Optimizing Vehicle Lane Change and Merge using Model Predictive Control

Shara Lawande
slawande@andrew.cmu.edu
Carnegie Mellon University

Saudamini Ghatge
sghatge@andrew.cmu.edu
Carnegie Mellon University

Shrudhi Ramesh Shanthi
srameshs@andrew.cmu.edu
Carnegie Mellon University

***Abstract*—When navigating in dynamic traffic environments, autonomous vehicles face fundamental challenges such as lane changing and merging maneuvers. This work aims to develop a model and control structure for autonomously merging a vehicle into traffic lanes where certain speed and spacing characteristics or constraints are assumed. The approach involves generating a simple model of the system dynamics, including simplified vehicle dynamics, and utilizing model predictive control MPC) techniques to provide control effort that allows for lane changing under quantifiable constraints. As part of our methodology, we developed a sophisticated nonlinear system dynamics model and formulated an optimization problem that incorporates strict constraints to ensure safety and efficiency of these maneuvers. Based on simulation results, we demonstrate the application of MPC for such problem statement. The code for our implementation can be found here.**

I. Introduction

Autonomous vehicles represent a transformative technology poised to revolutionize transportation systems by offering safer, more efficient, and intelligent mobility solutions. It is crucial for autonomous vehicles to be able to perform complex maneuvers, such as lane changes and merges, seamlessly in dynamic traffic environments. These maneuvers demand not only efficient trajectory planning but also robust decision-making to ensure passenger safety and traffic efficiency.

In traffic scenarios that involve lane changes and merging maneuvers, autonomous vehicles face unique challenges caused by the dynamic nature of traffic, in which other vehicles exhibit varying behaviors and trajectories. Traditionally, trajectory planning has struggled to address these challenges, particularly in scenarios with multiple unknown or unpredictable agents.

In this context, Model Predictive Control (MPC) emerges as a promising methodology for optimizing vehicle maneuvers by leveraging predictive models and dynamic optimization techniques. The core principle of MPC involves formulating an optimization problem that predicts future system behavior over a finite time horizon and generates control actions to optimize a specified objective function while adhering to system constraints.

This paper introduces a novel application of MPC tailored specifically for optimizing vehicle lane change and merge behaviors in dynamic traffic scenarios as shown in Fig: 1 Fig: 2. Our approach addresses the following key aspects:

1) *System Dynamics Modeling:* We develop a sophisticated nonlinear system dynamics model that accurately represents the vehicle's behavior during lane change and merge maneuvers. This model accounts for critical factors such as vehicle dynamics, control inputs, and environmental conditions.
2) *Trajectory Optimization Problem:* We formulate a trajectory optimization problem within an MPC framework to generate optimal control actions that facilitate safe and efficient lane changes and merges. Our formulation incorporates stringent constraints to ensure collision avoidance and adherence to traffic regulations.
3) *Integration of MPC Frameworks:* We review and analyze various MPC frameworks and game-theoretic methods proposed in recent literature to address similar challenges.
4) *Simulation and Validation:* Through extensive simulation studies, we demonstrate the efficacy of our MPC-based approach in achieving seamless and collision-free lane changes amidst complex multi-agent traffic scenarios. Simulation results validate the effectiveness of our methodology under diverse traffic conditions and interaction scenarios.

The contributions of this work lie in the development of a practical and effective MPC-based strategy for optimizing autonomous vehicle lane change and merge behaviors. By integrating sophisticated system modeling, optimization techniques, and insights from recent research, our approach represents a significant step towards enhancing the autonomy and safety of next-generation vehicles in dynamic traffic environments.

The remainder of this paper is organized as follows: Section II provides a comprehensive review of related work in MPC for vehicle interaction scenarios, highlighting key modeling and optimization techniques. In Section III, we detail our methodology, including problem formulation, system dynamics modeling, and constraint definition. Section IV presents simulation results and performance evaluation of our approach, followed by conclusions and directions for future research in Section V.

II. Related Works

A. Different Model Predictive Frameworks

Literature has many variations of applying MPC to this problem of interaction of ego vehicle with other agents. Yuxiao

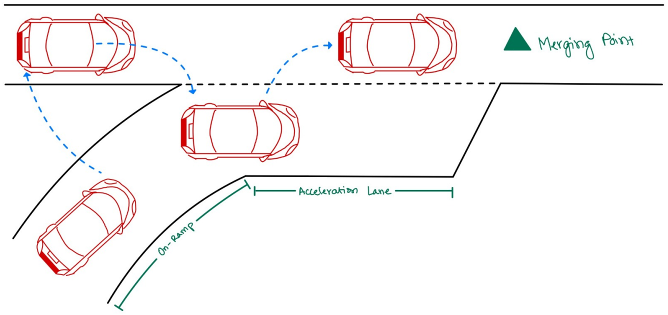

Fig. 1. Merging in highway

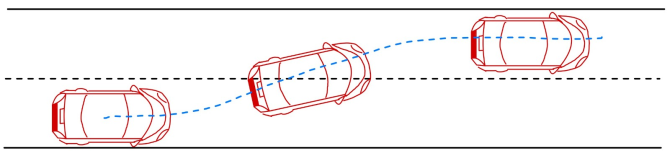

Fig. 2. Changing Lanes

Chen et al., propose a branch Model Predictive Framework [4] which plans over feedback policies to leverage the reactive behaviour of the unknown agent. Typical algorithms first design a predictive model and then the motion planning algorithm plans the motions in a way that the autonomous agent avoids the reachable set of the unknown agent [5]. But this leads to conservative motion plans because the motion planner ignores the reactive nature of the unknown agent within the prediction horizon, which basically assumes a fixed reactive behaviour of the unknown agent. This is not ideal since the behaviour is nondeterministic and the planner needs to account for this uncertain behaviour. Their proposed solution combines the continuous motion planning with the discrete modes which represent the unpredictable behaviour of the unknown agent. This algorithm claims to capture the uncertain behaviour of the uncontrolled agent and plan appropriate trajectories. Although this algorithm does not perform well if the number of unknown agents are increased.

Lixing Huang et al. [6], proposed a receding horizon control which parameterized the system trajectory with the control input and employs a nonlinear model on the ego vehicle dynamics. To model the uncertain behaviour of the unknown agent, they proposed a probabilistic model along with a collision avoidance model. But their method assumes that the ego vehicle always follows a reference center line and does not consider vehicles which are not in the sensing range of the ego vehicle.

B. Game Theoretic Methods

David Fridovich-Keil et al. [7], proposes a Differential Game setting for solving the problem of multiple agents interacting with each other. But since most numerical solution techniques are not suitable for real-time applications when the state dimensions are increased, and its a common practice to predict the future actions of other agents and solve the decoupled optimal control problem for each agent. They propose to use Iterative Linear Quadratic Regulator (ILQR), which solves repeated convex function approximations. Based on a Monte Carlo study, their solution demonstrated the ability of the algorithm to identify complex interactive strategies for multi agent interaction. However they fail to provide convergence guarantee if arbitrary initial conditions are set.

Another work similar to above paper, in terms of solving the problem of interacting with multiple agents was proposed by Simon Le Cleac'h et al. [8]. The difference lies in the method to solve the optimisation problem. The algorithm makes use of Augmented Lagrangian method to enforce constraints. Based on a Monte Carlo Analysis, the algorithm shows the ability to solve the optimization problem on nonlinear and non-convex constraints.

III. Methodology

A. Problem Formulation

We are solving a discretized trajectory-optimization problem with N time steps. The problem is characterized by nonlinear system dynamics where the side-slip parameter of the vehicle represents the non-linearity.
We model the dynamics of the car with a simple nonlinear bicycle model [2], with the following states and controls -

$$x = \begin{bmatrix} p_x \\ p_y \\ \theta \\ \delta \\ v \end{bmatrix}, \quad u = \begin{bmatrix} a \\ \dot{\delta} \end{bmatrix} \tag{1}$$

where, p_x, p_y are the 2D position of the vehicle, θ is the orientation, δ is the steering angle and v is the velocity. The control inputs are a which is the acceleration and $\dot{\delta}$ which is the rate of steering angle.
We define the non-linear continuous time model as -

$$\dot{x} = f(x, u) \tag{2}$$

We will linearize the non-linear dynamics along a reference trajectory and approximate the nonlinear system locally using Taylor Series expansion. This linearization is justified by assuming that while merging vehicles maintain a set velocity. We will define a reference trajectory along which we will linearize the system. Before linearizing, we will discretize the system using the 4th order Runge-Kutta Method [1], where -

$$\begin{aligned} k1 &= f(x,u) \\ k2 &= f(x + k1/2, u) \\ k3 &= f(x + k2/2, u) \\ k4 &= f(x + k3, u) \\ x &= x + \frac{1}{6} * (k1 + 2*k2 + 2*k3 + k4) \end{aligned} \tag{3}$$

The reference trajectory used for linearizing the system is defined as the trajectory of the obstacle vehicle in the second lane, which the ego vehicle needs to merge into, while avoiding collisions with the obstacle vehicles. The ego vehicle follows this reference center trajectory and follows all the control bound constraints which lets the ego vehicle stay in lane after merging into the target lane.

For our optimization problem, we are defining a Quadratic cost function which penalized control inputs and the distance between the current state and the desired state.

$$\begin{aligned} \min_{x_{1:N}, u_{1:N-1}} \quad & \sum_{i=1}^{N-1} \left[\frac{1}{2}(x_i - \tilde{x}_{\text{ref},i})^T Q (x_i - \tilde{x}_{\text{ref},i}) + \frac{1}{2} u_i^T R u_i \right] \\ & + \frac{1}{2}(x_N - \tilde{x}_{\text{ref},N})^T Q_f (x_N - \tilde{x}_{\text{ref},N}) \\ \text{st} \quad & x_1 = x_{\text{IC}} \\ & x_N = x_{\text{g}} \\ & x_{i+1} = Ax_i + Bu_i, \quad \text{for } i = 1, 2, \ldots, N-1 \\ & [-1, -1] \leq u_i \leq [1, 1], \quad \text{for } i = 1, 2, \ldots, N-1 \\ & \| x_i^{ego}[1,2] - x_i^{obs}[1,2] \| \geq 6 \quad \text{for } i = 1, \ldots, N \\ & 8 \geq x_i^{ego}[2] \geq -2.5 \quad \text{for } i = 1, 2, \ldots, N \\ & x_i^{ego}[5] > x_i^{obs}[5] \quad \text{for } i = 1, 2, \ldots, N \end{aligned} \tag{4}$$

The constraints for our problem statement are defined as follows -

1) Initial and Goal state condition constraints
2) Dynamics constraints
3) Control bounds constraints
4) Collision avoidance constraints
5) Staying in Lane constraints
6) Velocity constraints

1) *Initial and Goal state condition constraints*:
   - The initial state x_1 of the vehicle must match a specified initial condition x_{IC}.
   - The final state x_N of the vehicle must reach a desired goal condition x_{g}.
2) *Dynamics Constraints*:
   - The state transition at each time step is governed by a linear model that applies to the current state and control input.
3) *Control Bounds Constraints*:
   - The control inputs u_i are constrained within $[-1, -1]$ and $[1, 1]$, to ensure realistic and feasible control actions.
4) *Collision Avoidance Constraint*:
   - This constraint maintains a safe distance between the ego vehicle and any obstacle vehicle to prevent collisions. The minimum distance requirement is set to be twice the length of the ego vehicle.
5) *Lane Constraints*:
   - The lateral position (y-coordinate) of the ego vehicle is restricted to stay within the defined lane boundaries. This constraint ensures that the vehicle performs lane changes or merges appropriately without deviating from the intended path.
6) *Velocity Constraint*:
   - The velocity of the ego vehicle must be maintained at a level higher than that of any obstacle vehicle encountered during the trajectory. This constraint helps in that the ego vehicle will perform the lane change motion and arrive at the desired end position before the obstacle vehicle.

These constraints collectively define the boundaries and conditions under which the trajectory optimization problem is solved using Model Predictive Control (MPC). By integrating these constraints, the optimization process aims to generate safe and optimal control inputs for autonomous vehicle operations in dynamic traffic environments. The cost function associated with this optimization problem is quadratic, and the constraints are linear, making the entire optimization problem convex. Leveraging convex optimization enables efficient and reliable computation of control inputs, ensuring that the autonomous vehicle adheres to safety and performance requirements while navigating through complex traffic scenarios.

B. Model Predictive Control

Model Predictive Control (MPC) is an optimal control strategy to select control inputs based on minimising some objective function. One of the important key features of MPC is its ability to handle constraints directly, including constraints on the control inputs, outputs and internal states [3]. A convex MPC controller sets up and solves a convex optimization problem at each time step that incorporates the current state estimates as an initial condition. MPC reasons about the next N steps instead of the whole trajectory. This allows the model to compensate for any changes in the model and accordingly update its control strategy. MPC algorithm is as follows -

- Initialize the control position x_{IC}
- Solve a trajectory optimization problem looking at the next N steps
- Execute the first control u_1 from this problem on the system
- Repeat this whole process at the next time step

Thus, at each time t, given the state of the system $x(t)$, the action is determined by solving the following Finite Time Optimal Control Problem (FTOCP):

$$
\begin{aligned}
\min_{u_0,\ldots,u_{N-1}} \quad & \sum_{k=0}^{N-1} h(x_k, u_k) + V(x_N) \\
\text{s.t.} \quad & x_{k+1} = f(x_k, u_k), \quad \forall k \in \{0, \ldots, N-1\}, \\
& x_k \in \mathcal{X}, \quad u_k \in \mathcal{U}, \quad \forall k \in \{0, \ldots, N-1\}, \\
& x_0 = x(t), \\
& x_N \in \mathcal{X}_N,
\end{aligned} \tag{5}
$$

IV. RESULTS

We have simulated the optimization problem in `julia`, and used `Convex` library to solve the optimization problem. We simulated the solver for 20 seconds, with 100 time steps. We have used the `MeshCat` library to animate and visualize the lane changing motion.
For the purpose of this project, we have simulated the obstacle vehicle with a constant velocity, without any acceleration.
The following are the results we have obtained by performing convex trajectory optimization on our objective function and constraints to generate the states and controls for the ego vehicle for the following initial and final conditions:

$$
\begin{aligned}
x_{IC}^{ego} = [0,4,0,0,0] \; x_g^{ego} = [43,2,0,0,0] \\
x_{IC}^{obs} = [0,0,0,0,0] \; x_g^{obs} = [40,0,0,0,0]
\end{aligned} \tag{6}
$$

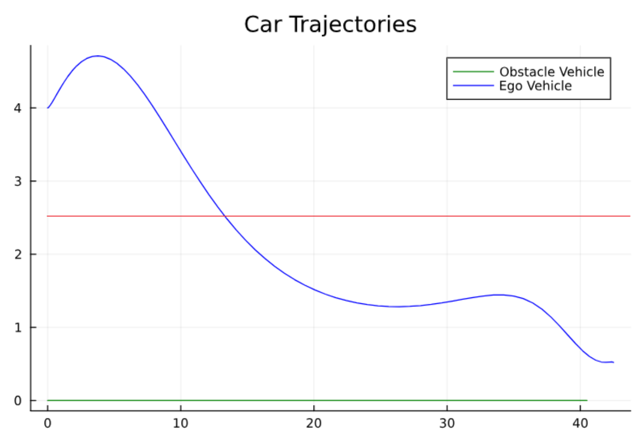

Fig. 3. Trajectories of the ego and obstacle vehicles

The full video of the simulation can be found here.
As seen in figures 3, 4, 5, 6, 7, and 8, by using the controls directly generated by the convex solver, the ego vehicle (red car) can change lanes without colliding with the obstacle vehicle (grey car) seamlessly, and while maintaining a higher velocity than the obstacle vehicle.
The results we get on simulating the control variables from convex trajectory optimization with a model predictive control loop, for the same initial and final conditions of the ego and obstacle vehicle, and an MPC sliding window of $N\_mpc = 20$, are shown in figures 9, and 10

As we can see from figures, 11, 12, 13, and 14, with the same constraints and objective function as before,the ego

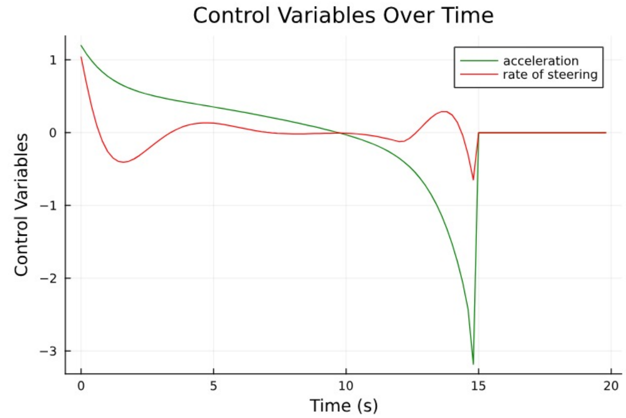

Fig. 4. Control Variables against time

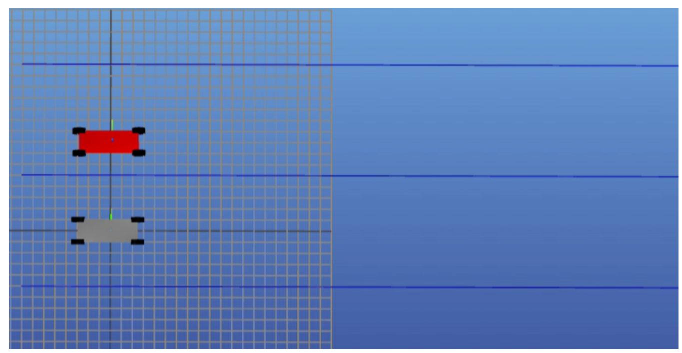

Fig. 5. Snippets of the simulation

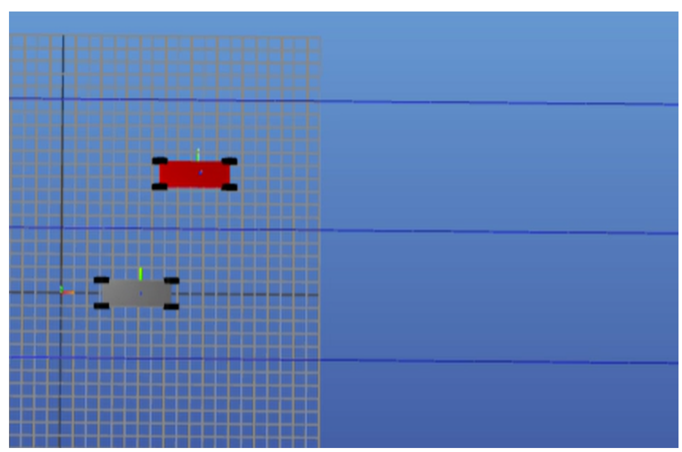

Fig. 6. Snippets of the simulation

vehicle has gained a lot more speed and is drifting forward. One of the reasons this could be happening is due to the iterative re-computation done in the MPC algorithm, which means it continuously recalculates trajectories for different horizons.
The full video of the implementation can be found here

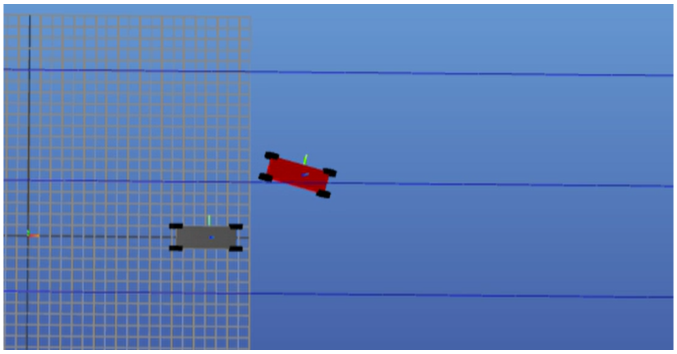

Fig. 7. Snippets of the simulation

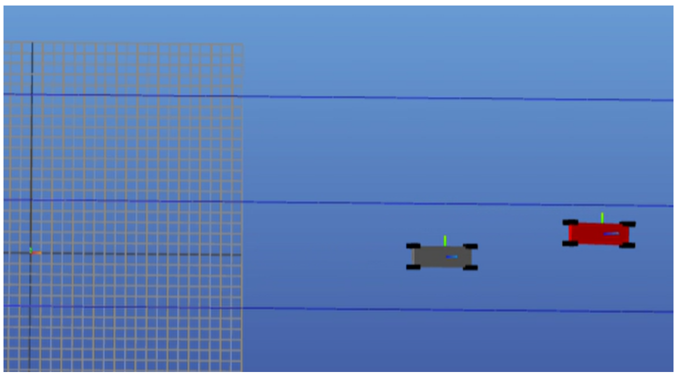

Fig. 8. Snippets of the simulation

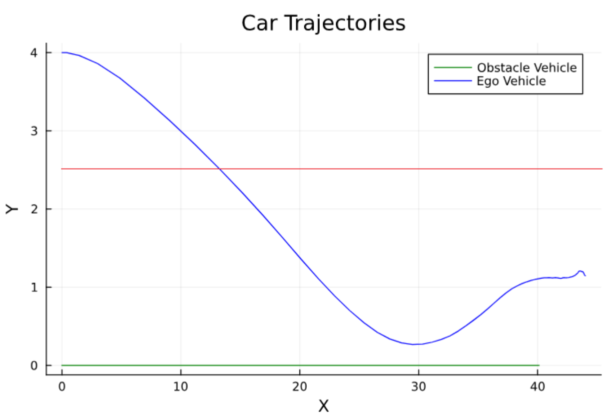

Fig. 9. Trajectories of the ego and obstacle vehicles using MPC

V. Conclusion

In this work, we have presented an application of Model Predictive Control (MPC) tailored specifically for optimizing autonomous vehicle lane change and merge maneuvers in dynamic traffic scenarios. Our approach leverages sophisticated system dynamics modeling and constraint-based trajectory optimization within an MPC framework to achieve safe and efficient vehicle behaviors. By formulating and solving a convex optimization problem alone with MPC, we demonstrated how MPC affects the behaviour of the system through extensive simulation studies.

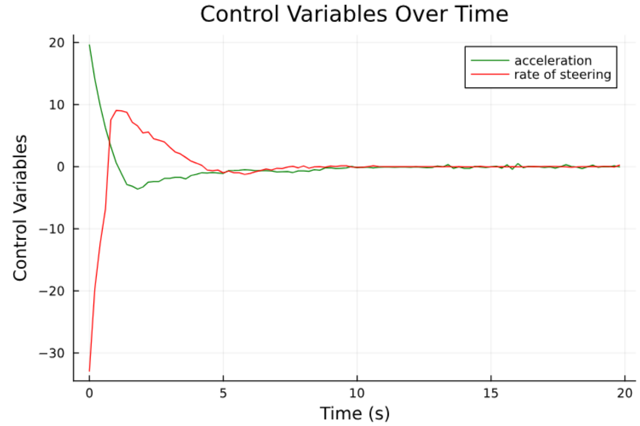

Fig. 10. Control Variables against time using MPC

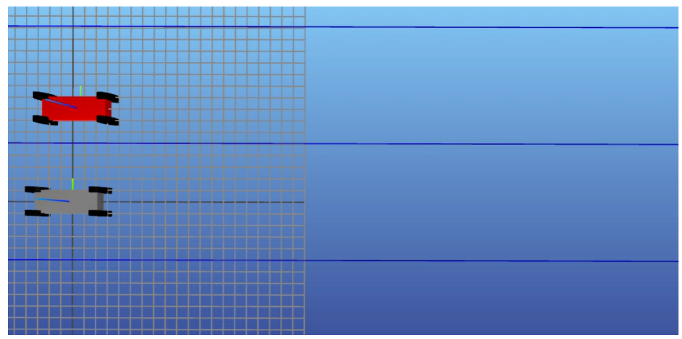

Fig. 11. Snippets of the simulation for MPC

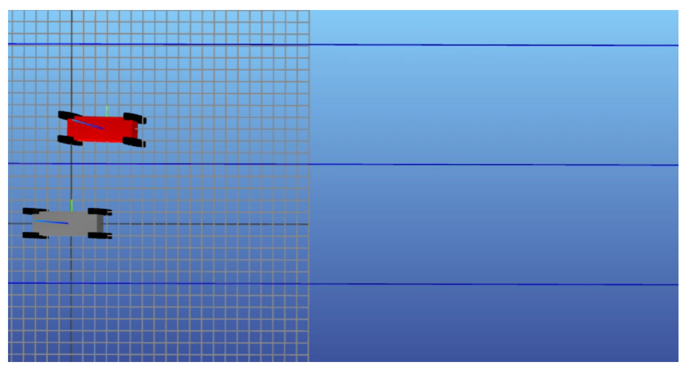

Fig. 12. Snippets of the simulation for MPC

The results indicate that without using MPC and generating a single control strategy for the entire strategy, we were able to perform seamless lane changes while adhering to stringent safety and performance constraints. After implementing MPC, our performance was affected. One of the reasons could be because of defining less constraints.

In conclusion, our study underscores the potential of MPC as a viable solution for optimizing autonomous vehicle maneuvers, paving the way for safer and more efficient transportation systems in the era of intelligent mobility.

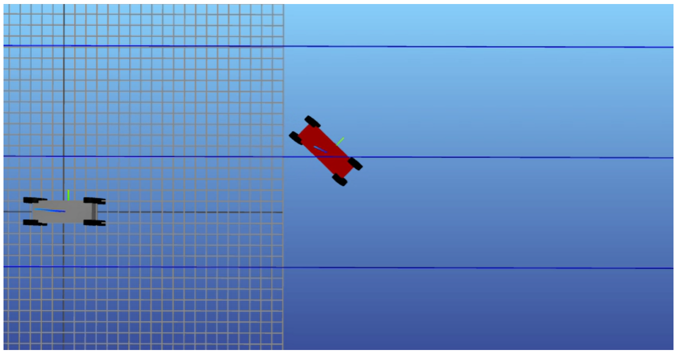

Fig. 13. Snippets of the simulation for MPC

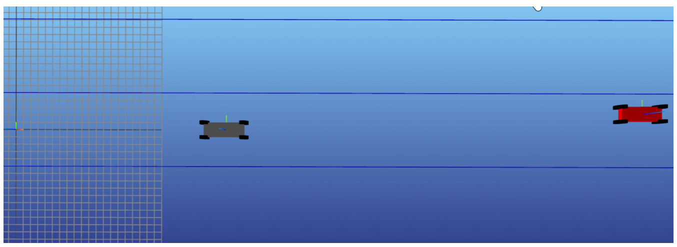

Fig. 14. Snippets of the simulation for MPC

VI. Future Work

Based on our results, we can improve the performance of MPC by introducing more constraints on the vehicle. Our current implementation can control the ego vehicle to merge into the second lane, but does not predict the behaviour of the other vehicles. This could be an interesting avenue to look into since in a real-world the behaviour of the other vehicles cannot be known in advance, making the problem more complex to solve. Researchers have implemented various strategies to face this challenge which include using game theory methods, a probabilistic model to predict the trajectories of the obstacle vehicles, etc. Another possible direction of future work is increasing the number of obstacle vehicles with different behaviours.

Acknowledgment

We express our profound appreciation to the Robotics Institute at Carnegie Mellon University for providing the necessary resources and environment for this research. We are particularly grateful to Prof. Zachary Manchester, the instructor of the 16745 Optimal Control and Reinforcement Learning course, whose expert guidance was essential during this project's development. Additionally, we thank our teaching assistants, for their dedicated support and valuable feedback, which were critical to our success. Their efforts were key in addressing the challenges of designing and implementing robotic systems.

References

[1] https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods
[2] Kong, Jason, et al. "Kinematic and dynamic vehicle models for autonomous driving control design." 2015 IEEE intelligent vehicles symposium (IV). IEEE, 2015.
[3] https://cse.sc.edu/ gatzke/cache/npc-Chapter5-scan.pdf
[4] https://doi.org/10.48550/arXiv.2109.05128
[5] Kousik, Shreyas, et al. "Safe trajectory synthesis for autonomous driving in unforeseen environments." Dynamic systems and control conference. Vol. 58271. American Society of Mechanical Engineers, 2017.
[6] https://websites.umich.edu/ dpanagou/assets/documents/LHuang_ACC17.pdf
[7] https://doi.org/10.48550/arXiv.1909.04694
[8] https://doi.org/10.48550/arXiv.1910.09713