Local NMPC on Global Optimised Path for Autonomous Racing

Dvij Kalaria, Parv Maheshvari, Animesh Jha, Arnesh Kumar Issar,
Prof. Debashish Chakravarty, Prof. Sohel Anwar, Prof. Andres Tovar

***Abstract*—The paper presents a strategy for control of an autonomous racing car in pre-mapped environments. Using a dynamical model of the vehicle, the optimal racing line is computed taking track boundaries into account. With the optimal racing line as reference, a local NMPC based controller is proposed which takes into account local objectives like making more progress along the race line, avoiding collision with opponent vehicles and use of drafting.**

***Index Terms*—Model Predictive Control, Autonomous Racing, Vehicle Dynamics at High Speeds, Drafting, Multi Agent Racing**

I. Introduction

Autonomous Driving is an active field of research that poses challenges in various fields including Motion Planning, Controls, and Perception. The task can be as simple as lane following and goal-reaching or can be complex as multi-agent driving scenarios. Finding optimal trajectories that are feasible is a key challenge associated with this field. However, for low speed autonomous driving, these challenges can be solved relatively easily and there is an extensive work on the same. High speed autonomous driving on the other hand although discussed in a lot of works, is not heavily researched due to the challenging experimental setups and safety aspects of high-speed driving. With the advent of various competitions like Roborace and the ongoing Indy Autonomous Challenge, the field of high speed autonomous racing is bound to attract a lot more attention. Various small-scale setups are also established for experimental work in this field which can help to standardize this field for future research. This paper describes the design and implementation details of an optimisation based planning and control algorithm used for autonomous racing in a pre-mapped environment as a 2 stage process. We first find an approximation of the global optimal trajectory and then use it for local Non linear Model Predictive Control (NMPC) with multiple objectives including inter vehicle collision avoidance. This work was carried out as a part of the Indy Autonomous Challenge (IAC) [7], all algorithms described in this paper were simulated and tested using ANSYS VRXPERIENCE simulator and the vehicle models and scenarios provided by the IAC team. This paper does not concern itself with localisation and perception in high speed scenarios i.e. it assumes the availability of the accurate position of the ego vehicle and of the opponent vehicles within a certain range admissible for the RADAR or LIDAR sensors to detect. The paper makes the following novel contributions

1) Use of an optimal racing line as reference as opposed to the conventional use of center line
2) Dynamically modelling the tire forces at high speeds using a modified Pacejka model.
3) Incorporating a drafting effect model within the control scheme for overtaking scenarios.
4) The proposal to merge with the reference line at turns which gets more progress along the track

II. Related works

Autonomous driving in real world scenarios i.e. urban day to day settings has been around for a long time and has been studied deeply in recent years. Autonomous racing on the other hand is less explored. However, recent work has made advances in this field.

Modelling vehicle dynamics using kinematic bicycle models for Model Predictive Control was studied by [10], a dynamic bicycle model with non linear tire forces was studied by [8]. Tire forces are generally modelled by the Pacejka tire model [4] however most previous works [8] use a simplified model and do not consider the dynamic nature of the model at higher speeds. The presence of a pre-mapped environment allows us to use an optimised reference line instead of simply using the center line. Obtaining a minimum time race-line has been studied by [1] where they solve a global optimization problem to obtain the race-line.

The algorithms used for local motion planning in the context of autonomous racing can be divided into 3 classes: 1) Sampling based approaches 2) Graph based approaches 3) Numerical optimization based approaches. Sampling based approaches/incremental search finds its strength under unstructured or unknown environment [9]. Graph-based approaches span a spatial or spatio-temporal hierarchical tree along the drivable area. Commonly each node in the graph hold a cost calculated based on a heuristic. Using graph-search algorithms, the optimal path is chosen. Basically, these approaches discretize the action or spatial space in some way and choose the optimal track accordingly by exhaustively exploring all options by building a tree. They have proven to be effective in autonomous driving scenarios [14] [11] [6] but however one of the key issue that they face in the case of racing is the precise and sensitive application of controls to achieve which, the level of discretization has to be increased leading to high computation time. The third class is the optimization based methods which are mostly dedicated to Model Predictive Control (MPC) [2] [8]. They mostly focus on optimizing an objective to get the control commands. They have achieved good results in the context of autonomous racing, but is an active research topic both in terms of computation time and achieving more optimal results. Optimization based approaches for control of autonomous racing vehicles like NMPC and MPCC have been explored by [12] which further incorporate obstacle avoidance with the controller.

III. Vehicle Model

Modelling a race car is a challenging task due to non linear behaviour at high speeds. Our vehicle model has three sub-models, a dynamic model which predicts the next state given the current state and control commands, a tire model which models the forces on the tires in high speed scenarios, and a drafting model which takes into account the changes in vehicle dynamics due to drafting.

A. Dynamic model

Our base model has been adapted from [8] and acts under the similar assumptions

- The vehicle drives on a flat surface
- Load transfer can be neglected
- The longitudinal drive-train forces act on the center of gravity.

The equation of motion can be described using

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\phi} \\ \dot{v}_x \\ \dot{v}_y \\ \dot{r} \end{bmatrix} = \begin{bmatrix} v_x cos(\phi) - v_y sin(\phi) \\ v_x sin(\phi) + v_y cos(\phi) \\ r \\ \frac{1}{m}(F_{R,x} - F_{F,y} sin(\delta) + m v_y r) \\ \frac{1}{m}(F_{R,y} + F_{F,y} cos(\delta) - m v_s r) \\ \frac{1}{I_z}(F_{F,y} l_F cos(\delta) - F_{R,y} l_R) \end{bmatrix} \tag{1}$$

$$\tilde{\mathbf{u}} = \begin{bmatrix} \delta & D \end{bmatrix}^T \tag{2}$$

$$\tilde{\mathbf{x}} = \begin{bmatrix} X & Y & \phi & v_x & v_y & r \end{bmatrix}^T \tag{3}$$

$$\begin{aligned} F_{F,y} &= f_F(\alpha_F, F_{F,z}) \\ &\alpha_F = -\arctan\left(\frac{\omega l_F + v_y}{v_x}\right) + \delta, \\ &\text{and } F_{F,z} = F_{s,F,z} + \frac{1}{2} C_{lF}.\rho.S.v_x^2 \end{aligned} \tag{4}$$

$$\begin{aligned} F_{R,y} &= f_R(\alpha_R, F_{R,z}) \\ &\alpha_R = \arctan\left(\frac{\omega l_R - v_y}{v_x}\right), \\ &\text{and } F_{R,z} = F_{s,R,z} + \frac{1}{2} C_{lR}.\rho.S.v_x^2 \end{aligned} \tag{5}$$

$$\begin{aligned} F_{R,x} &= (C_{m1} - C_{m2} v_x) D - C_R - F_d \\ &F_d = \frac{1}{2} v_x^2 \rho \cdot C_d \cdot S \cdot \alpha_{cd} \end{aligned} \tag{6}$$

where the car has a mass m and an inertia I_z, l_R and l_F represent the distance form the center of gravity to the rear and the front wheel respectively, F_d is the aerodynamic drag exerted on the race car, α_{cd} is the coefficient introduced due to drafting (Refer to Section III-C), ρ is air density, S is the area of the race car and C_d is the drag coefficient of the car. $F_{R,y}$ and $F_{F,y}$ are the lateral tire forces of the rear/front wheel, F_x is the combined force produced by the drive-train. The state of the model $\tilde{\mathbf{x}}$, consists of X, Y and ϕ the position and heading in a global coordinate system, as well as the v_x, v_y the longitudinal and lateral velocities, and r the yaw rate. The control input $\tilde{\mathbf{u}}$ consists of δ the steering angle and D the driving command. The driving command replicates the pedals of a driver where $D = 1$ corresponds to full throttle and $D = -1$ to full braking. $F_{s,F,z}$ and $F_{s,R,z}$ are the normal forces exerted on the front and rear tires of the vehicle when static. Further we consider the effect of the lift force exerted on the race car. Considering C_{lF}, C_{lR}, the coefficients of lift for front and rear, ρ, air density, and S, the effective contact area of the race car, we can model $F_{F,z}$ and $F_{R,z}$, the vertical loads on front and rear tires at velocity v_x. The forces described given slip angle are modelled using functions f_R and f_F for front and rear tires which represent the interaction between tires and track surface calculated using the tire model defined in Section III-B.

B. Tire model

The Tire-Road Interaction has been modelled using the Pacejka model [4]. Due to the design of race-cars, tire load increases at high speeds due to a negative coefficient of lift. Race-cars work in high speed scenarios for a large part of their operation, therefore it is important to consider these load dependent changes. For this we have used a dynamic load dependent Magic Formula which models coefficients as a function of tire load [13]. The steady state tire characteristic function $F_y = f(\kappa, F_z)$ calculates the longitudinal force F_x on the tire and is based on F_z, the vertical load and the wheel slip angle κ.

$$V_{sx} = r_w \Omega - V_x \tag{7}$$

$$\kappa = \frac{V_{sx}}{|V_x|} \tag{8}$$

Using these quantities we can now model a dynamic Pacejka model as follows:

$$df_z = \frac{F_z - F_{z0}}{F_{z0}} \tag{9.1}$$

$$S_{Hy} = P_{Hy_1} + P_{Hy_2} \cdot df_z \tag{9.2}$$

$$S_{Vy} = F_z \cdot (P_{Vy_1} + P_{Vy_2} \cdot df_z) \tag{9.3}$$

$$\kappa_y = \kappa + S_{Hy} \tag{9.4}$$

$$C_y = P_{Cy_1} \tag{9.5}$$

$$D_y = \mu_y \cdot F_z \tag{9.6}$$

$$\mu_y = P_{Dy_1} + P_{Dy_2} \cdot df_z \tag{9.7}$$

$$\begin{aligned} E_y = (&P_{Ey_1} + P_{Ey_2} \cdot df_z + P_{Ey_3} \cdot df_z^2) \\ &[1 - P_{Ey_4} \cdot sgn(\kappa_y)] \end{aligned} \tag{9.8}$$

$$K_{y\kappa} = F_z \cdot (P_{Ky_1} + P_{Ky_2} \cdot df_z) \cdot e^{P_{Ey_3} \cdot df_z} \tag{9.9}$$

$$\tag{9.10}$$

$$= \frac{K_{y\kappa}}{C_y D_y + \epsilon_y} \tag{9.11}$$

$$F_{y0} = D_y \sin(C_y \arctan(B_y \kappa_y - E_y[B_y \kappa_y - \arctan(B_y \kappa_y)])) + S_{Vy} \tag{9.12}$$

where, df_z represents the fractional change in tire load wrt the reference tire load, F_{z0}, S_{Hy} and S_{Vy} represent offsets to the slip and longitudinal force in the force-slip function, or horizontal and vertical offsets if the function is plotted as a curve. μ_y is the longitudinal load-dependent friction coefficient. ϵ_y is a small number inserted to prevent division by zero as F_z approaches zero.

C. Drafting model

Drafting or slip streaming is an aerodynamic technique where two vehicles or other moving objects are caused to align in a close group, reducing the overall effect of drag on the preceding vehicle due to exploiting the lead object's slipstream. The effects of drafting on vehicle dynamics in the context of autonomous racing has been studied in [3] We use similar formulation to express the draft forces as a reduction of the aerodynamic drag forces. In the Figure 1 if the preceding vehicle which is vehicle B in this case, is in the “area of application” of the vehicle A, a “draft force” is applied on vehicle B. As in Section III-A, “normal drag” slows the car and is applied in –x, the “drafting-induced drag force” should be applied in +x. Mathematically we express drafting by multiplying a coefficient α_{cd} to the normal drag force (F_d in Equation 6) experienced by the vehicle.

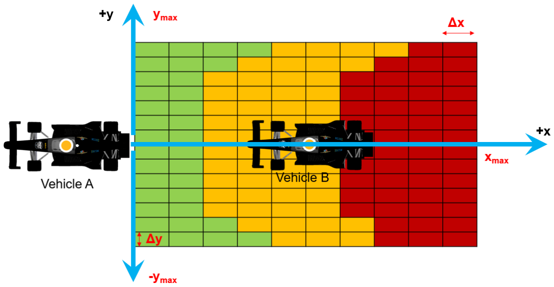

Fig. 1: Area of application for drafting

$$\alpha_{cd} = 1 + (\frac{v}{v_{max}}) \cdot (\beta_{cd} - 1), \quad \text{where } \beta_{cd} = \min(k_c + k_x \cdot \Delta x + k_y \cdot \Delta y, 1) \tag{10}$$

IV. Methodology

Assuming that we have pre-mapped track with accurate lane boundaries and static obstacles if any, we propose to divide the problem into 2 parts: 1) generating the optimal racing line using the known lane boundaries and well defined track and 2) using local MPC with the optimal racing line and sensor data as input for considering opponent vehicle collision avoidance and other objectives. We propose an objective function for the local MPC which uses optimal racing line to estimate the optimal step towards lap completion and also other strategies like leveraging the use of drafting and constraints such as staying within the lane limits and collision avoidance with opponent vehicles.

A. Calculating global racing line

The first step in planning is to get the optimal global racing line. We need to get the time optimal trajectory as described in [1] following the optimization problem described in Equation 11, where θ_k is the state, u_k is the action command, and $l(\theta_k, u_k)$ is the increase in progress along the center line for the time step k. x_{start} is the start state which can be kept as the start condition within the admissible states X, and F is the vehicle model function described in Section III. This ultimately maximizes the distance travelled along the center reference line and thus minimizing time. The continuous set X encapsulates the lane boundaries and static obstacles if any. We have used the Global Race Trajectory Optimization package provided by TUMFTM which has similar implementation. In this package, we gave the following inputs:- track info, lateral and longitudinal accelerations limit due to friction and the model parameters. The output of this package is a optimal racing line on the given race track in from which we use an array of size R^{N*4} where N is the number of points depends on the stepsize and track length. The 4 columns are structured as follows: X-coordinate of raceline point, Y-coordinate of raceline point, target velocity X, and target velocity Y at raceline point.

$$\begin{aligned} \max_{\theta,u} \quad & \sum_{k=0}^{N-1} l_k(\theta_k, u_k) \\ s.t. \quad & \theta_0 - x_{start} = 0, r \\ & \theta_{k+1} - \Delta t F(\theta_k, u_k) = 0 \\ & x_k \in X \end{aligned} \tag{11}$$

B. Local planner using NMPC

With the optimal trajectory and local obstacles and opponent vehicle positions as input, we propose to formulate a non linear Model Predictive Control formulation which considers both the motives of making maximum progress along the track while avoiding collisions with other vehicles and at the same time leveraging the advantage of local factors like drafting effect (Section III-C) to gain speed due to spatially local region created with reduced air resistance. However, most of the works have proposed a 2 level solution to this with a high level planner that outputs a traversable trajectory and a low level controller to track that trajectory as described under previous works (Section II), we propose a single step solution that merges both the steps i.e. it plans a trajectory by solving an optimization problem and taking the first planned control command as output. For solving the non linear problem we use Interior Point Optimization (IPOPT).

1) Overtaking strategy

The primary condition for the optimality of non linear MPC is the convexity of the loss function. However, in the case of overtaking, the optimization problem becomes non convex based on the side chosen to overtake against the target vehicle with IPOPT. Also, with more no of opponents or local obstacles, the decision becomes combinatorial in nature. The situation is well depicted in Figure 2. One solution to this problem is to use some heuristic to determine the topology or use a Dynamic Programming solution as depicted in [12] to get the optimal combination. However, we propose a simpler approximate solution to this by using the planned trajectory from previous step to determine the class. In case the distance with an opponent vehicle is less than a certain threshold T_d, we choose 2 combinations, one from the left of the target vehicle and the other from the right. We optimize both trajectories using the NMPC problem described later setting then as the initial trajectory. This approach can be fully parallelized on a multi-core system. The same solution can be extended based on the amount of computation and parallel cores available to include even more classes. For getting the initial trajectories, we use the follow-the carrot algorithm [5] with target positions as $X_t - d$ and $X_t + d$ where $X_t \in R^2$ is the position of the target vehicle at time step t and d is the displacement vector with directions perpendicular to the target vehicle and length can be chosen proportionately as per the vehicle dimensions. These initial trajectories are used for further optimization.

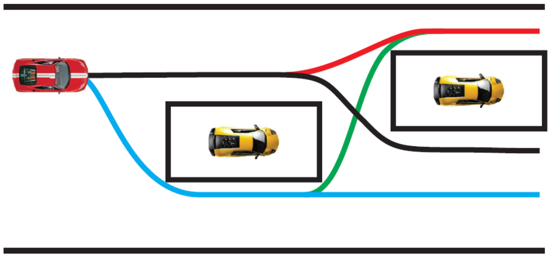

Fig. 2: Combinatorial nature of the overtaking problem, as it is possible to overtake each opponent on the left or the right side

2) *Cost functions*

The objectives to be considered by NMPC are: 1) making locally maximum progress along the track using global racing line as the reference, 2) avoiding collision with opponent vehicles and other local static or dynamic obstacles, 3) using drafting effect to increase chances of overtaking the vehicle ahead and 4) avoiding longitudinal or lateral slipping of vehicle or any other unsafe state and the constraint to stay within the lane boundaries

There has been a substantial study into deriving a strategy to maximize progress along the reference line using local MPC. One such work is [12] which proposes a trade-off between tracking the reference trajectory by minimizing the contouring error which they define as the perpendicular distance from the reference line and the maximization of progress which they define as distance along the trajectory from the corresponding perpendicular projection points from the start and end position of the locally planned trajectory. However, we argue that this fits well when the reference line is the center line as they have used, but since we are using the globally optimal trajectory as the reference, we propose a novel cost term which incorporates both the tracking of the race line and what we believe as the optimal way to merge with the race line only at the time of turns to not interfere with the overtaking or obstacle avoidance objective. First, we define the projection function P as defined in Equation 12 that finds the projection point on the reference line. However this itself is an optimization problem, hence we find the approximate projection point by starting with the projection point along X axis with reference frame located at the ego vehicle's center position and optimizing with r no. of iterations. Choosing $r = 3$ gave empirically very close projection point. The cost term is as defined in Equation 13, where v_t denotes the speed of the ego vehicle at time step t, θ_t denotes the heading of the ego vehicle at time t, α denotes the parameter for perpendicular projection point on the approximate spline of the race line, $\theta(\alpha)$ and $R(\alpha)$ are the tangential angle and the radius of curvature at the perpendicular point. This term would thus put more weight on merging with race-line only during turns when the radius of curvature of the projection point is less. The comparison in the results when the contouring error cost term was used vs when the new cost term was used has been carried out in section V.

$$P(X,Y) \triangleq \min_{\theta} (X - X^{ref}(\theta))^2 + (Y - Y^{ref}(\theta))^2 \tag{12}$$

$$\begin{aligned} C_{pt} &= \frac{v_t cos(\theta_t - \theta(\alpha))R(\alpha)}{R(\alpha) + |D_k|}, \\ &\text{where } D_k = \sqrt{(X - X(\alpha))^2 + (Y - Y(\alpha))^2}, \\ &\text{and } \alpha = P(x_t, y_t) \end{aligned} \tag{13}$$

The second objective is to avoid collision and maintain safe distance for opponent vehicles and static obstacles if any. This has been studied extensively in prior works. The method in [2] sets up a cost term in the objective function to penalize coming close to the obstacle with a penalty inversely proportional to to the distance from the obstacle and directly proportional to the speed. However, as we have dynamic obstacles, we replaced the speed of the vehicle with the relative velocity magnitude between the opponent vehicle and the ego vehicle and instead of euclidean distance, we use the distance scaled with different proportions along x and y axis proportionate to the opponent vehicle dimensions such that we have ellipsoidal equivalent region. The resultant cost term $C^d_{i,j,t}$ at time step t, for target vehicle or obstacle i and sensor j is described

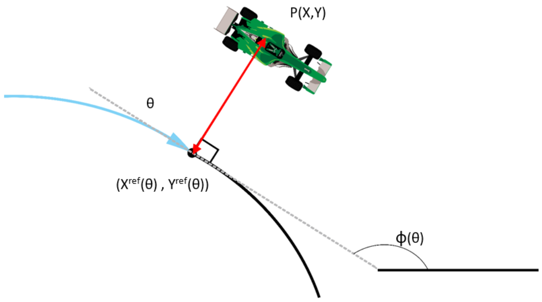

Fig. 3: The projection point on the optimal racing line

in equation 14 where (δx_{ijt},δy_{ijt}) denotes the position of the sensor j wrt the local coordinate frame of the target vehicle as shown in Figure 4 and $v_{i,t}$ denotes the relative velocity magnitude with the i^{th} target vehicle detected at time step t, and l_x, l_y denotes the length and width of the target vehicle. The term is also multiplied with sigmoid term to ensure non overlap and smooth transition from the drafting objective to the obstacle avoidance objective. S_{do} and K_{do} are constants which determine the smoothness of the transition and the distance at which the transition must occur.

$$C_{ot} = \sum_{i=0}^{M}\sum_{j=0}^{k} \frac{v_{ijt}}{(\frac{\delta x_{ijt}}{l_x})^2 + (\frac{\delta y_{ijt}}{l_y})^2}\sigma(S_{do}(K_{do} - \delta x_{ijt})),$$
$$\text{where } \sigma(z) = \frac{e^z}{1+e^z} \tag{14}$$

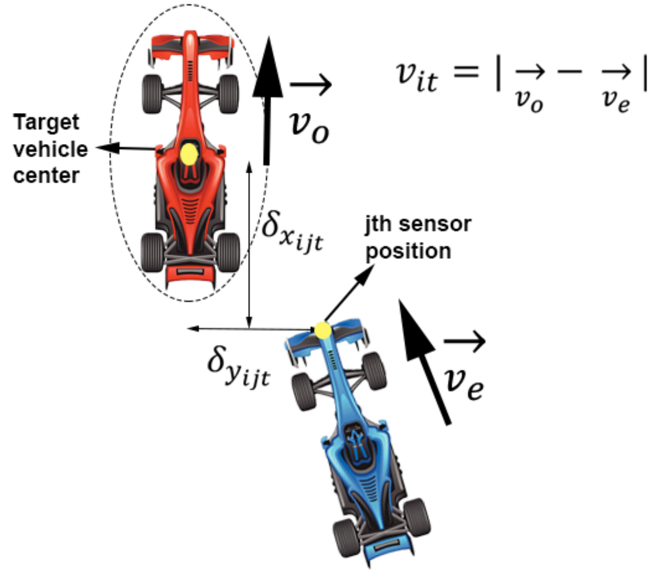

Fig. 4: The reference frame for δx_{ijt} *and* δy_{ijt}

The third objective is to use the drafting effect which is basically the spatially local regions created behind the opponent vehicles where a reduced wind force is experienced. To consider this, we propose the cost term as described in Equation 15 where M is the no. of vehicles and (δx_{it},δy_{it}) denotes the position of the ego vehicle wrt the local coordinate frame of the target vehicle i at time step t. This objective term would attract the vehicle behind a certain vehicle and thus leverage the use of drafting. Also the term is multiplied by an activation term with opposite activation to the obstacle avoidance term to smoothly transition back to the obstacle avoidance objective when the vehicle comes close to the target for overtaking. The path followed by the vehicle before and after the drafting term has been demonstrated in Section V.

$$C_{dt} = \sum_{i=0}^{M} |\delta x_{it}|(1 - \sigma(S_{do}(K_{do} - \delta x_{ijt}))) \tag{15}$$

For additional safety, we put up an extra penalty cost term if the lateral slip angles (α_F and α_R) goes beyond a safe value (α_{safe}) inspired from [8] as described in Equation 16 and a constraint that the vehicle remains within the lane boundaries resulting into an admissible space X_l. This has been implemented by spline fitting on the local lane boundaries as observed from camera or from the pre-mapped region if the localization is accurate and setting the constraint that the position (x_t, y_t) falls right to the left lane boundary approximate local fit spline and left to the right lane spline. The resultant optimization objective is described as below in Equation 17 where Δt is the time step and x_{start} is the start state which is passed as the current state of the vehicle, N is the no of time steps and F is the vehicle model function explained in Section III, k_{pt}, k_{ot}, k_{dt}, k_{st} are the constants to scale the individual cost terms described above.

$$\begin{aligned} C_{st} = (\alpha_F > \alpha_{safe})(\alpha_F - \alpha_{safe})^2 + \\ (\alpha_R > \alpha_{safe})(\alpha_R - \alpha_{safe})^2 \end{aligned} \tag{16}$$

$$\begin{aligned} \min_{\theta,u} \quad & \sum_{t=0}^{N-1} k_{pt}C_{pt} + k_{ot}C_{ot} + k_{dt}C_{dt} + k_{st}C_{st} \\ s.t. \quad & \theta_0 - x_{start} = 0, \\ & \theta_{t+1} - \Delta t F(\theta_t, u_t) = 0 \\ & x_t \in X_l \end{aligned} \tag{17}$$

C. *Algorithm*

The final algorithm is briefly described in Algorithm 1. Basically, we maintain a list of active obstacles to be passed on to NMPC, and also maintain the state for each of the dynamic obstacle where we store the previously decided direction for overtake. This will avoid possible anomaly due to numeric errors when the conditions are symmetric and the cost for overtaking from right or left is similar for the target vehicle. We take the N-1 planned steps from the previous NMPC step and check if for them the ego vehicle comes in vicinity to any opponent vehicle. If so, the overtaking strategy described above (Section IV-B1) is used to decide the direction of overtake and the initial trajectory is setup accordingly to be passed to NMPC for optimization. The first step on the resultant trajectory is executed and the process is repeated.

V. Results

In order to support the argument that using the new cost term for making more progress along the track using the optimal racing line, we compare the qualitative results using the new term with the contouring error term as used in [12] in Figure 5. From the results it can be empirically put that with the new cost term, as the merging happens at the time of turns, the net distance covered is less, also the term doesn't interfere with the overtaking objective during straight region of the track. Secondly, to demonstrate the effect of using drafting cost term in the presence of drafting, we compare the results with and without using drafting cost term as demonstrated in Figure 6 using the coefficient values: k_c = 0.805, k_x = 0.003, k_y = 0.0825 (refer to Section III-C)

Algorithm 1 Real-time NMPC algorithm

```
$State_{opp} \leftarrow []$
while TRUE do
    $Opp_{detected} \leftarrow$ Ego Ids of all detectable obstacles
    get current position and velocities
    function MpcCallBack
        $N$-1 planned steps from the previous NMPC step are checked.
        if Ego vehicle comes in vicinity to any opponent vehicle in $Opp_{detected}$ during any step. then
            $Target_{index}$ = index of closest vehicle
            $Target_{id} = Opp_{detected}[Target_{index}]$
            $Pref_{direction} \leftarrow Null$
            if Closest opponent same as closest opponent in previous iteration then
                $Pref_{direction} \leftarrow State_{opp}[Target_{id}]$
            else
                $State_{opp} \leftarrow []$
            $Traj_{init} \leftarrow$ overtaking strategy
            $State_{opp}[Target_{id}] \leftarrow$ Overtake Direction
                                                ▷ Overtake Direction = right or left
        else
            $Traj_{init} \leftarrow$ Straight Trajectory
            $State_{opp} \leftarrow []$
        $Traj_{init}$ is optimised using IPOPT.
    send first control at the end of time slot.
```

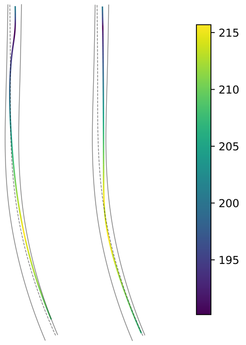

Fig. 5: Dotted trajectory is the optimal racing line, hard lines are the lane boundaries Left : Local trajectory followed with contouring error formulation, Right: Local trajectory followed with new cost term

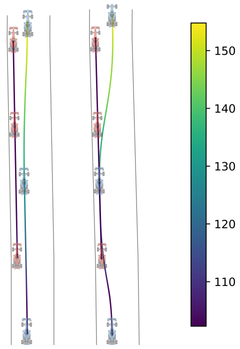

Fig. 6: Left : Local trajectory followed without drafting cost term, Right: Local trajectory followed with drafting cost term

VI. Conclusion

In conclusion, the algorithm proposed is able to compete considering multiple objectives while racing and able to effectively use the optimal racing line for taking decision. It uses numeric optimization on an initial decision path to consider multiple objectives like collision avoidance and effectively leverage the use of drafting for overtaking by staying behind the lead vehicle to gain more speed. The algorithm was tested on VRXPERIENCE simulator provided by Ansys as part of the Indy Autonomous Challenge and it is able to satisfactorily follow the objectives on the multi-agent head to head race.

Acknowledgment

This work was done as part of collaborative IUPUI-IITKGP-USB participating in the Indy Autonomous Challenge (IAC). All the algorithms mentioned were tested on VRXPERIENCE simulator provided by Ansys as part of the challenge. Special thanks to all the team members for their extensive support.

References

[1] Fabian Christ et al. “Time-optimal trajectory planning for a race car considering variable tyre-road friction coefficients”. In: *Vehicle System Dynamics* 59 (2019), pp. 588–612.

[2] Y. Gao et al. “Predictive Control of Autonomous Ground Vehicles With Obstacle Avoidance on Slippery Roads”. In: 2010.

[3] A. Guerrero and R. Castilla. “Aerodynamic Study of the Wake Effects on a Formula 1 Car”. In: *Energies* 13 (2020), p. 5183.

[4] E. Bakker H. Pacejka. “The magic formula tyre model”. In: *Vehicle System Dynamics* 21 (1991), pp. 1–18.

[5] R. Hogg et al. “Algorithms and sensors for small robot path following”. In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)* 4 (2002), 3850–3857 vol.4.

[6] Xuemin Hu et al. “Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles”. In: *Mechanical Systems and Signal Processing* 100 (2018), pp. 482–500.

[7] *Indy Autonomous Challenge*. URL: https://www.indyautonomouschallenge.com/challenge.

[8] Juraj Kabzan et al. “AMZ Driverless: Autonomous Racing System”. In: *ArXiv* abs/1905.05150 (2020).

[9] S. Karaman and Emilio Frazzoli. “Optimal kinodynamic motion planning using incremental sampling-based methods”. In: *49th IEEE Conference on Decision and Control (CDC)* (2010), pp. 7681–7687.

[10] Jason Kong et al. “Kinematic and dynamic vehicle models for autonomous driving control design”. In: *2015 IEEE Intelligent Vehicles Symposium (IV)* (2015), pp. 1094–1099.

[11] Xiaohui Li et al. “Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles”. In: *Mechanical Systems and Signal Processing* 87 (2017), pp. 118–137.

[12] Alexander Liniger, A. Domahidi, and M. Morari. "Optimization-Based Autonomous Racing of 1: 43 Scale RC Cars". In: *ArXiv* abs/1711.07300 (2017).

[13] H. Pacejka. "Tire and Vehicle Dynamics". In: 1982.

[14] T. Stahl et al. "Multilayer Graph-Based Trajectory Planning for Race Vehicles in Dynamic Scenarios". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (2019), pp. 3149–3154.