Experimental Validation of Event-Triggered Model Predictive Control for Autonomous Vehicle Path Tracking

Zhaodong Zhou and Jun Chen*, *Senior Member, IEEE*
Department of Electrical and Computer Engineering
Oakland University
Rochester, MI 48309, USA
Email: {zhaodongzhou,junchen}@oakland.edu

Mingyuan Tao, Peng Zhang and Meng Xu
Isuzu Technical Center of America, Inc
Plymouth, MI 48309, USA
Email: {Mingyuan.Tao,Peng.Zhang,Meng.Xu}@isuzu.com

***Abstract*—This paper presents an experimental validation of an event-triggered model predictive control (MPC) for autonomous vehicle (AV) path-tracking control using real-world testing. Path tracking is a critical aspect of AV control, and MPC is a popular control method for this task. However, traditional MPC requires extensive computational resources to solve real-time optimization problems, which can be challenging to implement in the real world. To address this issue, event-triggered MPC, which only solves the optimization problem when a triggering event occurs, has been proposed in the literature to reduce computational requirements. This paper then conducts experimental validation, where event-triggered MPC is compared to traditional time-triggered MPC through real-world testing, and the results demonstrate that the event-triggered MPC method not only offers a significant reduction in computation compared to time-triggered MPC but also improves the control performance.**

I. Introduction

As electric vehicles have gained popularity in recent years, there is increasing interest in autonomous driving as a promising technology to enhance traffic efficiency while reducing accidents and congestion [1]. Autonomous vehicles (AVs) rely on a complex network of sensors, algorithms, and control systems to navigate the road safely and efficiently. Model predictive control (MPC) is a class of algorithms that is well suited for AVs, as it can handle complex optimization problems and constraints [2]–[4]. Through the study of stability, robustness, and feasibility in MPC, as demonstrated by various research papers [5]–[7], the stability of MPC applications in AVs has been further ensured.

Despite its advantages, model predictive control (MPC) has some limitations. One of the main challenges is the computational burden it imposes, which can be particularly problematic for AVs due to their limited computing power. To address this issue, event-triggered control has been proposed with the goal of reducing computation [8]–[13]. Unlike time-triggered MPC, where MPC activates periodically, event-triggered MPC is an approach in which the optimization problem is solved when a triggering event occurs. To demonstrate the benefits of event-triggered MPC in the field of AV, particularly in terms of its effectiveness in reducing computational requirements, many researchers have used simulation software to simulate the performance of event-triggered MPC on AVs [14]–[20]. In [14], event-triggered MPC is implemented for multi-vehicle control, achieving simultaneous tracking with collision and obstacle avoidance. Another study [15] applies event-triggered MPC to vehicle-following control with unreliable vehicle-to-vehicle communications. The problem of multiple vehicle cooperative path following is explored in [17], [18], while [16], [19] investigate vehicle platooning and employ MPC for longitudinal control to track inter-vehicle distance. Nonlinear MPC is used in [20] for lateral trajectory tracking of AVs, which enhances real-time performance while maintaining accuracy. Previous works by the authors [21]–[23] also propose event-triggered MPC and LPV-MPC for AV path tracking problems, all utilizing event-triggered MPC to reduce the computational burden. These studies demonstrate the accuracy and computational efficiency of event-triggered MPC in various application scenarios.

While the benefits of event-triggered MPC have been demonstrated in simulation environments, the validation of this approach in realistic settings is still limited. In particular, there is a need to demonstrate the effectiveness of event-triggered MPC for AV path tracking, as this is a critical component of AV control systems. To address this need, we present an experimental validation of event-triggered MPC for AV path tracking, by using a real-world testing platform to show the advantages of event-triggered MPC over time-triggered MPC. Specifically, the testing vehicle is a full-size sedan equipped with a drive-by-wire system and a Polynav 2000P OME GNSS-Inertial system, a Calmcar front view camera with lane detection capabilities, and a Dspace Autera computing unit. The GNSS unit provides real-time vehicle location information that is consumed by control systems. Moreover, a predefined path is recorded using GNSS and serves as a reference trajectory for subsequent testing. Both time-triggered and event-triggered MPC are tested, and it is found that event-

This work is supported in part by SECS Faculty Startup Fund from Oakland University.

*Jun Chen is the corresponding author.

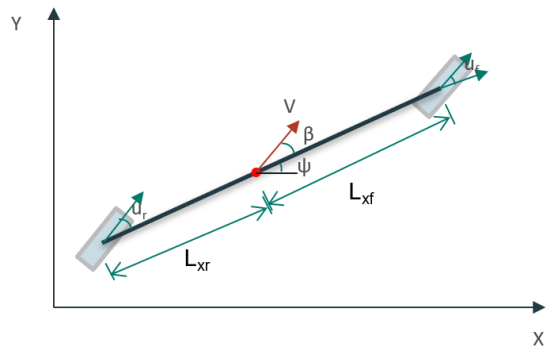

Fig. 1. Schematic of the bicycle model.

triggered MPC offers a significant reduction in computation compared to time-triggered MPC, which at the same time improves the control performance. The improvement in control performance is likely due to the fact that when optimization is not triggered, event-triggered MPC can provide real-time control action without any delay.

The remainder of this paper is organized as follows. Section II discusses the vehicle model in the MPC and the algorithm of time-triggered MPC and event-triggered MPC. The vehicle platform setup and the test result are shown in Section III. Section IV concludes the paper.

II. MPC-Based Path Tracking

A. Vehicle Model

In normal on-road driving, which is the focus of this paper, the vehicle dynamics can be conveniently approximated by the bicycle model [24] shown in Fig. 1. Define $x = \begin{bmatrix} p_x & p_y & \psi \end{bmatrix}^T$ as the state vector for the vehicle model at the center of gravity (CG), where p_x and p_y are the vehicle longitudinal and lateral positions, respectively, and ψ is the vehicle heading angle, all the states are in the vehicle frame. Then $\dot{x} = \begin{bmatrix} \dot{p}_x & \dot{p}_y & \dot{\psi} \end{bmatrix}^T$, the set of differential equations of the vehicle is modeled by the following,

$$\dot{p}_x = V\cos(\psi + \beta) \tag{1a}$$
$$\dot{p}_y = V\sin(\psi + \beta) \tag{1b}$$
$$\dot{\psi} = \frac{V\cos(\beta)}{L_{xf} + L_{xr}}(\tan(u_f) - \tan(u_r)), \tag{1c}$$

where V is the velocity of the vehicle's CG; L_{xf} and L_{xr} are the distances from the center of gravity to the front and rear axles; u_f and u_r are the front and rear steering angles. Since the vehicle used in this paper is the front-wheel steering vehicle, the u_r is equal to zero. Furthermore, β is the vehicle slip angle, which is defined by the following equation.

$$\beta = \arctan\left(\frac{L_{xr}\tan(u_f)}{L_{xf} + L_{xr}}\right). \tag{2}$$

B. Time-Triggered MPC for Path Tracking

The MPC controller designed here has to track a desired path, and the vehicle model described in Section II-A is employed to make predictions in the MPC algorithm. To make use of the bicycle model by MPC, the forward Euler method [25] is used to discretize the bicycle model,

$$x_{t+1} = x_t + \dot{x}_t T_s, \tag{3}$$

where T_s is the sampling time and x_t is the system state at discrete time t. For MPC-based path tracking control, at time instance t, the general MPC algorithm performs the following operations. Initially, it measures the current state of the system. Subsequently, it solves an optimal control problem formulated on the system model, constraints, and current state to find the optimal state sequence $X_t = \{x_{t+1}, x_{t+2}, \ldots, x_{t+p}\}$ and the optimal control sequence $U_t = \{u_t, u_{t+1}, \ldots, u_{t+p-1}\}$, where p is the prediction horizon. Lastly, it sends the first element of the optimal control sequence to the actuators. The optimal control problem (OCP) is formulated as follows,

$$\begin{aligned}\min J = &\sum_{k=1}^{p} \left|\left|x_{t+k}(1) - p^{ref}_{x,t+k}\right|\right|^2_{Q_p} \\ &+ \sum_{k=1}^{p} \left|\left|x_{t+k}(2) - p^{ref}_{y,t+k}\right|\right|^2_{Q_p} + \sum_{k=0}^{p-1} ||u_{t+k}||^2_{Q_u} \\ &+ \sum_{k=0}^{p-1} ||u_{t+k} - u_{t+k-1}||^2_{Q_d} \end{aligned} \tag{4a}$$
$$\text{s.t.} \quad x_t = \hat{x}_t \tag{4b}$$
$$\text{System dynamics (3)}, \quad 1 \le k \le p \tag{4c}$$
$$u_{min} \le u_{t+k} \le u_{max}, \quad 0 \le k \le p-1 \tag{4d}$$
$$\Delta_{min} \le u_{t+k} - u_{t+k-1} \le \Delta_{max}, \quad 0 \le k \le p-1 \tag{4e}$$

Note that the first two terms in equation (4a) denote the deviation from the reference path, while the third term penalizes a high steering angle, and the final term decreases the amount of actuator busyness (i.e., limits the rate of actuator change). The weights for path following error, steering efforts, and control activity are represented by Q_p, Q_u, and Q_d respectively.

Remark 1: To derive a reference path suitable for MPC, the default path needs to rearrange due to varying intervals between consecutive waypoints. As a result, the waypoints on the default path should be re-sampled based on the current vehicle speed v and sampling time T_s. As MPC depends on short-term prediction, it is sensible to presume that V will remain constant over the prediction horizon. Therefore, the waypoints throughout the prediction horizon are re-sampled so that they are equally distanced from each other, where the distance d is equal to current vehicle speed v multiplied by sampling time T_s.

C. Event-Triggered MPC for Path Tracking

Time-triggered MPC can be computationally heavy since the OCP (4) needs to be solved at every time step. Unlike time-triggered MPC, event-triggered MPC solves the OCP (4) only when an event is triggered. This paper considers the threshold-based event-trigger mechanism adopted by [21], [26]. Since

TABLE I
MPC PARAMETERS.

p	10	Q_u	35	$u_{min}\ (rad)$	-0.97
$T_s\ (ms)$	200	Q_d	30	$\Delta u_{max}\ (rad)$	0.15
Q_P	2	$u_{max}\ (rad)$	0.97	$\Delta u_{min}\ (rad)$	-0.15

MPC does not control the longitudinal speed, the lateral offset Y is primarily considered when determining an event, which is the closest distance from the current position to the target path. Denote the closest points on the path before and after the current position as (x_1, y_1) and (x_2, y_2). Then the lateral offset Y is calculated by equation (5),

$$Y = \frac{|(x_2 - x_1)(y_1 - p_y) - (x_1 - p_x)(y_2 - y_1)|}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}, \quad (5)$$

Finally, the event-trigger mechanism used in this paper is shown below.

$$e = \begin{cases} 1 & \text{if } Y > \sigma \text{ or } k > k_{max} \\ 0 & \text{Otherwise} \end{cases}. \quad (6)$$

In other words, the condition at which the event-triggered MPC is triggered depends on two calibration parameters: σ and k_{max}, where k represents the number of consecutive times that the MPC has not been triggered. It is important to note that k_{max} should not exceed the prediction horizon p. The event-triggered MPC solves the OCP (4) only when either the vehicle's lateral offset exceeds a predefined threshold σ (i.e., $Y > \sigma$), or the previously optimized control sequence U_{t_1} has been depleted (i.e., $k > k_{max}$), resulting in $e = 1$. Otherwise, $e = 0$, and the control action can be obtained by shifting the optimal sequence obtained during the last event.

III. EXPERIMENTAL SETUP AND RESULT ANALYSIS

The controller discussed in Section II is evaluated in the experimental test. To verify the efficiency of the event-triggered MPC, both time-triggered MPC and event-triggered MPC are implemented in the vehicle platform separately, and are used to track the vehicle to follow a same reference trajectory.

A. Experimental Setup

In this paper, a relatively empty site is used as the experimental testing track, as shown in Fig. 2, and the AV platform is shown in Fig. 3. Specifically, the testing vehicle is a full-size sedan, the distances from the CG to the front and rear axles are 1.2m and 1.65m, respectively. This vehicle is equipped with drive-by-wire systems, a Polynav 2000P GNSS-inertial system, a Calmcar front view camera with lane detection capabilities, and a Dspace Autera computing unit.

Meanwhile, to compare the performance of time-triggered MPC and event-triggered MPC, the MPC parameters, including cost function calibrations, actuator bound constraints, and rate constraints are maintained the same. Table I lists all the parameters for both time-triggered and event-triggered MPC.

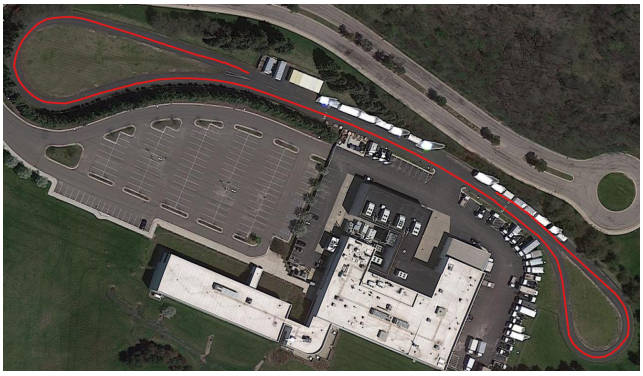

Fig. 2. Bird view of the testing track located in Plymouth MI.

Fig. 3. The AV testing platform, which is a full-size sedan equipped with a drive-by-wire system, a Polynav 2000P GNSS-inertial system, a Calmcar front view camera with lane detection capabilities, and a Dspace Autera computing unit.

B. Numerical Result

In the sequel, we denote time-triggered MPC as tMPC and event-triggered MPC as eMPC(σ) where σ represents the event-triggered threshold in (6). Additionally, the performances of event-triggered MPC with various σ values are examined to investigate the impact of the event-triggered threshold on eMPC control performance.

The reference trajectory and tracking results with different MPC are shown in Figs. 4 and 5. In general, all the controllers can control the vehicle to complete the entire path safely, and the tracking performances of each MPC are satisfactory. To compare the performance of all controllers, tracking errors for all controllers are plotted in Fig. 6, together with the maximum error and root mean square error, both on the lateral tracking error, being shown in Table II. Based on Table II, tMPC has the worst tracking performance, with both Max Error and RMSE being larger than eMPC. For eMPC, the Maximum Error and RMSE are similar with different thresholds. Comparing the peaks in Fig. 6, it is clear that the large error occurs at the starting and ending areas. Meanwhile, combined with Fig. 4, it can be seen that the area with a large error is the turning region at the beginning and end. This is further supported by zoom-in pictures of the starting and ending turns in Fig. 5.

In Table III, the number of MPC triggers and the trigger frequency of eMPC are used to compare the extent to which eMPC reduces the amount of computation compared to tMPC.

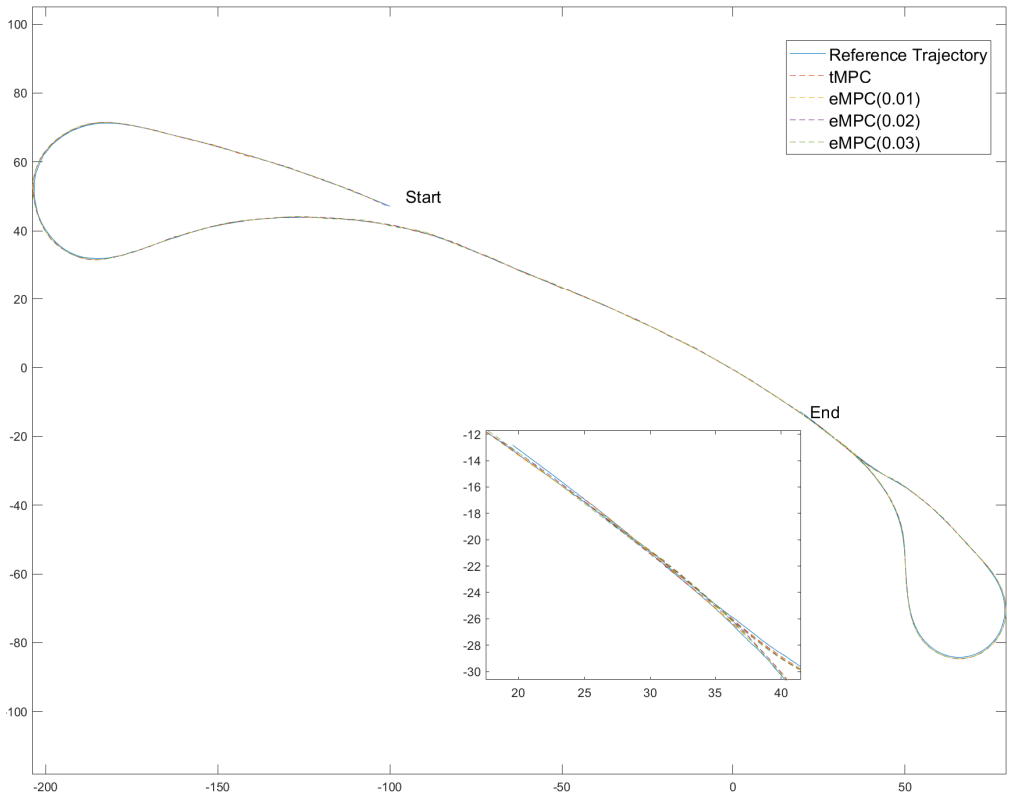

Fig. 4. Tracking trajectories with different controllers.

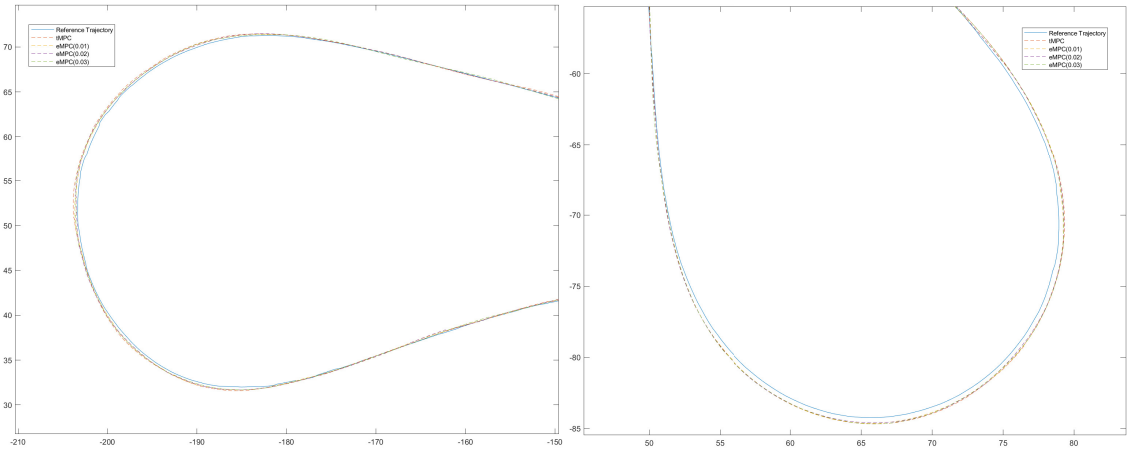

Fig. 5. Tracking trajectories with different controllers during the turn maneuvers (i.e., zoom-in version of Fig. 4).

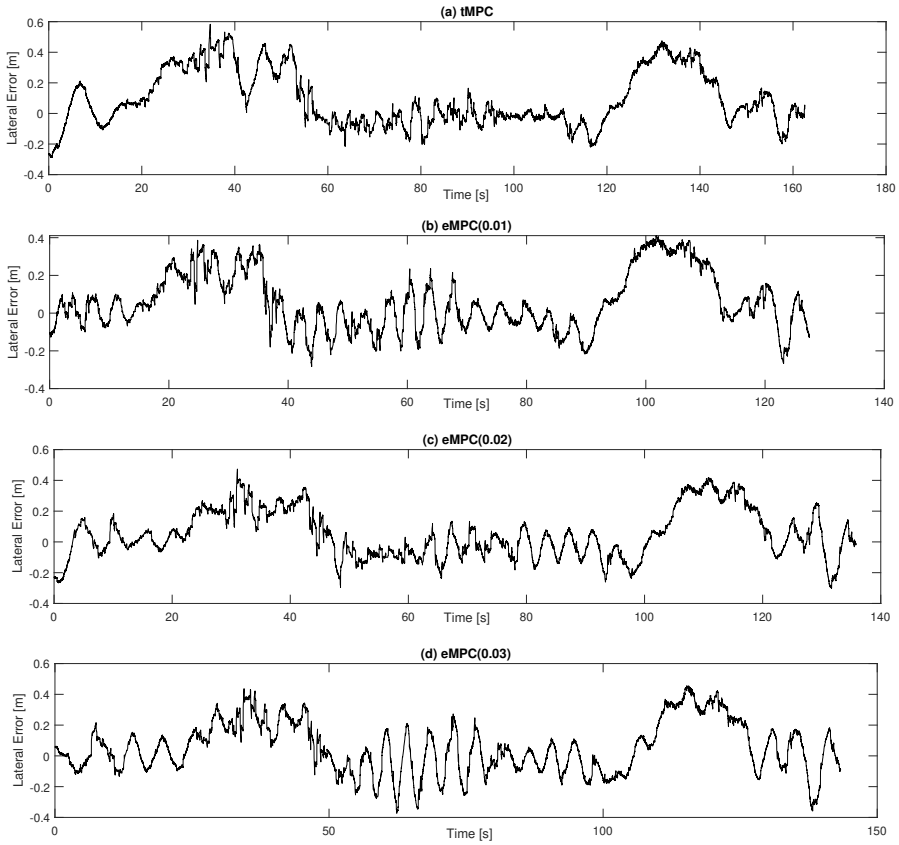

Fig. 6. Tracking errors with different controllers.

TABLE II
TRACKING PERFORMANCE WITH DIFFERENT CONTROLLERS.

	tMPC	eMPC(0.01)	eMPC(0.02)	eMPC(0.03)
Max Error (m)	0.5833	0.4108	0.4736	0.4559
RMSE (m)	0.1386	0.1056	0.1030	0.1058

Comparing the tracking performance within the three event-triggered MPC settings, as summarized in Tables II and III, it is apparent that as the event-trigger threshold increases, both the maximum error and RMSE almost equal, while the MPC trigger frequency decreases. However, based on the simulation result from the authors' previous work [21], the tracking accuracy of the time-triggered MPC is better than the event-triggered MPC. In addition, the tracking performance of event-triggered MPC decreases with the threshold increase. The result in this paper is different from the simulation result because of the delay caused by MPC computation. According to the driving time and control times in time-triggered MPC, it can be estimated that the average MPC calculation takes 75 ms. Specifically, there is a 75 ms gap from starting the MPC computation to sending the control signal to the vehicle. However, in the time-triggered MPC, the optimal steering control sent to the vehicle is based on the initial state, which can change significantly after 75 ms. Therefore, the optimal steering control that the time-triggered MPC sent is late. The MPC trigger frequencies of three event-triggered MPC settings are around 50%, indicating that on average the event is not triggered once every two control steps at which MPC needs to calculate the optimal control sequence. Note that when an event is not triggered, event-triggered MPC simply shifts the previous optimal control sequence to determine control action for the current step, which requires very minimum or negligible delay. Therefore, it is our conjecture this provides a timely compensation to the delay caused by MPC computation when an event is triggered and explains why event-triggered MPC outperforms time-triggered MPC in our experimental results.

IV. CONCLUSION

This paper implements the time-triggered model predictive control (MPC) and event-triggered MPC to solve a vehicle

TABLE III
COMPUTATION NEEDED WITH DIFFERENT CONTROLLERS.

	tMPC	eMPC(0.01)	eMPC(0.02)	eMPC(0.03)
Control Counts	2146	2581	3160	3894
Event Counts	2146	1789	1847	1870
Trigger Frequency (%)	100	69.31	58.45	48.02
Driving Time (s)	162.38	128.35	135.64	143.15
Average Speed (m/s)	3.35	4.59	4.35	3.94

path tracking problem in a real-world scenario. The main contribution of this paper is to demonstrate the advantage of event-triggered MPC. The study begins by developing a bicycle kinematic model, which is then utilized to apply MPC for controlling the lateral motion of the vehicle. In event-triggered MPC, the lateral offset of the vehicle's current position from the reference path is used to determine whether a new optimization problem is necessary. The testing platform is a full-size sedan equipped with a drive-by-wire system, a Polynav 2000P GNSS-inertial system, a Calmcar front view camera with lane detection capabilities, and a Dspace Autera computing unit. We conducted two experiments using time-triggered MPC and event-triggered MPC to validate the benifit of the event-triggered MPC over the time-triggered MPC. Experimental results show that the event-triggered MPC can reduce the computation in a vehicle path tracking problem by 50% while providing better control performance. For future work, more experiments should be run with different event-triggered MPC settings in more challenging driving maneuvers to validate the robustness of the proposed approaches.

REFERENCES

[1] H. Abraham, C. Lee, S. Brady, C. Fitzgerald, B. Mehler, B. Reimer, and J. F. Coughlin, "Autonomous vehicles and alternatives to driving: trust, preferences, and effects of age," in *Proceedings of the transportation research board 96th annual meeting*. Transportation Research Board Washington, DC, 2017, pp. 8–12.

[2] J. B. Rawlings, "Tutorial overview of model predictive control," *IEEE Control Systems Magazine*, vol. 20, no. 3, pp. 38–52, 2000.

[3] A. Irshayyid and J. Chen, "Comparative study of cooperative platoon merging control based on reinforcement learning," *Sensors*, vol. 23, no. 2, pp. 1–23, 2023.

[4] J. Chen, M. Liang, and X. Ma, "Probabilistic analysis of electric vehicle energy consumption using MPC speed control and nonlinear battery model," in *2021 IEEE Green Technologies Conference*, Denver, CO, April 7–9, 2021.

[5] A. Bemporad, F. Borrelli, M. Morari *et al.*, "Model predictive control based on linear programming˜ the explicit solution," *IEEE Transactions on Automatic Control*, vol. 47, no. 12, pp. 1974–1985, 2002.

[6] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.

[7] D. Bernardini and A. Bemporad, "Energy-aware robust model predictive control based on noisy wireless sensors," *Automatica*, vol. 48, no. 1, pp. 36–44, 2012.

[8] R. Badawi and J. Chen, "Enhancing enumeration-based model predictive control for dc-dc boost converter with event-triggered control," in *2022 European Control Conference*, London, UK, July 12–15, 2022.

[9] ——, "Performance evaluation of event-triggered model predictive control for boost converter," in *2022 IEEE Vehicle Power and Propulsion Conference*, Merced, CA, November 1–4, 2022.

[10] N. He and D. Shi, "Event-based robust sampled-data model predictive control: A non-monotonic lyapunov function approach," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 10, pp. 2555–2564, 2015.

[11] J. Yoo and K. H. Johansson, "Event-triggered model predictive control with a statistical learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 4, pp. 2571–2581, 2021.

[12] D. Lehmann, E. Henriksson, and K. H. Johansson, "Event-triggered model predictive control of discrete-time linear systems subject to disturbances," in *2013 European Control Conference (ECC)*. IEEE, 2013, pp. 1156–1161.

[13] F. Dang, D. Chen, J. Chen, and Z. Li, "Event-triggered model predictive control with deep reinforcement learning." [Online]. Available: https://arxiv.org/pdf/2208.10302.pdf

[14] H. Yang, Q. Li, Z. Zuo, and H. Zhao, "Event-triggered model predictive control for multi-vehicle systems with collision avoidance and obstacle avoidance," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 11, pp. 5476–5494, 2021.

[15] J. Liu, Z. Wang, and L. Zhang, "Event-triggered vehicle-following control for connected and automated vehicles under nonideal vehicle-to-vehicle communications," in *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2021, pp. 342–347.

[16] T. Wakasa, K. Sawada, and S. Shin, "Event-triggered switched pinning control for merging or splitting vehicle platoons," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15 134–15 139, 2020.

[17] N. T. Hung, A. M. Pascoal, and T. A. Johansen, "Cooperative path following of constrained autonomous vehicles with model predictive control and event-triggered communications," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 7, pp. 2644–2670, 2020.

[18] N. T. Hung and A. M. Pascoal, "Cooperative path following of autonomous vehicles with model predictive control and event triggered communications," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 562–567, 2018.

[19] Q. Han, G. Cheng, H. Yang, and Z. Zuo, "Bandwidth-aware transmission scheduling and event- triggered distributed mpc for vehicle platoons," in *2022 41st Chinese Control Conference (CCC)*, 2022, pp. 5532–5538.

[20] K. Zou, Y. Cai, L. Chen, and X. Sun, "Event-triggered nonlinear model predictive control for trajectory tracking of unmanned vehicles," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, p. 0954407021992165, 2021.

[21] J. Chen and Z. Yi, "Comparison of event-triggered model predictive control for autonomous vehicle path tracking," in *2021 IEEE Conference on Control Technology and Applications*, Austin, TX, USA, 2021, pp. 808–813.

[22] S. Huang and J. Chen, "Event-triggered model predictive control for autonomous vehicle with rear steering," *SAE Technical Paper*, no. 2022-01-0877, 2022.

[23] J. Chen, X. Meng, and Z. Li, "Reinforcement learning-based event-triggered model predictive control for autonomous vehicle path following," in *American Control Conference*, Atlanta, GA, June 8–10, 2022.

[24] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[25] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017.

[26] Z. Zhou, C. Rother, and J. Chen, "Event-triggered model predictive control for autonomous vehicle path tracking: Validation using carla simulator," *IEEE Transactions on Intelligent Vehicles*, accepted for publication April 2022.