equally spaced using the largest dimension of the ego and the opponent base footprint. The overtake trajectories are generated by projecting the global raceline by a constant offset from the geometric center of the racetrack. The offsets used in this paper is a multiple of D_{sepL}. To judge the feasibility of the overtake paths, we checked for potential collisions with the racetrack bounds using a simple ROS occupancy grid search of the racetrack.

Decision Module The ego racecar must maintain safe distances from the opponent in addition to the constraints imposed by the navigation stack. These are a set of distance triggers that help in calculating the logical inputs to the overtake state machine:

- D_{Fmin} - the minimum follow distance, which is the closest the ego racecar is allowed to follow the leading opponent without risking collision or violating the ego racecar's constraints.
- D_{Fmax} - the maximum follow distance, which is the farthest the ego racecar is allowed to follow the leading opponent when in the follow state (more info next section) .
- D_{sepF} - the minimum frontal separation, which is the minimum distance the ego racecar must achieve between the front fender of the opponent and the rear fender of the ego during overtake to be allowed to merge in front of the opponent.
- D_{sepB} - the minimum rear separation, which is the absolute minimum distance between the front fender of the racecar and the rear fender of the opponent that the ego racecar must achieve to consider the current overtake abandoned.

Race Controller The ego racecar is controlled by a robust Ackermann-steering adjusted pure-pursuit path planner [22] and a global trajectory server using costmap and occupancy grid layers from the Robot Operating System (R.O.S.). The trajectory server breaks down the given trajectory whether it is the reference raceline or the chosen overtake trajectory path into a set of equidistant waypoints that the pure-pursuit controller can use to navigate around the race track at high speeds. For our current implementation, we chose pure-pursuit over more sophisticated controllers such as the Model Predictive Controller described in [23] because of the ease of hardware implementation and computational simplicity. The pure-pursuit controller used in this paper may not be well suited for a full-scale racecar [24, 25]. Our decision to use the pure-pursuit controller over more sophisticated model-based controllers shown in [24, 25], is because (a) our F1/10 testbed [2] is based on a scaled radio-controlled (RC) car, and (b) our test race-track short and wide enough to reasonably accommodate the path tracking error and limitations of the pure-pursuit planner used. We have made the AutoPass framework modular in order to accommodate different race controllers (see Figure 4) as long as the necessary state information is made available from the race controller to the rest of the AutoPass framework. As can be seen in Figure 4, the right hand side is the race controller which comprises the following nodes that (a) monitor the pose of the ego racecar on the global raceline, (b) provide the synchronization signal for the other nodes in the AutoPass framework, and (c) provide steering and throttle commands to the vehicle controller. The scope of our work is focused on designing the architecture and algorithm for the overtake maneuver - which itself is decoupled from the low-level controller i.e. a different race controller such as MPC could be used in place of pure-pursuit. The race controller provides the timing signal necessary to synchronize all the processes in the AutoPass framework. The race controller also traction effort to overcome wheel-sli

We chose an automaton to implemer framework as it allowed us to easily assumptions about how the overtake structured, and in the future, we will model check our framework.

The overtake state machine is define $(\mathcal{P}, \mathcal{M}, \mathcal{I}, \delta)$, with P inputs, M stat and $\delta = P \times M$ is the set of transiti The state estimation module provide machine.

Figure 5 shows the AutoPass automa conditions governing state transition

Inputs

Relative Path Progress (y_{gap})

The distance y_{gap} is the absolute pat racecars. It is the product of differen of the corresponding racecar on the r resolution (actual distance between successive waypoints). We calculate the path distance instead of the minimum distance between racecars due to the geometric complexities of the racetrack.

$$y_{gap} = arg\min(y_{ego} - w_i) - arg\min(y_{opp} - w_i) \quad (4)$$

In Equation 4, w_i is a waypoint along the global raceline (or, reference trajectory) and the function $arg\min$ is calculating the index of the waypoint closest to the racecar's $base\_link$ (geometric center of racecar control - usually at the center of the rear axle).

Overtake Flag (OTF) The overtake flag (OTF) is a boolean flag that checks if an overtake maneuver is feasible by comparing the resources requested for the overtake and the resources available to the race controller. It takes into account the feasibility of the list of overtake trajectories generated and chooses the best trajectory to execute the overtake using minimum amount of boost energy available.

Position Match Flag (PMF) This input is a boolean flag that is set when the relative path progress is zero for the first time during the attempted overtake maneuver to separate the trigger conditions between the overtake attempt and merge front maneuver.

Safe Fall Back Flag (SFB) When an overtake attempt fails, the ego racecar is designed to transition to the the follow mode behind the opponent racecar, and the safe fallback flag is a boolean flag that is set $True$ when all safety distance thresholds have been met.

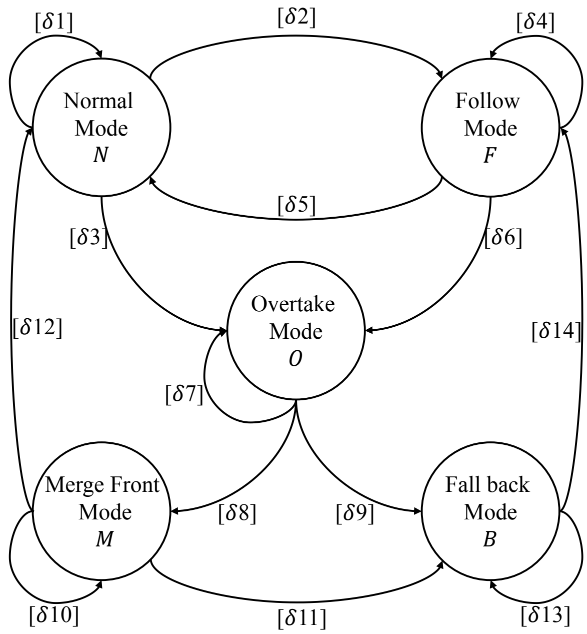


Figure 5: The AutoPass automaton, showing the different states involved in an overtake attempt and the transition (guard) conditions between the states

States

Normal Mode In this state, the ego racecar executes normal racing behavior on the raceline while within the same mode [$\delta 1$]. Two transitions are possible from this state to (a) Follow Mode and (b) Overtake Mode, as defined by guard conditions [$\delta 2$] and [$\delta 3$]

$$\begin{aligned} &\delta 1 : |y_{gap}| < D_{Fmin} \\ &\delta 2 : |y_{gap}| \epsilon [D_{Fmin}, D_{Fmax}] \& !OTF \\ &\delta 3 : |y_{gap}| \epsilon [D_{Fmin}, D_{Fmax}] \& OTF \end{aligned} \quad (5)$$

Follow Mode In follow mode, the ego racecar maintains a safe following distance from the opponent racecar while on the raceline [$\delta 4$] with two possible transitions to (a) Normal Mode and (b) Overtake Mode, as defined by guard conditions [$\delta 5$] and [$\delta 6$]

$$\begin{aligned} &\delta 4 : |y_{gap}| \epsilon [D_{Fmin}, D_{Fmax}] \& !OTF \\ &\delta 5 : |y_{gap}| < D_{Fmin} \\ &\delta 6 : |y_{gap}| \epsilon [D_{Fmin}, D_{Fmax}] \& OTF \end{aligned} \quad (6)$$

Overtake Mode In overtake mode, the racecar follows the selected overtake trajectory [$\delta 7$] while continuously monitoring the progress of the overtake maneuver against the available controller resources with two possible transitions to (a) Merge Front Mode and (b) Fall Back Mode, as defined by the guard conditions [$\delta 8$] and [$\delta 9$]

$$\begin{aligned} \delta 7 :& OTF\&!PMF \\ \delta 8 :& OTF\&PMF \\ \delta 9 :& !OTF \end{aligned} \tag{7}$$

Merge Front Mode In merge front mode, the ego racecar is tasked with merging back onto the main raceline in front of the opponent racecar [$\delta 10$] while maintaining a safe distance from the opponent racecar with two possible transitions to (a) Normal Mode and (b) Fall back Mode, as defined by the guard conditions [$\delta 11$] and [$\delta 12$]. The current design of the Merge Front mode assumes that the opponent is unable (incapable, too aggressive, etc.) to avoid a collision with the ego, thus forcing the ego to not violate its safety constraints and ultimately abandon an overtake.

$$\begin{aligned} \delta 10 :& |y_{gap}| <= D_{sepF}\&OTF\&PMF \\ \delta 11 :& |y_{gap}| > D_{sepF} \\ \delta 12 :& !OTF|!PMF \end{aligned} \tag{8}$$

Fall Back Mode If the current overtake attempt becomes infeasible, the ego racecar transitions to the Fallback state. In this state, the ego racecar engages its full available braking power to prevent a collision with the opponent or the racetrack bounds [$\delta 13$]. The primary objective of the fallback state is to prevent an impending collision and then safely guide the ego racecar back to the global raceline [$\delta 14$] in the normal mode.

$$\begin{aligned} \delta 13 :& |y_{gap}| <= D_{sepB}\&!SFB \\ \delta 14 :& |y_{gap}| > D_{sepB}\&SFB \end{aligned} \tag{9}$$

The AutoPass framework is currently designed only to overtake an opponent and cannot defend the ego racecar's position from an attempted overtake by an opponent. Each state's output enables the different behaviours that make up the various overtake sequences. More information about the overtake sequences are described in the experiments section.

Experiments & Results

Experiment Setup We chose the F1Tenth racecar [2] platform for our experiments and conducted multiple tests using an indoor racetrack. We deployed two autonomous F1Tenth racecars in real world and the racing simulator [3]. The opponent racecar was initialized in the lead position for all experiments, and each experiment was 25 laps long for different values of boost energy and opponent advantage, with a total of 225 laps across all different variables on the physical testbed and 600 laps in simulation leading to a total of 33 experiments covering all variables).

The **Ego Racecar** was deployed with the AutoPass framework, while the **Opponent Racecar** was made to autonomously navigate the racetrack on the global raceline at high speeds using a pure-pursuit controller. Each racecar used a single 2D planar scanning LiDAR as the primary perception and navigation sensor with a feedback enabled motor and steering controller providing odometry data. Both LiDAR and odometry data were used by an online GPU particle filter [21] for fast and dependable localization at high speeds around the racetrack. To simplify the ability of the ego racecar in tracking the opponent, we enabled $base\_link$ (which is the standard ROS name for the racecar's main control frame) sharing across both racecar's through a centralized control computer.

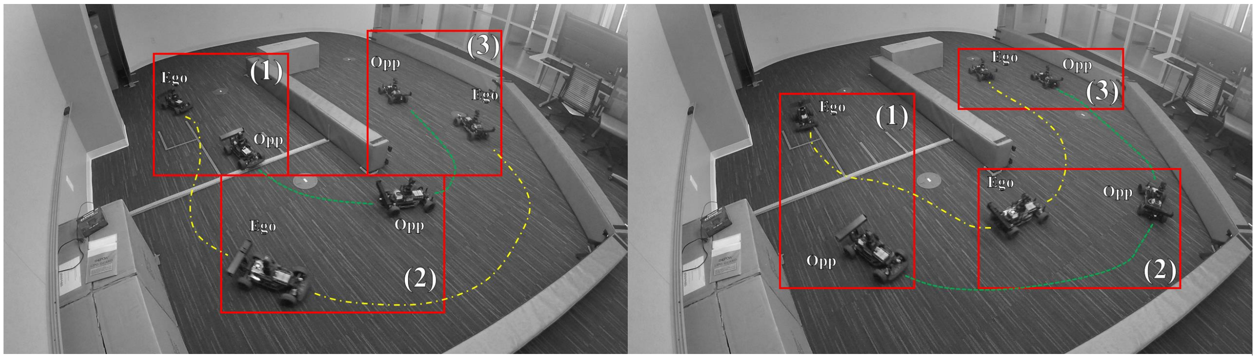
Figure 6 shows two outcomes of the AutoPass framework implemented on the F1Tenth racecars. The left half of the figure shows the ego racecar attempting to overtake the opponent from the outside of the turn and running out of boost energy, thus abandoning the overtake attempt. The right half of the figure shows the ego racecar attempting to pass the opponent from the inside of the turn using the boost energy provided to successfully pass the opponent and continue on the global raceline. This behavior in Figure 6 is similar to those observed in real motorsport racing.

Defined Overtake Sequences Figure 7[Left] provides the complete state transition sequence for the different overtake sequence labels described in this section. Each of the sequence labels define the outcome of an overtake attempt

- High Speed Overtake: (best case scenario), when the ego racecar approaches the opponent at a high speed, it immediately attempts an overtake because an overtake trajectory exists and completes the overtake by merging in front of the opponent. The ego racecar uses very little boost energy in this case.

- Normal Overtake: the ego racecar approaches the opponent and determines that an overtake is currently infeasible, so it follows the opponent until an overtake attempt is feasible and executes the overtake maneuver using the boost energy and successfully passes the opponent then merges in front of the opponent

- Normal Abandoned Overtake: the ego racecar executes the Normal Overtake sequence because it estimated that the overtake was feasible but a recalculation with updated state information finds that the attempt is no longer feasible (eg: ego racecar uses the entire boost energy before the attempt is successful, the opponent manages to increase the lateral separation during the attempt etc.) forcing the



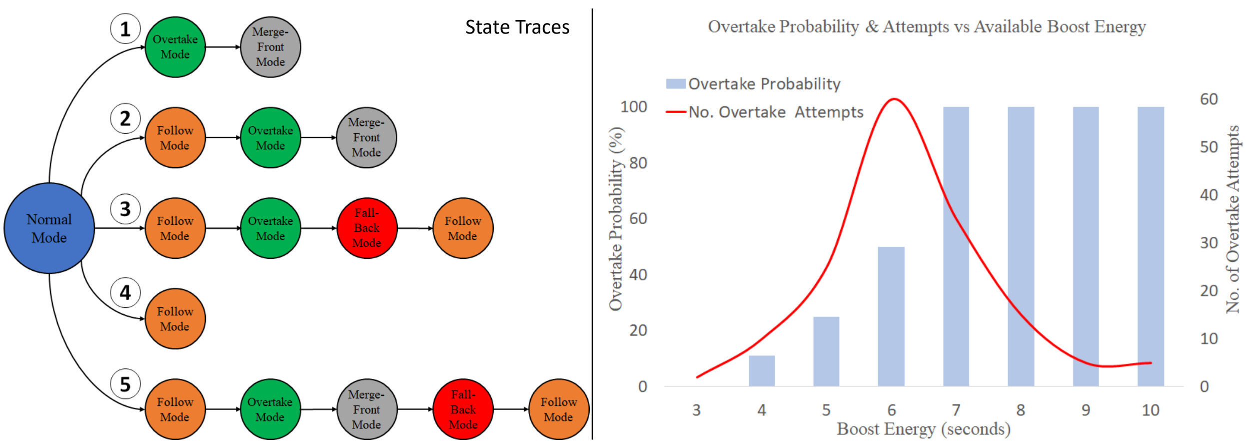


Figure 7: [Left]: Traces labels of valid state transitions (1) high speed overtake, (2) normal overtake, (3) normal abandoned overtake, (4) overtake infeasible, and (5) hybrid abandoned overtake; [Right]: Overtake probability and attempts vs boost energy. Red trace is the overtake probability and the Blue histogram is the size of the boost energy bank T

ego racecar to fallback behind the opponent declaring the overtake unsuccessful

- Overtake Infeasible: The ego racecar approaches the leading opponent but an overtake attempt is infeasible and will continue to remain infeasible for an extended time (narrow racetrack, not enough boost energy etc.), thus making the ego racecar follow the opponent in perpetuity until an overtake attempt becomes feasible

- Hybrid Abandoned Overtake: the ego racecar successfully passes the opponent in an overtake attempt but estimates that a merge-front maneuver might violate safety its constraints (e.g. when the opponent is overly aggressive and denies the ego an opportunity to complete the merge front maneuver by forcing an imminent collision between the racecars, etc.), thus the ego racecar abandons the current overtake attempt using all available braking effort to safely fallback behind the opponent. This trace is not frequently observed, but it shows the robustness of the AutoPass framework and its designed emphasis on vehicular safety

Figure 7[Right] shows (a) the probability that an overtake may be successful for an experiment for the different values of boost energy available, and (b) the number of overtake attempts for the corresponding values of boost energy. The results from this figure show that the probability of a successful overtake proportionally increase with the available boost energy. An interesting note here is that the ego racecar attempts to overtake the most number of times when the probability of success is around 50%, and continues to decrease with higher boost energy values showing that the ego racecar attempts fewer overtakes, but successfully completes each attempt at the higher boost energy values. This is because, in our two car experiment setup, when an overtake attempt is guaranteed to be successful - and the racecar does successfully overtake an opponent, it returns to the Normal Mode and continues to race along the global raceline and is unlikely to encounter the opponent until it leads the opponent by an entire lap. This situation is either likely to happen a long time into the future (depending on the opponent's comparative disadvantage, and entirely unlikely if both racecars are the same) and, it may not occur again before the end of the race

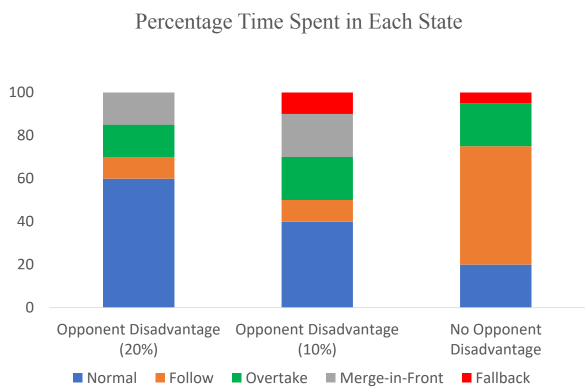


Figure 8: Time spent in each state vs opponent disadvantage

Figure 8 shows the amount of time the ego racecar spends in each state for a lap, averaged for all laps for different boost energy values and compared to a different opponent setup. When the ego racecar and opponent have the same max. rated velocities and the opponent starts in the pole position, the ego racecar attempts to overtake as seen in Figure 8[Right], and fails all the time, thus spending most of the time following the opponent. When the opponent is slightly disadvantaged in terms of the max. rated velocities (see Figure 8[Left, Middle]), the ego racecar is more likely to successfully overtake and merge in front of the opponent with only a small number of failures. The ego racecar spends equal amounts of time in both cases to pass the opponent and then merge in front of the opponent and less time following the opponent. This demonstrates that, when considering an equal or slightly disadvantaged opponent, a leading opponent on the global raceline will continue to lead the race and the inclusion of the Boost Energy Management system is necessary to perform the overtake since the overtake trajectories are often less efficient compared to the global raceline and the ego racecar needs the added performance boost to overcome this.

Method	Overtake Attempts	Successful Overtakes	Success Ratio
Reactive Overtake	59	1	0.017
ROS Navigation	26	3	0.115
TEB Planner	14	4	0.286
AutoPass	**43**	**17**	**0.395**

Table 3: Comparison of AutoPass with other methods capable of overtaking. Timed-Elastic Band (TEB) is a plugin to the ROS navigation stack.

Table 3 shows the performance of the AutoPass framework compared to other model free approaches to autonomous overtake. We define the success ratio as the number of successful overtakes to the total number of attempted overtakes. This metric shows (a) the effectiveness of the method being tested (number of successful overtakes), and (b) the efficiency of the methods when planning an overtake maneuver (number of overtake attempts). A purely reactive system - such as the generic highway lane departure and pass systems where a vehicle will attempt to take a low weighted cost passing (overtake) trajectory - produces a success ratio of 0.017, while the AutoPass system produces a success ratio 0.395. The AutoPass framework also outperforms the standard ROS navigation stack (which has a 0.115 success ratio), and the more sophisticated TEB planner [26] (which has a 0.286 success ratio). This improvement is most likely because the ROS planners emphasized hard constraints of safety over mission objective and proved to be extremely risk averse, whereas AutoPass works with soft constraints for mission objective while maintaining comparable safety standards.

Conclusion

In this paper we presented AutoPass - a novel framework for overtaking in a multi-agent setting. AutoPass uses the structure of an automaton to break town the complex task of overtaking into sub-maneuvers that balance overtaking likelihood and risk with safety of the ego vehicle. We presented real world implementation of 1/10 scale autonomous racing using two F1Tenth cars to demonstrate the effectiveness of AutoPass for the overtaking task. Our results indicate that the overtake success ratio for the AutoPass framework is 0.395 or 23 times more likely, compared to a purely reactive system at 0.017, while traditional ROS based path planners (depending on the navigation plugin used) are placed between 0.115 to 0.286. Our future work on this project is three-fold: first, we are working on a method to formally verify the AutoPass automaton to ensure that we can accurately predict when a successful overtake will be feasible; second, we plan to incorporate additional racing strategies into the AutoPass framework (e.g., strategic overtake attempts at certain sections of a race-track); and finally, we will implement adversarial characteristics on the opponent to further improve the AutoPass framework to work under these conditions.

References

1. Global championship of driverless cars. url=https://roborace.com/, journal=Roborace.

2. Matthew O'Kelly, Varundev Sukhil, Houssam Abbas, Jack Harkins, Chris Kao, Yash Vardhan Pant, Rahul Mangharam, Dipshil Agarwal, Madhur Behl, Paolo Burgio, et al. F1/10: An open-source autonomous cyber-physical platform. *arXiv preprint arXiv:1901.08567*, 2019.

3. Varundev Suresh Babu and Madhur Behl. f1tenth. dev-an open-source ros based f1/10 autonomous

racing simulator. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1614–1620. IEEE, 2020.

4. Albert Boretti. Energy flow of a 2018 fia f1 racing car and proposed changes to the powertrain rules. *Nonlinear Engineering*, 9:28–34, 08 2019.

5. Nikolce Murgovski and Jonas Sjöberg. Predictive cruise control with autonomous overtaking. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 644–649. IEEE, 2015.

6. Shilp Dixit, Saber Fallah, Umberto Montanaro, Mehrdad Dianati, Alan Stevens, Francis Mccullough, and Alexandros Mouzakitis. Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects. *Annual Reviews in Control*, 45:76–86, 2018.

7. Joshue Perez, Vicente Milanes, Enrique Onieva, Jorge Godoy, and Javier Alonso. Longitudinal fuzzy control for autonomous overtaking. In *2011 IEEE International Conference on Mechatronics*, pages 188–193. IEEE, 2011.

8. Alexander Buyval, Aidar Gabdulin, Ruslan Mustafin, and Ilya Shimchik. Deriving overtaking strategy from nonlinear model predictive control for a race car. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2623–2628. IEEE, 2017.

9. A. Buyval, A. Gabdulin, R. Mustafin, and I. Shimchik. Deriving overtaking strategy from nonlinear model predictive control for a race car. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2623–2628, 2017.

10. H. Huang and T. Wang. Learning overtaking and blocking skills in simulated car racing. In *2015 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 439–445, 2015.

11. D. Loiacono, A. Prete, P. L. Lanzi, and L. Cardamone. Learning to overtake in torcs using simple reinforcement learning. In *IEEE Congress on Evolutionary Computation*, pages 1–8, 2010.

12. Alexander Liniger and John Lygeros. A non-cooperative game approach to autonomous racing, 2019.

13. Tim Stahl, Alexander Wischnewski, Johannes Betz, and Markus Lienkamp. Multilayer graph-based trajectory planning for race vehicles in dynamic scenarios. *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Oct 2019.

14. Trent Weiss and Madhur Behl. Deepracing: a framework for autonomous racing. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1163–1168. IEEE, 2020.

15. Zheng Zhang, Peifeng Gao, Zilong He, Hongbin Chen, and Jianfeng Wang. Autonomous driving system design for formula racing. In *Journal of Physics: Conference Series*, volume 1651, page 012125. IOP Publishing, 2020.

16. Alexander Liniger, Alexander Domahidi, and Manfred Morari. Optimization-based autonomous racing of 1: 43 scale rc cars. *Optimal Control Applications and Methods*, 36(5):628–647, 2015.

17. B. Houska, H.J. Ferreau, and M. Diehl. ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.

18. Keith R Pullen. The status and future of flywheel energy storage. *Joule*, 3(6):1394–1399, 2019.

19. Wikipedia. Push-to-pass, 2020. [Online; accessed 15-Dec-2020].

20. D. Choi. Development of open-source motor controller framework for robotic applications. *IEEE Access*, 8:14134–14145, 2020.

21. Corey Walsh and Sertac Karaman. Cddt: Fast approximate 2d ray casting for accelerated localization. abs/1705.01167, 2017.

22. Zheng Zhang, Peifeng Gao, Zilong He, Hongbin Chen, and Jianfeng Wang. Autonomous driving system design for formula racing. In *Journal of Physics: Conference Series*, volume 1651, page 012125. IOP Publishing, 2020.

23. Robin Verschueren, Stijn De Bruyne, Mario Zanon, Janick V Frasch, and Moritz Diehl. Towards time-optimal race car driving using nonlinear mpc in real-time. In *53rd IEEE conference on decision and control*, pages 2505–2510. IEEE, 2014.

24. Z. Lu, B. Shyrokau, B. Boulkroune, S. van Aalst, and R. Happee. Performance benchmark of state-of-the-art lateral path-following controllers. In *2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)*, pages 541–546, 2018.

25. Nishant Chowdhri, Laura Ferranti, Felipe Santafé Iribarren, and Barys Shyrokau. Integrated nonlinear model predictive control for automated driving. *Control Engineering Practice*, 106:104654, 2021.

26. Christoph Rösmann, Frank Hoffmann, and Torsten Bertram. Integrated online trajectory planning and optimization in distinctive topologies. *Robotics and Autonomous Systems*, 88:142–153, 2017.