Hybrid MPC and Spline-based Controller for Lane Change Maneuvers in Autonomous Vehicles

Navil Abdeselam[1], Rodrigo Gutiérrez-Moreno[1], Elena López-Guillén[1], Rafael Barea[1], Santiago Montiel-Marín[1], Luis M. Bergasa[1]

***Abstract*— This paper presents a novel approach for collision detection and overtaking maneuver. The proposed technique is considered hybrid as it combines the smoothness of spline curves for nominal trajectories and the capabilities of Model Predictive Control (MPC) to respond to unexpected obstacles on the lane. To detect collisions, a temporal spline parameterization procedure is introduced. In the case of a possible collision, two decoupled longitudinal and lateral MPC controllers adjust the vehicle's path without modifying the nominal trajectory, using linear models to ensure short computation times. The complete control system has been implemented on Robot Operating System (ROS) using the hyper-realistic simulator CARLA, with successful results in overtaking scenarios.**

Lane change, hybrid planning, Model Predictive Control, collision prediction, overtaking

I. Introduction

In urban driving scenarios, obstacles are encountered that can partially or completely block the intended trajectory of vehicles. Thus, autonomous vehicles must be able to generate alternative trajectories avoiding such obstacles, or in the worst-case, come to a complete stop. Moreover, it must be ensured that the new path is both reliable and comfortable for passengers, while also guaranteeing kinematic and dynamic safety. These requirements pose a significant challenge, as the vehicle's target trajectory must be continuously updated in real-time.

Decision and control processes for autonomous driving have been traditionally divided into three levels [1]: (a) strategic level for decision-making tasks, (b) tactical level for trajectory generation, and (c) operational level for tracking of the planned trajectory. Maneuvers as overtaking, lane changing and obstacle avoidance are associated with the tactical and operational levels, which involve trajectory generation and tracking.

Thus, there are two general approaches to the lane-change problem: sampling-based methods [2] and Optimal Control Problem (OCP) methods [3]. Both techniques tipically involve discarding the nominal trajectory based on current obstacles. While this use of OCP is valid, it may be inefficient in challenging situations where the environment is constantly changing, especially when using Model Predictive Control (MPC) techniques [4] with the currently available solvers [5]. Furthermore, these solutions depend on the type of curve used (Bézier, Spline, etc.) to generate the nominal path [6].

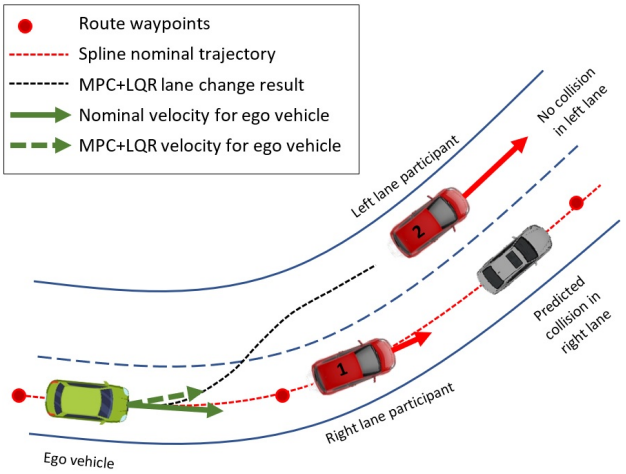

Fig. 1. Common scenario: The grey vehicle is the position where a collision is assumed to occur if no action is taken. For lane changing, MPC+LQR controllers are combined and generate lateral offset and speed adjustments resulting in black trajectory for overtaking.

The need to constantly update the ego-vehicle trajectory implies a high dependence on the processing speed of the system. For this reason, when using optimization techniques, the update frequency of the path may be limited. Thus, the method used to solve the problem must guarantee two things: (a) the existence of a reliable solution, and (b) short computation times. While the use of quadrating programming (QP) dates back to the 1950s [7], the first algorithms developed cannot be considered efficient. However, the results obtained by [8] demonstrate that the OSQP (Operator Splitting for Quadratic Programming) method can be applied to satisfy both requirements.

An authors' previous work developed a trajectory controller for tracking routes defined by waypoints using spline curves [9]. The study demonstrated the effectiveness of spline curves to interpolate a smooth trajectory, from which a velocity profile can be easily generated. Aditionally, a lateral LQR controller was implemented to ensure trajectory tracking. However, this study only addressed the tracking of nominal trajectories and did not consider the presence of other vehicles or obstacles that may obstruct the path.

The objective of this current study is to integrate an

This work has been supported from the Spanish PID2021-126623OB-I00 project, funded by MICIN/AEI and FEDER, and TED2021-130131A-I00, PDC2022-133470-I00 projects, funded by MICIN/AEI and the European Union NextGenerationEU/PRTR.

[1]N. Abdeselam, R. Gutiérrez-Moreno, E. López-Guillén, R. Barea, S. Montiel-Marín and L.M. Bergasa are with the Electronics Departament, University of Alcalá (UAH), Spain.{navil.abdeselam, rodrigo.gutierrez, elena.lopezg, rafael.barea, santiago.montiel, luism.bergasa}@uah.es

online MPC controller (with OSQP optimization), which can modify the trajectory with respect to the nominal path (spline curve) when lane changes are necessary to execute overtaking or obstacle avoidance maneuvers. The proposed method directly acts on the control signals and therefore requires an added system for collision prediction. For this task, vision based methods have been presented [10], as well as stochastic models [11]. In this work we will address the problem of lane changing by applying predictive control according to the temporal characterisation of the nominal spline-based trajectory to predict collisions. An overtaking criterion will be proposed based on these predictions. Figure 1 shows a common scenario in which vehicles 1 and 2 are possible obstacles for the nominal behavior of the ego-vehicle. The collision prediction system detects a future collision with vehicle 1 (grey vehicle) and an overtaking maneuver is started. The black pointed line is the hybrid trajectory (a combination of the spline nominal trajectory and the MPC).

The main contributions of this work are as follows:

- We introduce a new hybrid structure in the control architecture, coupled with a novel method for collision prediction based on time-parametrized splines.
- The proposed method has been successfully validated using the hyper-realistic AD simulator CARLA [12].
- Our experimental results demonstrate state-of-the-art accuracy, ensuring collision-free trajectories while avoiding the need for recalculating the nominal path.

II. Architecture

Figure 2 shows the global control architecture of the proposed system. It is said to be hybrid as, in the operational level, it combines a classical LQR controller for tracking a nominal spline-based trajectory (green blocks), and optimal control techniques from the MPC family to lane change execution (red blocks). In the tactical level, a collision prediction module and a decision making system (blue blocks) generate the signals that activate the lane change behaviour through the MPC modules when a future collision is detected on the nominal trajectory.

The trajectory generator using spline curves and the classical LQR controller for tracking them have been extensively documented in [9]. Spline curves provide a smooth and parameterized trajectory which allows for easy generation of velocity profiles ($V_{nominal}$) based on the radius of curvature. Moreover, the LQR controller enables minimization of tracking errors (angular error θ_e and lateral error d_e) in the nominal trajectory by intuitively adjusting its parameters. Its low execution time is also compatible with the high speeds of autonomous road vehicles. However, this previous work is constrained to tracking the nominal trajectory and does not consider potential hindrances such as obstacles or other vehicles.

In this study, the operational level of the control architecture is completed in order to perform lane changes when required. For this purpose, two MPC controllers (one for lateral and the other for longitudinal control) are introduced, which modify the control signals in real-time without the need to recalculate the nominal trajectory. The proposed hybrid structure ensures that these controllers are only activated when a lane change is necessary, allowing for the benefits of predictive control to be leveraged for this maneuver (as detailed in section IV), while mitigating its primary limitation, the high computation time.

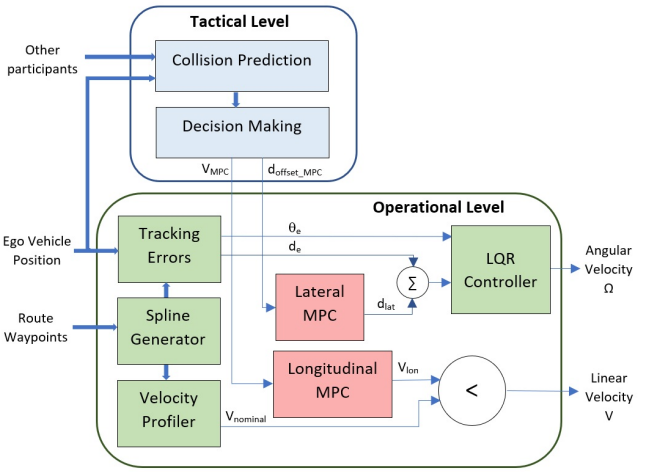

Fig. 2. Global architecture of the hybrid controller.

To initiate a lane change, a lateral offset reference signal d_{offset_MPC} equal to the lane width $\pm W$ (depending on wheter the change is to the left or right lane) is set as reference to the lateral MPC controller. This uses a linear model and some constraints to generate the lateral offset d_{lat} which, added to the nominal later error d_e, will produce a smooth lane change using the LQR controller. This approach is referred to as a surrogate control architecture, as the lateral MPC controller somewhat deceives the path control to obtain the desired behavior. In case that the lane change is not feasible to avoid collision with the preceding vehicle, a reduction of its linear velocity will be made through the longitudinal MPC module. To do this, its reference signal V_{MPC} is set to the preceding vehicle speed to achieve a cruise control.

In the tactical level, the collision prediction module uses the positions and velocities of the ego vehicle and other participants to predict collision times within a limited time horizon. These collision times are used by the decision making module to generate the references and constraints for the MPC controllers. The decision making module is part of a wider local planning system based on reinforcement learning [13] which is outside the scope of this paper. A simpler version of the module based on decision trees is proposed in section III.c to validate the control architecture in two-lane scenarios.

In the following sections, a comprehensive account of both the collision prediction approach and the MCP controllers will be provided.

III. Collision prediction method

Collision detection between dynamic objects requires efficient perception techniques as well as accurate models to predict future trajectories [14]. However, the movement of vehicles on roads occurs in lanes parallel to the nominal path. We propose a method that uses this nominal trajectory to obtain collision times with other vehicles in a fast and simple way.

A. Temporal parametrization of nominal trajectory

In [9] we develop the spline-based nominal trajectory generator. This curve allows n waypoints to be interpolated using $(n-1)$ third-order polynomial segments. Each segment i is characterised by two polynomials (one for each coordinate X_i and Y_i) as a function of a parameter u along the segment, normalized between 0 and 1:

$$\begin{aligned} X_i(u) &= a_{ix} + b_{ix} \cdot u + c_{ix} \cdot u^2 + d_{ix} \cdot u^3 \\ Y_i(u) &= a_{iy} + b_{iy} \cdot u + c_{iy} \cdot u^2 + d_{iy} \cdot u^3 \end{aligned} \tag{1}$$

For overtaking maneuvers, it is necessary to predict collisions of ego vehicle with the preceding vehicle in the same lane (numbered 1) and the vehicle in the left-hand lane (numbered 2), that can travel in the same or opposite direction. Assuming the velocities of the vehicles are known, it is possible to perform a temporal parameterisation of the nominal spline for each vehicle, using the following variable change:

$$u = \frac{t - t_i}{t_{i+1} - ti} = \frac{t - t_i}{\Delta_i} \tag{2}$$

where t_i is the time allocated to waypoint i according to the current vehicle speed. In this way, a continuous parameterisation of the curve as a function of time t is obtained, where each trajectory segment is defined by the following polynomials:

$$\begin{aligned} X_i(t) &= A_{ix} + B_{ix} \cdot t + C_{ix} \cdot t^2 + D_{ix} \cdot t^3 \\ Y_i(t) &= A_{iy} + B_{iy} \cdot t + C_{iy} \cdot t^2 + D_{iy} \cdot t^3 \end{aligned} \tag{3}$$

being:

$$\begin{aligned} A_{ix} &= -\frac{d_{ix} t_i^3}{\Delta_i^3} + \frac{c_{ix} t_i^2}{\Delta_i^2} - \frac{b_{ix} t_i}{\Delta_i} + a_{ix} \\ B_{ix} &= \frac{3 d_{ix} t_i^2}{\Delta_i^3} - \frac{2 c_{ix} t_i}{\Delta_i^2} + \frac{b_{ix}}{\Delta_i} \\ C_{ix} &= -\frac{3 d_{ix} t_i}{\Delta_i^3} + \frac{c_{ix}}{\Delta_i^2} \\ D_{ix} &= \frac{d_{ix}}{\Delta_i^3} \end{aligned} \tag{4}$$

and the same expresions for $Y_i(t)$ coeficients. Once the trajectories have been obtained as a function of time, the prediction of their evolutions can be made in order to determine possible spatio-temporal cut-off intervals between the vehicles involved in the scene.

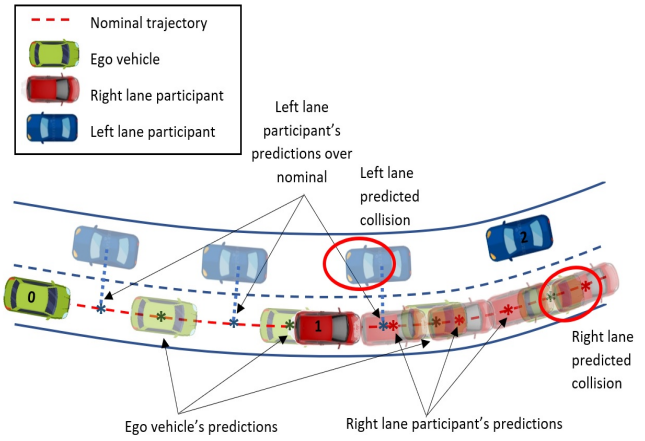

Fig. 3. Example scenario for predicting collision times: participant 1 is driving in front of the vehicle at a lower speed; in the left lane, participant 2 is driving in the opposite direction. All positions are projected onto the nominal trajectory, which is time-parametrised for each vehicle according to its speed (see figure 4). This makes it possible to predict the future positions of the vehicles on a time horizon (4 seconds) and to detect the collision times with the participants (circled positions).

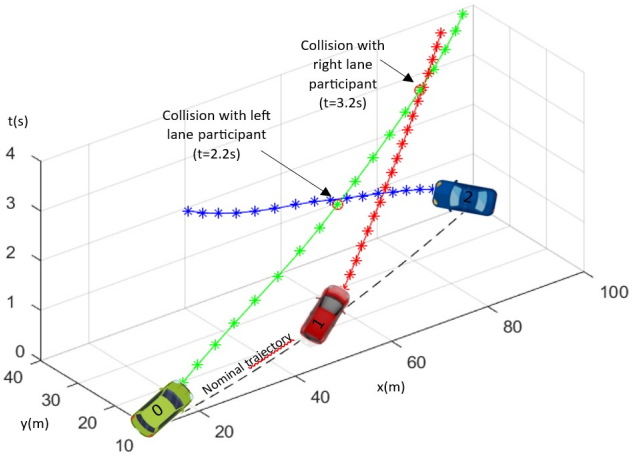

Fig. 4. Time parameterisation of the nominal trajectory for the three vehicles in the scenario in figure 3, as a function of their current speeds. The vertical axis corresponds to the time axis. The intersections between the trajectories occur in space-time, and allow the detection of collision times and positions (red circles in figure 3).

B. Collision time detection

The classical approach to collision detection using parameterised curves considers the trajectories of all agents in the scene to be known [15]. In domains where all agents are controlled, taking the references of all other objects for granted may be valid. However, autonomous driving aims to address an open and fully dynamic work frame. Therefore, considering a competent perception module, or assuming V2V communication, at best the position, orientation and velocity of objects surrounding the main vehicle will be known.

In this work we have approached the problem in a different way, projecting all the opposing vehicles onto the nominal trajectory, regardless of the lane in which they are travelling.

Once their positions have been projected onto the nominal spline, each vehicle performs its own temporal parameterisation of this spline according to its current speed. To do this, we calculate the times associated with each waypoint t_i, according to the lengths of the spline segments and the speed of each vehicle.

Figure 3 shows an example where vehicle 1 is driving ahead of the ego vehicle at a lower speed, while vehicle 2 is travelling in the left lane and opposite direction. After projecting the positions of each vehicle onto the nominal trajectory, each vehicle generates its own temporal spline (Figure 4, where the vertical axis is time), with a finite future time horizon T (4 seconds in the example). The intersections between the trajectories correspond to the prediction of collisions in space-time domain. The intersection between the ego vehicle trajectory and vehicle 1 predicts the collision time when travelling in the right lane, and between the ego vehicle and vehicle 2, the collision time travelling in the left lane. In the last case, there is a small temporal error due to the projection that will be compensated by the decision making system.

C. Decision making for lane change maneuvers

A complete decision making system for complex driving scenarios using reinforcement learning has been published by authors in [13]. In this paper we propose a simpler method that uses a decision tree (figure 5) to deal with two-lane scenarios.

The root of the decision tree is evaluated once the previous manuever has been fully completed. Using as inputs the collision times with the preceding vehicle and the vehicle in the opposite lane (within a time horizon T), the system provides as outputs the references for the lateral and longitudinal MPC controllers (d_{offset_MPC},V_{MPC}). For the lateral controller, d_{offset_MPC} can take values of the tuple $\{+W,-W,0\}$ to produce a change to the right lane, to the left lane, or to keep the current lane, being W the lane width. In the case of the longitudinal controller, V_{MPC} can take values from $\{V_0,V_1\}$, being V_0 the nominal speed of the ego vehicle and V_1 the speed of the preceding vehicle in the current lane. As shown in figure 5, the combination of these two reference signals will produce one of the following behaviours: (1) change to the left lane, (2) return to the right lane, (3) keep the nominal speed in the current lane, or (4) reduce the speed in the current lane to accommodate the preceding vehicle. The smoothness of these manoeuvres, as well as compliance with certain constraints on their control signals, is ensured by the MPC controllers.

IV. MPC Controllers

In the field of autonomous driving, the conventional approach to decision and control using MPC has relied on the kinematic bicycle model and dynamic model of the vehicle [16]. However, these nonlinear models require significant computation time for real-time execution at high velocities. Our hybrid method, which employs a nominal smooth trajectory with a velocity profiler for curves, does not require an exact vehicle model and can achieve better computation performance by using linear models. Additionally, since the MPC controllers are only used in lane change maneuvers, precision in the model is not a critical factor, since the LQR control operates continuously.

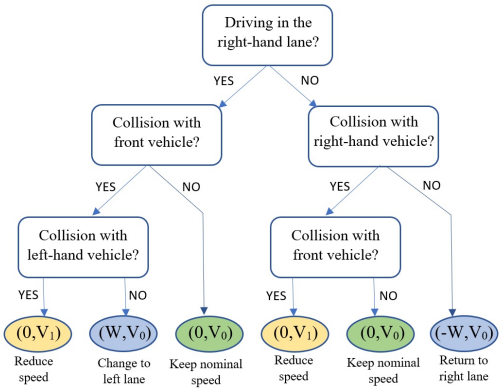

Fig. 5. Decision tree for two-lane scenarios. Outputs correspond to MPC controllers references: (d_{offset_MPC},V_{MPC}).

A. Motion integral models

The linear model used is a decoupled integrator chain for longitudinal and lateral dynamics.

For longitudinal dynamics a triple integrator chain based on jerk is used:

$$d_{lon} = \iiint j_{lon}(t)dt^3 \tag{5}$$

where d_{lon} is the longitudinal distance and j_{lon} the longitudinal jerk. Using d_{lon}, the longitudinal speed v_{lon} and the longitudinal acceleration a_{lon} as state variables, the discrete linear state-space representation with sample time T_s of this longitudinal model is:

$$\begin{bmatrix} \dot{d}_{lon} \\ \dot{v}_{lon} \\ \dot{a}_{lon} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_{lon} \\ v_{lon} \\ a_{lon} \end{bmatrix}_k + \begin{bmatrix} \frac{T_s^3}{6} \\ \frac{T_s^2}{2} \\ T_s \end{bmatrix} j_{lon_k} \tag{6}$$

The lateral model is a double integrator chain described by:

$$d_{lat} = \iint a_{lat}(t)dt^2 \tag{7}$$

being d_{lat} the lateral offset (distance) to the nominal spline trajectory and a_{lat} the lateral acceleration. Using as states d_{lat} and the lateral speed v_{lat}, the following state-space representation is obtained:

$$\begin{bmatrix} \dot{d}_{lat} \\ \dot{v}_{lat} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d_{lat} \\ v_{lat} \end{bmatrix}_k + \begin{bmatrix} \frac{T_s^2}{2} \\ T_s \end{bmatrix} a_{lat_k} \tag{8}$$

The lateral output of the MPC is a smooth lateral offset that is adapted to the presence of obstacles on the road.

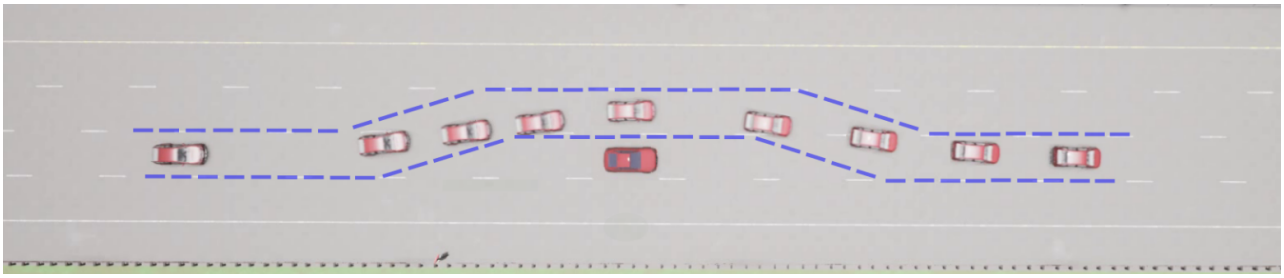

Fig. 6. Simulation scenario in CARLA. Dashed lines depict MPC constraints. The secondary vehicle is stationary and ego vehicle's movement is represented by multiple frames.

B. Constraints

All the states and control variables are bounded constrainst. In the longitudinal model, the jerk has been included to add a component of comfort, being its value limited to j_{max}. The acceleration is bounded between maximum deceleration a_{min} and maximum acceleration a_{max}, and the speed is bounded between 0 and the maximum speed given by the velocity profiler of the spline trajectory $v_{lon_{max}}$. The longitudinal distance is also bounded to avoid a collision with the front vehicle located at a distance D_{front}. These are the constraints for the longitudinal model:

$$\begin{aligned} 0 \leq d_{lon} &\leq D_{front} \\ 0 \leq v_{lon} &\leq v_{lon_{max}} \\ a_{min} \leq a_{lon} &\leq a_{max} \\ -|j_{max}| \leq j_{lon} &\leq |j_{max}| \end{aligned} \tag{9}$$

In order to achieve a smooth lane change without encroaching into adjacent lanes, constraints play a crucial role in lateral control. Being W the lane width, V_w the vehicle width, and $v_{lat_{max}}$ and $a_{lat_{max}}$ the maximum lateral velocity and acceleration, the lateral constrainst are:

$$\begin{aligned} -\tfrac{1}{2}W + \tfrac{1}{2}V_w \leq d_{lat} &\leq \tfrac{3}{2}W - \tfrac{1}{2}V_w \\ -|v_{lat_{max}}| \leq v_{lat} &\leq |v_{lat_{max}}| \\ -|a_{lat_{max}}| \leq a_{lat} &\leq |a_{lat_{max}}| \end{aligned} \tag{10}$$

V. Results

Our experiments run on a PC equipped with an Intel® Core™ i7-9700 CPU at 3.00 GHz and 32 GB of RAM. The proposed control architecture is implemented using C++ in the ROS Noetic framework. Besides, the MPC problem has been adapted to a QP type problem so that using the OSQP solver [8] we obtain the optimal solutions as well as the solvability. First of all, we analyse the use of MPC compared to the direct lateral offset introduction to the LQR trajectory controller on our own scenario. Then, a typical scenario proposed in [17] is used to compare our approach against three representative state-of-the-art methods: Minimun Snap Method (MSM) [18], Picewise Jerk Method (PJM) [19] and Polinomyal QP-based method [17].

A. Ablation study

We have designed an overtaking scenario in CARLA to analyze the behavior of our system in an ablation study. The scenario involves a motionless vehicle positioned in the middle of the ego vehicle's lane. At a safe distance, the collision detection system triggers the overtaking maneuver, as illustrated in Figure 6. Thanks to the incorporation of both constraints and the suggested integrators, our system achieves a successful response.

In Figure 7, a comparison is presented between our whole proposed method and an alternative approach with the lateral MPC (our main contribution) disabled. In the alternative case, the decision-making module's signal is directly introduced as an offset to the trajectory follower's input. As observed, the LQR exhibits inadequate handling of a step signal at the error input, resulting in oscillations in the response. Although the overtaking maneuver in this particular case is of short duration, it should be noted that the alternative method would not be viable for prolonged lane changes, where extended travel in a lane different from the nominal one becomes necessary.

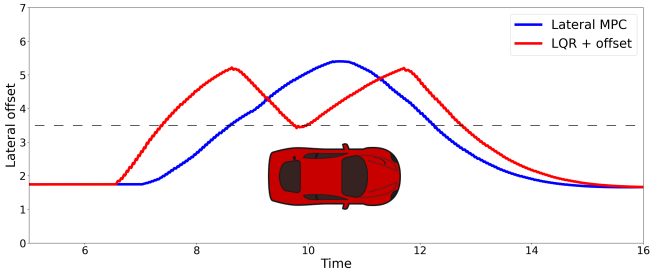

Fig. 7. Lane change maneuver example: The MPC-based overtaking is illustrated in blue, while the results of directly introducing the offset signal to the LQR controller are depicted in red.

The observed oscillations in the alternative method's response can be partially attributed to the controller's hyperparameter values. While the calibration of the LQR primarily focuses on enhancing tracking accuracy, it may not adequately address overtaking maneuvers. In contrast, model-based control also utilizes hyperparameters, but due to the linear and relatively simple nature of the state variable systems, once calibrated, they are well-matched to the trajectory controller's response, ensuring smoothness and reliability.

B. Comparison with other state-of-the-art methods

To evaluate our approach in comparison to the current state of the art, we analyze a common scenario proposed by [17] where traffic lanes are partially occupied by obstacles.

The prevailing methods primarily rely on optimization-based planning, mainly focusing on path generation. In Figure 8, we showcase the nominal trajectories generated by the state-of-the-art methods with the same well-tuned cost function parameters, juxtaposed with the results of the avoidance maneuver implemented in our method.

As indicated in [17], while the MSM method ensures snap continuity, its reliability is heavily dependent on cost function parameters. This reliance on parameter values makes the approach less reliable in complex scenarios. On the other hand, the PJM method does not guarantee jerk continuity, resulting in increased vibration and reduced smoothness in the final trajectory. Regarding our approach to the matter, when considering comfort along the nominal trajectory, the utilization of splines and the velocity profiler effectively highlights their functionality. However, with regards to lateral deviation, as explained in the preceding section, smoothness during maneuvers is achieved by filtering the d_{offset_MPC} signal through the integrator chain and appropriately configuring the operational limits of the state variables.

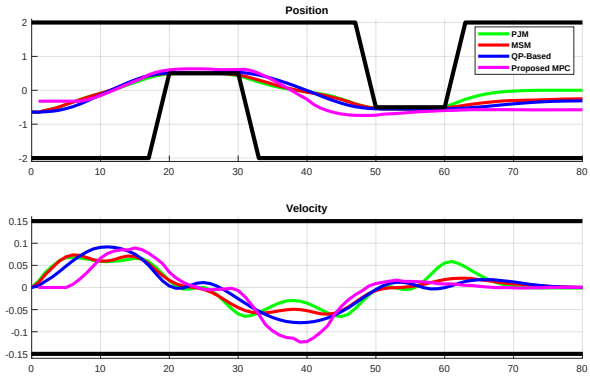

Fig. 8. Comparison between our method, MSM, PJM, and Two-Phase QP-based: The figure illustrates the optimized paths and velocity.

In contrast to state-of-the-art methods that assume strict adherence of the vehicle to generated trajectories without considering posible errors in the path tracking controller, our proposal takes a different approach. This key distinction ensures that the trajectories are always collision-free, unlike the method proposed by Yuncheng Jiang et al. [17] that may require multiple iterations to achieve this goal.

VI. Conclusion

We present a novel hybrid approach for collision detection and overtaking maneuvers in autonomous vehicles. The proposed technique combines the smoothness of spline curves for nominal trajectories and the capabilities of MPC to respond to unexpected obstacles on the lane. Our experiments demonstrated successful results using the CARLA simulator, implementing the control system in ROS. By introducing decoupled longitudinal and lateral MPC controllers, we achieved effective collision avoidance while maintaining short computation times. Compared to state-of-the-art methods, our approach demonstrated improved performance and reliability in complex scenarios. The utilization of splines and the velocity profiler highlighted the functionality and comfort along the nominal trajectory, while addressing lateral deviation through the integration of the MPC signal and appropriate operational limits. Our system consistently generated collision-free trajectories, setting it apart from existing approaches.

References

[1] S. Ulbrich and M. Maurer, "Probabilistic online pomdp decision making for lane changes in fully automated driving," in *16th International IEEE Conf. on Intel. Transp. Systems (ITSC 2013)*, pp. 2063–2067, 2013.

[2] T. Gu, J. M. Dolan, and J.-W. Lee, "On-road trajectory planning for general autonomous driving with enhanced tunability," in *Intelligent Autonomous Systems 13*, pp. 247–261, Springer Intern. Publish., 2016.

[3] W. Lim, S. Lee, M. Sunwoo, and K. Jo, "Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 613–626, 2018.

[4] M. Aoki, K. Honda, H. Okuda, and T. Suzuki, "Comparative study of prediction models for model predictive path- tracking control in wide driving speed range," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1261–1267, 2021.

[5] G. Frison, H. H. B. Sørensen, B. Dammann, and J. B. Jørgensen, "High-performance small-scale solvers for linear model predictive control," in *2014 European Control Conference (ECC)*, pp. 128–133, 2014.

[6] A. Li, Y. Jiang, X. Sun, H. Chi, C. Niu, and G. Liu, "Research status of intelligent electric vehicle trajectory planning and its key technologies: A review," *Electrochem*, vol. 3, no. 4, pp. 688–698, 2022.

[7] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.

[8] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, pp. 637–672, feb 2020.

[9] R. Gutiérrez, E. López-Guillén, L. M. Bergasa, R. Barea, Pérez, C. Gómez-Huélamo, F. Arango, J. del Egido, and J. López-Fernández, "A waypoint tracking controller for autonomous road vehicles using ros framework," *Sensors*, vol. 20, no. 14, 2020.

[10] F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla, "A systematic review of perception system and simulators for autonomous vehicles research," *Sensors*, vol. 19, no. 3, p. 648, 2019.

[11] H. Alghodhaifi and S. Lakshmanan, "Autonomous vehicle evaluation: A comprehensive survey on modeling and simulation approaches," *IEEE Access*, vol. 9, pp. 151531–151566, 2021.

[12] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

[13] R. Gutiérrez-Moreno, R. Barea, E. López-Guillén, F. Arango, N. Abdeselam, and L. M. Bergasa, "Hybrid decision making for autonomous driving in complex urban scenarios," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, 2023.

[14] P. Karle, M. Geisslinger, J. Betz, and M. Lienkamp, "Scenario understanding and motion prediction for autonomous vehicles—review and comparison," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 16962–16982, 2022.

[15] R. Lattarulo, D. He, and J. Pérez, "A linear model predictive planning approach for overtaking manoeuvres under possible collision circumstances," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1340–1345, 2018.

[16] M. Abe, *Vehicle handling dynamics: theory and application*. Butterworth-Heinemann, 2015.

[17] Y. Jiang, Z. Liu, D. Qian, H. Zuo, W. He, and J. Wang, "Robust online path planning for autonomous vehicle using sequential quadratic programming," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, pp. 175–182, 2022.

[18] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, pp. 2520–2525, 2011.

[19] Y. Zhang, H. Sun, J. Zhou, J. Pan, J. Hu, and J. Miao, "Optimal vehicle path planning using quadratic optimization for baidu apollo open platform," in *2020 IEEE Intel.Vehicles Symp.(IV)*, pp. 978–984, 2020.