IET Control Theory & Applications

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

Autonomous Vehicle Path Tracking Using Event-Triggered MPC With Switching Model: Methodology and Real-World Validation

Zhaodong Zhou[1] | Mingyuan Tao[2] | Jiayi Qiu[2] | Peng Zhang[2] | Meng Xu[2] | Jun Chen[1]

[1]Department of Electrical and Computer Engineering, Oakland University, Rochester, Michigan, USA | [2]Autonomous Vehicle & Motion Control, Isuzu Technical Center of America, Inc., Plymouth, Michigan, USA

Correspondence: Jun Chen (junchen@oakland.edu)

Received: 13 October 2024 | **Revised:** 7 March 2025 | **Accepted:** 2 June 2025

Funding: This work was supported in part by the faculty startup fund from Oakland University and in part by National Science Foundation through Award #2237317.

ABSTRACT

Model predictive control (MPC) is advantageous for autonomous vehicle path tracking but suffers from high computational complexity for real-time implementation. Event-triggered MPC aims to reduce this burden by optimizing the control inputs only when needed instead of every time step. Existing works in literature have been focused on algorithmic development and simulation validation for very specific scenarios. Therefore, event-triggered MPC in real-world full-size vehicle has not been thoroughly investigated. This work develops event-triggered MPC with switching model for autonomous vehicle lateral motion control, and implements it on a production vehicle for real-world validation. Experiments are conducted under both closed road and open road environments, with both low speed and high speed maneuvers, as well as stop-and-go scenarios. The efficacy of the proposed event-triggered MPC, in terms of computational load saving without sacrificing control performance, is clearly demonstrated. It is also demonstrated that event-triggered MPC can sometimes improve the control performance, even with less number of optimizations, thus contradicting to existing conclusions drawn from simulation.

1 | Introduction

Autonomous vehicles (AV) promise to significantly impact our future by making roads safer, making transportation more accessible, and cutting transportation costs [2–7]. To make autonomous driving a reality, researchers are developing complex motion control algorithms to ensure the vehicle can track the desired path. Among many methods for AV motion control, model predictive control (MPC) stands out for its ability to handle complex optimization challenges and constraints, offering a robust solution [8–10]. For example, in [10], MPC is used for trajectory tracking for multi-vehicle racing scenario where the trajectory is planned by a Nash–Stackelberg nested game framework. Meanwhile, the stability and feasibility of MPC has also been extensively studied [11–13].

However, traditional MPC requires a particularly large amount of real-time computation, so it is a challenge for vehicles with

Some results on closed road testing have been presented in IEEE International Conference on Electro Information Technology [1], which incorrectly analyzes event-trigger frequency with inaccurate conclusion. This paper corrects these oversights, extends the controller to work on full speed range, adds the test results and analysis on open road tests, and offers new insights to explain the contradictory conclusions between real-world test and simulation analysis.

limited computing power. To address this issue, event-triggered control [14–19] has been integrated into MPC. The resulting event-triggered MPC aims to reduce the computational resources by initiating the optimization process only when essential, determined by the system states or predefined criteria [20–27]. The studies on the feasibility and stability of event-triggered MPC for different types of systems are explored in various works, such as [28] for continuous-time nonlinear systems and [29, 30] for discrete-time systems. Event-triggered MPC has been investigated in the context of AV [31–40]. For example, in [31], the application of event-triggered MPC to vehicle-following dynamics addresses challenges such as unreliable vehicle-to-vehicle communication. Additionally, event-triggered MPC is implemented in [34] to simultaneously perform path tracking control and collision and obstacle avoidance. Moreover, [36] focuses on vehicle platooning, implementing event-triggered MPC for longitudinal control to ensure precise inter-vehicle spacing, and the challenge of cooperative path following involving multiple vehicles is addressed in [37, 38].

Finally, reinformance learning-based event-triggered MPC for addressing AV path tracking challenges are investigated in [39, 40]. The aforementioned studies collectively underscore the potential of event-triggered MPC not just in reducing computational demands but also in maintaining and in some instances enhancing, operational accuracy across a range of AV applications. However, it is worth noting that most of these work focus on a very specific range of driving speed and rely on simulation tools such as Matlab/Simulink or CARLA to test the computational efficiency and control performance of event-triggered MPC, while the challenges of real-world implementations and on-road testings are often overlooked.

Although event-triggered MPC has demonstrated significant benefits in simulation environments, its implementation and validation in realistic settings has not been reported. To fill this gap, our study undertakes a real-world experimental validation of event-triggered MPC for AV path tracking by developing an event-triggered MPC motion controller that works for full speed range and implementing it in a production full-size vehicle equipped with a Calmcar front view camera, a Dspace Autera computing unit, a drive-by-wire system, and a Polynav 2000P GNSS-inertial system. Note that the GNSS system plays a crucial role by providing real-time location data to the control systems, while a previously recorded GNSS path serves as the target trajectory for the tests. Extensive tests are conducted to explore the tracking accuracy and efficiency of event-triggered MPC in both low and high speed path tracking scenarios, through which we demonstrate the practical benefits and robustness of event-triggered MPC in real-world settings, underlining its potential for path tracking in autonomous driving systems.

In particular, the contributions of this paper include:

1. To avoid the numerical singularity due to inaccurate time model at low speed, a switching prediction model between dynamic model (with tire force model) and a kinematic model (without tire force model) is developed and implemented in MPC. Therefore, the proposed event-triggered MPC motion controller works at full speed range.
2. The proposed event-triggered MPC lateral motion control is implemented in a full-size vehicle, together with a conventional time-triggered MPC for benchmarking.
3. Extensive experiments are conducted for both closed road and open road environments, the latter of which includes high speed cruising, low speed turns, as well as stop-and-go manoeuvres.
4. Our findings demonstrate that event-triggered MPC not only reduces computational demand compared to the conventional time-triggered MPC but also enhances control accuracy, hence contradicting to the conclusions drawn by simulation study in [32]. This enhancement in control performance is attributed to event-triggered MPC's ability to instantly adjust control actions without delay whenever optimization triggers are not activated.

The remainder of this paper is organized as follows. Section 2 discusses the vehicle kinematic and dynamic models, which will be used as prediction model for MPC. The algorithm of time-triggered MPC and event-triggered MPC are presented in Section 3. The experiment setup and test results are discussed in Section 4, while the paper is concluded in Section 5 with future work directions.

2 | Switching Prediction Models

To accommodate both high speed and low speed scenarios, a kinematic bicycle model is used for low speed situations, while a dynamic bicycle model is applied in high-speed conditions. Note that in low speed, the lateral tire force model can be highly inaccurate, resulting in a stiff optimization problem. Therefore, the use of kinematic model at low speed is commonly adopted in literature [41], while during high-speed driving, the dynamic characteristics of the vehicle, such as air resistance, tire-to-road friction, and the distribution of the vehicle's mass, can significantly influence its behaviour. Therefore, to more accurately predict the vehicle state at high speed, a dynamic model is adopted in this paper for high speed manoeuvres. Consequently, the MPC motion control then employs a switching prediction model, i.e. kinematic model during low speed and dynamic model during high speed. More specifically, for experiments discussed later in Section 4, the kinematic model is used whenever the vehicle speed is less than 10 m/s and otherwise the dynamic model is used. This section briefly discusses both kinematic and dynamic models. More details can be found in [41].

2.1 | Vehicle Kinematic Model

The vehicle kinematic bicycle model is illustrated in Figure 1. We define the state vector $x = \begin{bmatrix} p_x & p_y & \psi \end{bmatrix}^T$ at the vehicle's centre of gravity (CG). Here, p_x and p_y denote the longitudinal and lateral positions of the vehicle, while ψ represents the vehicle's heading angle, with all states being relative to the vehicle frame. The time derivative of this state vector is given by $\dot{x} = \begin{bmatrix} \dot{p}_x & \dot{p}_y & \dot{\psi} \end{bmatrix}^T$, defined as follows.

$$\dot{p}_x = V \cos(\psi + \beta) \tag{1a}$$

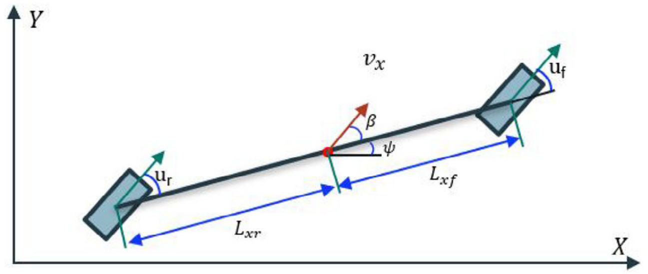

FIGURE 1 | Vehicle kinematic bicycle model.

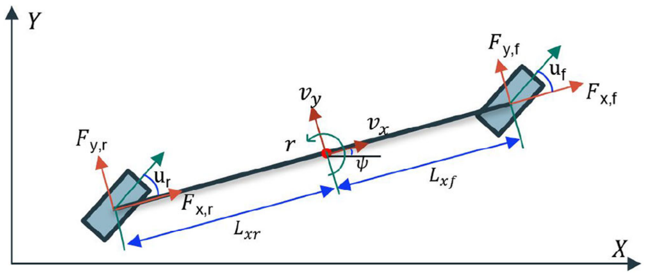

FIGURE 2 | Vehicle dynamic bicycle model.

$$\dot{p}_y = V\sin(\psi + \beta) \tag{1b}$$

$$\dot{\psi} = \frac{V\cos(\beta)}{L_{xf} + L_{xr}}(\tan(u_f) - \tan(u_r)) \tag{1c}$$

$$\beta = \arctan\left(\frac{L_{xr}\tan(u_f)}{L_{xf} + L_{xr}}\right), \tag{1d}$$

where β represents the slip angle of the vehicle, V denotes the velocity of the vehicle's CG, L_{xf} and L_{xr} are the lengths from the CG to the front and rear axles, respectively, and u_f and u_r correspond to steering angle at the front and rear. Since the testing vehicle used in this paper is front-wheel steering only, u_r can be set to zero or removed from the above equations.

2.2 | Vehicle Dynamic Model

The dynamic bicycle model used at high speed is shown in Figure 2. Similar to the kinematic model, the vehicle frame is placed at the vehicle's CG, and the state vector for the vehicle dynamic model is defined as $x = \begin{bmatrix} p_x & p_y & v_y & \psi & r \end{bmatrix}^T$. Here, p_x and p_y represent the global coordinates of the CG, while r and ψ denote the vehicle's yaw rate and heading angle in the counter-clockwise direction, respectively. Unlike the kinematic model, which uses the overall velocity at the CG, the dynamic model divides the total velocity into longitudinal velocity v_x and lateral velocity v_y along the vehicle frame. Since this paper focuses only on vehicle lateral dynamics, the longitudinal vehicle velocity v_x is treated as a measured disturbance in the vehicle dynamics. Consequently, only lateral velocity v_y is considered in the state vector x. The derivative of x is given by $\dot{x} = \begin{bmatrix} \dot{p}_x & \dot{p}_y & \dot{v}_y & \dot{\psi} & \dot{r} \end{bmatrix}^T$. The set of differential equations of the vehicle dynamic model is given as follows.

$$\dot{p}_x = v_x\cos\psi - v_y\sin\psi \tag{2a}$$

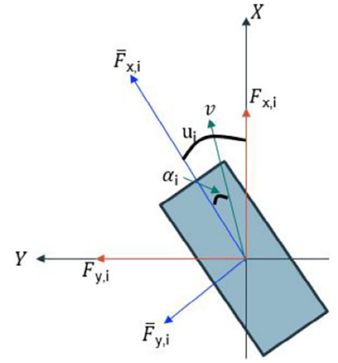

FIGURE 3 | Schematic of the tire model.

$$\dot{p}_y = v_x\sin\psi + v_y\cos\psi \tag{2b}$$

$$\dot{v}_y = -v_x r + \frac{1}{m}\sum_{i=f,r} F_{y,i} \tag{2c}$$

$$\dot{\psi} = r \tag{2d}$$

$$\dot{r} = \frac{1}{I}\left(L_{xf}F_{y,f} - L_{xr}F_{y,r}\right), \tag{2e}$$

where m is the vehicle mass, I is the vehicle rotational inertia, $F_{y,f}$ and $F_{y,r}$ are the lateral tire force on the front and rear tires, respectively.

The tire model used in this paper is shown in Figure 3. In this tire model, the tire force is discussed in two frames. Denote $\bar{F}_{x,f}$, $\bar{F}_{y,f}$, $\bar{F}_{x,r}$ and $\bar{F}_{y,r}$ as the tire forces in the wheel frame, $F_{x,i}$ and $F_{y,i}$ are the tire force in vehicle frame which is used in Equation (2). The following equations show the relationship between the tire force in the vehicle and wheel frames,

$$F_{x,i} = \bar{F}_{x,i}\cos u_i - \bar{F}_{y,i}\sin u_i \tag{3a}$$

$$F_{y,i} = \bar{F}_{x,i}\sin u_i + \bar{F}_{y,i}\cos u_i, \tag{3b}$$

where $i = \{f, r\}$ represents the front or rear wheels and u_f and u_r correspond to the steering angles of the front and rear wheels, respectively, similar to the case of kinematic model. Given the focus on front-wheel steering vehicles in this study, u_r is consistently zero. The lateral tire force $\bar{F}_{y,i}$ can be derived from a linear tire force model as:

$$\bar{F}_{y,i} = Cf_i\alpha_i, \tag{4}$$

where C is the wheel corning stiffness. Note that f_i is the friction force generated by the front/rear wheel, as obtained by:

$$f_f = \mu\frac{L_{xr}mg}{2(L_{xf} + L_{xr})}, \tag{5a}$$

$$f_r = \mu\frac{L_{xf}mg}{2(L_{xf} + L_{xr})} \tag{5b}$$

The term α_i in Equation (4) is the slip angle of the wheel, defined as $\alpha_i = \arctan\left(\bar{V}'_{y,i}/\bar{V}'_{x,i}\right)$, where $\bar{V}'_{y,i}$ and $\bar{V}'_{x,i}$ represent the corner velocities of the vehicle in the wheel frame. These velocities can be determined from the vehicle's corner velocities in the vehicle frame ($\bar{V}_{x,i}, \bar{V}_{y,i}$) through the trigonometric relations:

$$\bar{V}'_{x,i} = \bar{V}_{x,i} \cos u_i + \bar{V}_{y,i} \sin u_i \tag{6a}$$

$$\bar{V}'_{y,i} = -\bar{V}_{x,i} \sin u_i + \bar{V}_{y,i} \cos u_i. \tag{6b}$$

The vehicle's corner velocities in the vehicle frame, $\bar{V}_{x,i}$ and $\bar{V}_{y,i}$, are derived from the longitudinal (v_x) and lateral (v_y) speeds at the center of gravity (CG) as follows:

$$\bar{V}_{x,f} = v_x, \tag{7a}$$

$$\bar{V}_{y,f} = v_y + rL_{xf} \tag{7b}$$

$$\bar{V}_{x,r} = v_x, \tag{7c}$$

$$\bar{V}_{y,r} = v_y - rL_{xr} \tag{7d}$$

Since MPC does not control longitudinal dynamics, the longitudinal forces are treated as measured disturbance, which can be calculated as follows. Since the testing vehicle is front-drive only, the longitudinal force on rear tires in the tire frame is zero, i.e. $\bar{F}_{x,r} = 0$. Consequently, the longitudinal force on the front tire in the vehicle frame is $F_{x,f} = ma$, where a is longitudinal acceleration of the vehicle. Substituting the $F_{x,f} = ma$ in Equation (3a), the front tires' longitudinal force on tire frame is equal to $\bar{F}_{x,f} = (ma + \bar{F}_{y,f} \sin u_f)/\cos u_f$.

3 | MPC-Based Path Tracking

3.1 | Time-Triggered MPC for Path Tracking

In this section, MPC is used to track a desired path, by utilizing the switching prediction model described in Section 2. At each time step, the optimal control problem (OCP) being solved by MPC is given as follows.

$$\begin{aligned}\min_{U_t} \quad J = &\sum_{k=1}^{p} \left|\left|x_{t+k}(1) - p^{\text{ref}}_{x,t+k}\right|\right|^2_{Q_p} \\ &+ \sum_{k=1}^{p} \left|\left|x_{t+k}(2) - p^{\text{ref}}_{y,t+k}\right|\right|^2_{Q_p} + \sum_{k=0}^{p-1} ||u_{t+k}||^2_{Q_u} \\ &+ \sum_{k=0}^{p-1} ||u_{t+k} - u_{t+k-1}||^2_{Q_d}\end{aligned} \tag{8a}$$

$$\text{s.t.} \quad x_t = \hat{x}_t \tag{8b}$$

$$\text{Vehicle model (1) or (2)}, \quad 1 \le k \le p \tag{8c}$$

$$u_{\min} \le u_{t+k} \le u_{\max}, \quad 0 \le k \le p-1 \tag{8d}$$

$$\Delta_{\min} \le u_{t+k} - u_{t+k-1} \le \Delta_{\max}, \quad 0 \le k \le p-1, \tag{8e}$$

where $U_t = \{u_t, u_{t+1}, \dots, u_{t+p-1}\}$ is the control sequence and p is the prediction horizon.

In the objective function (8a), J comprises several terms: the first two terms penalize the deviations of the vehicle path from the target path; the third term discourages steering angles that are excessively high; the last term minimizes steering actuator change. Matrices Q_p, Q_u, and Q_d, represent the weights for each term, respectively. In the constraint (8b), $\hat{x}_t$ denotes the current state feedback. The system dynamic constraint (8c) can be obtained by discretizing the kinematic vehicle model (1) or the dynamic vehicle model (2) depending on the vehicle velocity. In this paper, the forward Euler discretization method is used when converting the continuous-time differential equations into discrete-time difference equations for MPC. Actuator constraints (8d) and (8e) limit the range and change rate of the steering angle, respectively. The objective function and constraints of the OCP (8) ensure that the resulting control actions keep the system within operational bounds while making a trajectory that closely follows the desired path with minimal steering angle and actuator changes.

3.2 | Event-Triggered MPC for Path Tracking

In conventional time-triggered MPC, the OCP (8) is triggered and solved at a fixed sampling time T_s, with the first element of U_t being implemented at the actuator and the rest of U_t being abandoned. This process repeats at the next sampling time. Time-triggered MPC is computationally demanding, requiring the solution of the OCP (8) at every time step. In contrast, event-triggered MPC adopts a more efficient approach by triggering the solving of the OCP (8) only when needed, which optimizes computational resources by reducing the frequency of calculation. In the event-triggered MPC, the decision to solve the OCP is not based on time but instead relies on certain conditions or events. Different event-trigger policy have been studied in literature, such as threshold-based (or emulation-based) policy [24, 28, 32, 33], cost function-based policy [20, 25], and self-trigger policy [26, 27]. This paper adopts the threshold-based event-trigger mechanism, as discussed and applied to AV lateral control in [32, 33].

In particular, this paper pays more attention to the lateral offset d_y, the shortest distance between the vehicle's current position and the reference path. Therefore, the event-trigger policy used in this study is defined by the following equation:

$$e = \begin{cases} 1 & \text{if } d_y > \sigma \text{ or } k > p \\ 0 & \text{Otherwise.} \end{cases} \tag{9}$$

This event-trigger policy operates based on two key parameters: σ and k. Here, σ represents the threshold for the lateral offset beyond which an event is considered to have occurred, prompting the MPC to take action. The parameter k tracks the count of consecutive instances where the OCP has not been activated, with prediction horizon p also serving as the upper limit for this count, ensuring that a control action is always available during the absence of an event. In other words, the event-triggered MPC solves the OCP (8) either when the vehicle's lateral deviation is greater than σ or when the count exceeds p, indicated by $e = 1$. If neither condition is met, $e = 0$ indicates that no new computation is needed, and the control action can be derived by shifting the optimal sequence obtained during the last event. In other words,

FIGURE 4 | The AV platform used for testing event-triggered MPC motion control.

the control action u is computed as follows.

$$u = \begin{cases} \text{Solution of (8)} & \text{if} \quad e = 1 \\ U_{t_1}(k+1) & \text{Otherwise,} \end{cases} \tag{10}$$

where U_{t_1} is the optimal sequence computed at last event.

Remark 1. Compared to the event-trigger policy used in [32], where a new optimization is triggered when the deviation from prior prediction exceeds certain threshold, the event-trigger policy (9) focuses on the deviation from the reference path. This subtle difference is crucial for real-world implementation. Note that in simulation environment such as that of [32], skipping optimization instances is the only source of error, while in the real-world testing, the model inaccuracy and stochastic nature of the environment make the prior prediction not robust. Therefore, using the deviation from the target path as criteria to determine the event can provide much robust control performance.

4 | Real-World Testing

The two MPC-based AV lateral motion controllers discussed in the previous section are implemented and evaluated using a production vehicle (described shortly), where the vehicle longitudinal control is performed manually. To ensure reproducibility of the tests and to remove the impact of a path planner, predefined paths as recorded by GPS are used as reference trajectories.

4.1 | Experimental Setup

The vehicle used for experiments is shown in Figure 4, which is a full-size sedan equipped with advanced technology features, e.g. a Calmcar front view camera, a Dspace Autera computing unit, a drive-by-wire system, and a Polynav 2000P GNSS-inertial system. For the purpose of testing event-triggered MPC, it is found out that the distances from CG to the front and rear axles are 1.2 m and 1.65 m, respectively, i.e. $L_{xf} = 1.2$ and $L_{xr} = 1.65$.

Experiments are conducted in both closed road and open road environments. The closed road testing is performed in a testing track located in Plymouth, Michigan, USA, as shown in Figure 5. Such a closed test track ensure the repeatability of the test condition. On the other hand, to enhance the evaluation of the controller's performance and incorporate conditions more reflective of everyday driving, another test was conducted in a real daily driving area, as shown in Figure 6, with the speed set to follow the actual traffic flow. Therefore, during this test, the vehicle speed can range between 0 to around 17 m/s and the test includes high speed cruising, low speed turns, and stop-and-go maneuvers. As mentioned in Section 2, to achieve accurate prediction, the prediction model used by MPC is switched between the dynamic model (when the velocity is greater than 10 m/s) and the kinematic model (when the velocity is lower than 10 m/s).

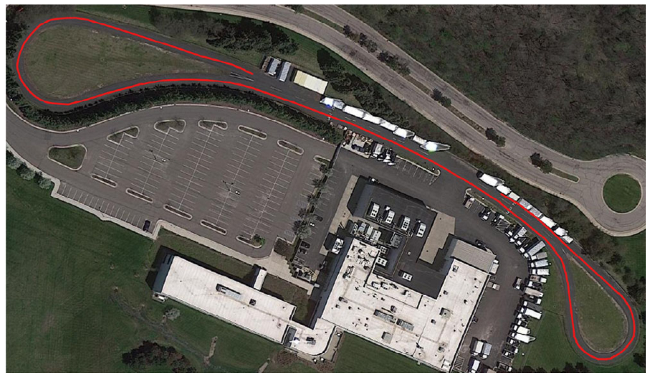

FIGURE 5 | Bird view of the testing track located in Plymouth, Michigan, USA.

FIGURE 6 | Bird view of the open road testing route on a section of public roads located in Plymouth, Michigan.

TABLE 1 | MPC parameters.

p	10	Q_u	35	u_{min} (rad)	−0.97
T_s (ms)	200	Q_d	30	Δu_{max} (rad)	0.15
Q_P	2	u_{max} (rad)	0.97	Δu_{min} (rad)	−0.15

To ensure a fair comparison, all MPC parameters are the same for these two controllers (time-triggered v.s. event-triggered MPC). This includes the calibration of cost functions, constraints on actuator capabilities, and rate constraints. The specific parameters employed for both controllers are listed in Table 1. Furthermore, this paper explores the performance of event-triggered MPC across a range of threshold σ values to analyse the influence of the event-trigger threshold.

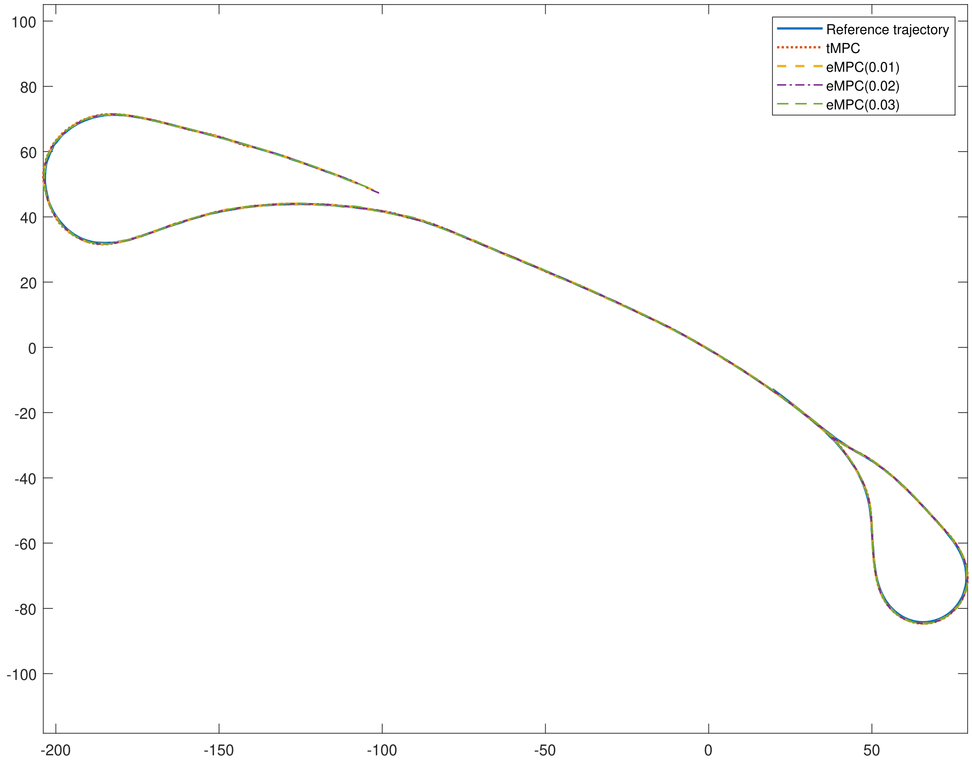

FIGURE 7 | Tracking trajectories with different controllers in the closed road test.

TABLE 2 | Computation needed and performance with different controllers in the closed road test.

	tMPC	eMPC(0.01)	eMPC(0.02)	eMPC(0.03)
Control counts	2146	2581	3160	3894
Event counts	2146	1789	1847	1870
Trigger frequency (Hz)	13.21	13.93	13.61	13.07
Driving time (s)	162.38	128.35	135.64	143.15
Average speed (m/s)	3.35	4.59	4.35	3.94
Max error (m)	0.5833	0.4108	0.4736	0.4559
RMSE (m)	0.1386	0.1056	0.1030	0.1058

4.2 | Numerical Results

In the sequel, we will denote time-triggered MPC as tMPC and denote event-triggered MPC as eMPC(σ), where σ symbolizes the event-trigger threshold.

4.2.1 | Closed Road Test

The comparison of reference trajectories and tracking paths for various MPCs in the closed road tests is shown in Figures 7 and 8. Generally, all controllers are capable of guiding the vehicle through the entire path safely. To evaluate the tracking accuracy of controllers, tracking errors are illustrated in Figure 9, which also presents the maximum error (max error) and the root mean square error (RMSE) for lateral tracking. Also see Table 2. Note that tMPC exhibits the weakest tracking performance. Specifically, tMPC has larger max error as well as a higher RMSE. Moreover, the eMPC shows similar max error and RMSE across various thresholds. An analysis of the error peaks in Figure 9 reveals that significant errors occur near the start and finish of the path. This observation is corroborated by Figure 7, indicating that the regions with notable errors correspond to the turning regions at the route's beginning and end.

In the first three rows of Table 2, "Control Counts" represents the total number of controls throughout the entire path, "Event Counts" indicates the number of events that trigger OCP resolution, and "Trigger Frequency" shows the average number of OCP resolutions within 1 s. By comparing these three rows, we can assess the eMPC's potential to reduce computational demands compared to tMPC. For example, it can be seen that eMPC(0.05) will slightly decrease the trigger frequency compared to tMPC, thereby reducing computational load. On the other hand, when assessing the tracking performance across three event-triggered MPC as listed in Table 2, it's noted that eMPC can significantly improve the control performance compared to tMPC. This is an expected result and contradict with simulation results presented in [32], which will be explained in the remark below.

Remark 2. It is noteworthy to mention that the results in this paper are different from the simulation results presented in literature [32], where tMPC demonstrates better tracking precision over eMPC and eMPC sees a decline in the performance as the threshold rises. One potential reason that leads to the different conclusion from the simulation is that the computational latency in the MPC approach, or more specifically, the computation of OCP, is not accounted for in the simulation environment. For example, solving OCP (8) generally takes 75 ms, and hence there is a 75 ms delay between initiating the MPC computation and actually transmitting the control signal to the vehicle. Such latency is usually not considered in simulation research such as [32], where the physical part of the simulation usually waits for the MPC to finish its computation. Such discrepancy makes the test conditions too ideal for tMPC and leads to inaccurate conclusions as drawn in [32]. However, on the other hand, in real-word testing as conducted in this paper, such a latency does introduce positional delay to tMPC, which worsen its closed loop control performance.

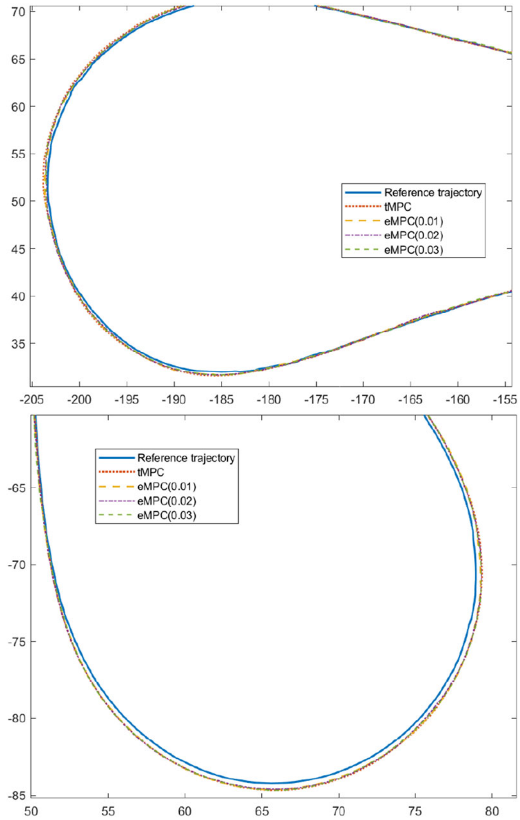

FIGURE 8 | Tracking trajectories with different controllers during the turn maneuvers in the closed road test (i.e. zoom-in version of Figure 7).

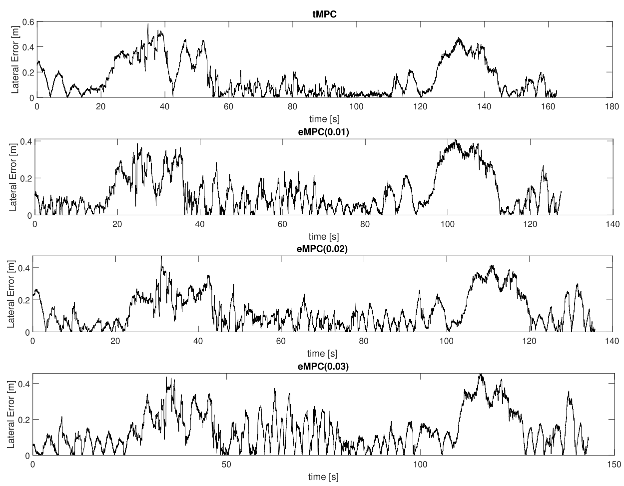

FIGURE 9 | Tracking errors with different controllers in the closed road test.

TABLE 3 | Computation needed and performance with different controllers in the open road test.

	tMPC	eMPC(0.05)	eMPC(0.05) with 100Hz call freq
Control counts	4120	20,289	11,710
Event counts	4120	4278	3328
Trigger frequency (Hz)	10.25	10.73	9.12
Driving time (s)	401.9	398.7	364.8
Average speed (m/s)	12.82	12.87	14.14
Max error (m)	0.732	0.582	0.718
RMSE (m)	0.1253	0.1232	0.1281

Remark 3. To see why eMPC can improve control performance, it is important to note that when the event-triggered condition is not met, eMPC simply uses the previous optimal control sequence to determine the current control action. This requires only a minimal amount of additional computational effort. Such a timely compensation in between two events can be the reason why the event-triggered MPC demonstrated superior performance compared to the time-triggered MPC in real-world experiments.

4.2.2 | Open Road Test

Figure 10 displays the reference trajectory and the tracking paths for various MPC controllers in the open road test. Overall, all controllers successfully guide the vehicle through the entire path without significant offsets. Figure 11 shows the tracking errors for each controller, with the max error and RMSE being presented in Table 3. Note that tMPC has a higher max error value compared to eMPC, which means tMPC has the least tracking accuracy, for a similar reason discussed previously in Section 4.2.1. It is also noted that, on average, in eMPC, for every four control signals only one needs to consume time to solve the OCP, and the rest of the three control signals can provide timely compensation for the delay caused by OCP computation.

However, a more thorough data analysis reveals a drawback here. In particular, for eMPC(0.05), within 398.7 s, there are 20,289

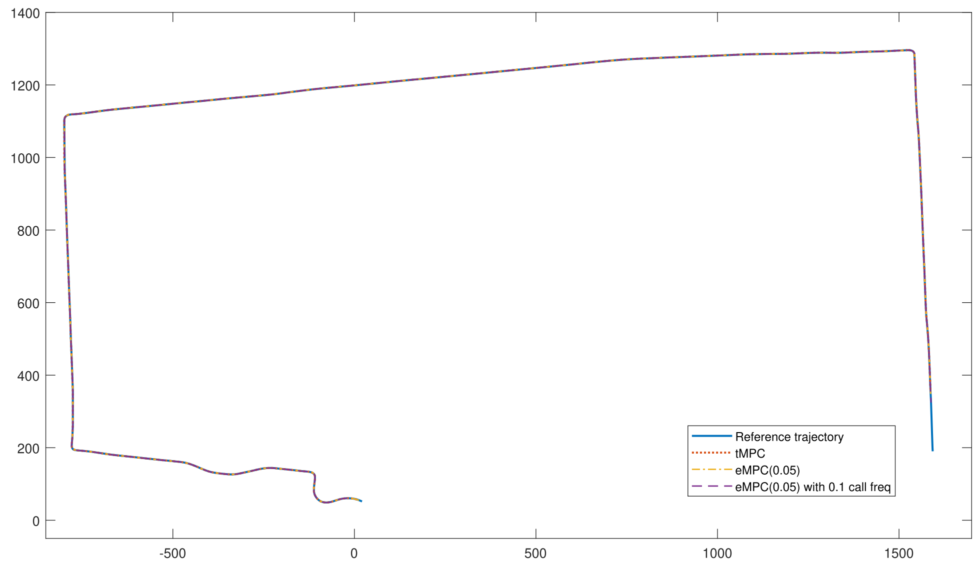

FIGURE 10 | Tracking trajectories with different controllers in the open road test.

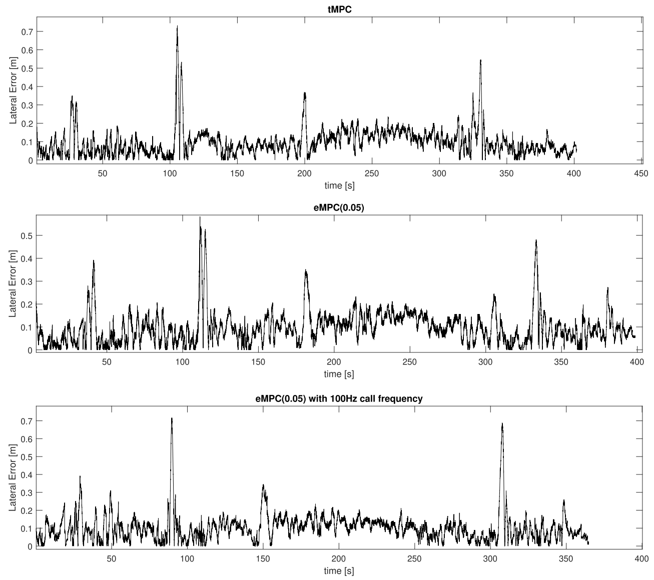

FIGURE 11 | Tracking errors with different controllers in the open road test.

controls, 4278 of which requires MPC computation. Therefore, it is estimated that when an event is not triggered, there is a new control command every 5 ms or less, which can be shorter than the time constant of the steering actuators used in our test vehicle. Therefore, to allow sufficient time for the steering actuator to respond to the control signal, another test is conducted in which a 100 Hz calling frequency in the controller is enforced. This means during the absence of an event, if the time interval between two control signals is less than 10 ms, it will be delayed until reaching 10 ms, resulting in a drop in event-trigger frequency by 1 to 9.12 Hz. Compared to eMPC(0.05), the Max Error and RMSE increase but are still comparable with tMPC. Additionally, it is observed that the average speed in this test is higher than in the previous two tests, making the lateral control more difficult in this particular run. Normally, under higher speeds, control during turning maneuvers is more challenging than at lower speeds. Nonetheless, the tracking accuracy of eMPC(0.05) with a 100 Hz calling frequency is still better than that of tMPC, and the event-trigger frequency is lower than both tMPC and the standard eMPC(0.05). Note that the reason for the different average speed is attributed to the speed control not being managed by the drive-by-wire system and also to the fact that in open road test the traffic flow is dynamic and acts as random noise to our test.

In Figure 11, the lateral errors at each time step for the three controllers are presented. Each figure contains four protrusions when compared to Figure 10, showing that the areas with significant errors are located in the four turning regions. Figure 12 zooms in on these areas for a closer examination, with the image in the top right corner representing the second turning point. It can be seen that, compared to the other three turns, there is a more noticeable deviation of the three control trajectories from the purple reference trajectory in this turn. This deviation is also reflected in Figure 11, where the second peak is significantly higher than the other three.

5 | Conclusion

This paper presented an event-triggered MPC for AV lateral motion control. To overcome the different speed conditions as often encountered in real-world driving, a switching prediction model is used to switch between kinematic and dynamic models according to the real-time vehicle speed. The developed control strategy is implemented and tested in a production vehicle, i.e. a full-size sedan equipped with a Calmcar front view camera, a Dspace Autera computing unit, a drive-by-wire system, and a Polynav 2000P GNSS-inertial system. Both closed road test and open road test are conducted, the latter of which is performed in public road section and includes high speed cruising, low speed turns, and stop-and-go. The result shows that compared to the time-triggered MPC, event-triggered MPC can achieve better tracking performance compared to traditional time-triggered MPC, while reducing computational burden. Moreover, under both test conditions, the event-triggered MPC has a better

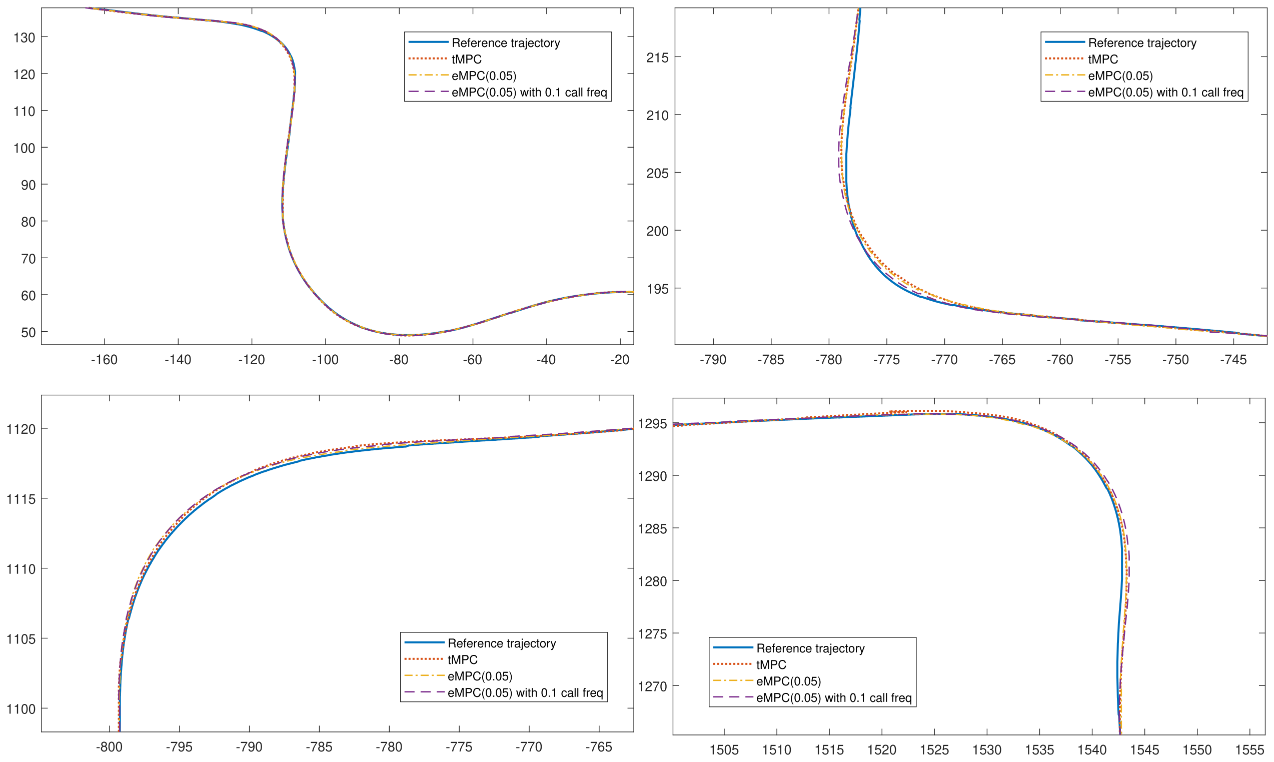

FIGURE 12 | Tracking trajectories with different controllers during the turn manoeuvres in the open road test (i.e. zoom-in version of Figure 10).

tracking accuracy, because the event-triggered MPC can provide timely compensation (when an event is not triggered) for the delay caused by MPC computation. The results presented here supplement existing work in literature and contradict to the conclusions drawn from prior simulation research that did not account for the MPC computation delay. Future work directions include the implementation of longitudinal speed control using MPC to remove the variation of test speed and the comparison with other event-triggered control methods. In addition, the use of a constant gain feedback controller during the absence of an event will also be investigated.

Author Contributions

Zhaodong Zhou: data curation, formal analysis, writing – original draft. **Mingyuan Tao:** data curation. **Jiayi Qiu:** data curation, **Peng Zhang:** data curation. **Meng Xu:** data curation. **Jun Chen:** conceptualization, funding acquisition, methodology, supervision, writing – review & editing.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. Z. Zhou, J. Chen, M. Tao, P. Zhang, and M. Xu, "Experimental Validation of Event-Triggered Model Predictive Control for Autonomous Vehicle Path Tracking," in *2023 IEEE International Conference on Electro Information Technology* (IEEE, 2023), 18–20.

2. M. Martínez-Díaz and F. Soriguera, "Autonomous Vehicles: Theoretical and Practical Challenges," *Transportation Research Procedia* 33 (2018): 275–282.

3. D. Parekh, N. Poddar, A. Rajpurkar, et al., "A Review on Autonomous Vehicles: Progress, Methods and Challenges," *Electronics* 11, no. 14 (2022): 2162.

4. A. Rasouli and J. K. Tsotsos, "Autonomous Vehicles that Interact With Pedestrians: A Survey of Theory and Practice," *IEEE Transactions on Intelligent Transportation Systems* 21, no. 3 (2020): 900–918.

5. Y. Sun, D. Ren, S. Lian, S. Fu, X. Teng, and M. Fan, "Robust Path Planner for Autonomous Vehicles On Roads With Large Curvature," *IEEE Robotics and Automation Letters* 7, no. 2 (2022): 2503–2510.

6. Y. Zeng, Y. Hu, S. Liu, J. Ye, et al., "RT3D: Real-Time 3-D Vehicle Detection in Lidar Point Cloud for Autonomous Driving," *IEEE Robotics and Automation Letters* 3, no. 4 (2018): 3434–3440.

7. L. Wang, L. Sun, M. Tomizuka, and W. Zhan, "Socially-Compatible Behavior Design of Autonomous Vehicles With Verification on Real Human Data," *IEEE Robotics and Automation Letters* 6, no. 2 (2021): 3421–3428.

8. J. B. Rawlings, "Tutorial Overview of Model Predictive Control," *IEEE Control Systems Magazine* 20, no. 3 (2000): 38–52.

9. P. Hang, X. Chen, and W. Wang, "Cooperative Control Framework for Human Driver and Active Rear Steering System to Advance Active Safety," *IEEE Trans. Intelligent Vehicles* 6, no. 3 (2021): 460–469.

10. Y. Cui, J. Tang, Q. Luo, Z. Feng, and T. Huang, "A Nash-Stackelberg Game Theoretic Planner for Many-To-Few Multi-Vehicle Racing," *IEEE Transactions on Intelligent Vehicles* (2024), https://doi.org/10.1109/TIV.2024.3478391.

11. A. Boccia, L. Grüne, and K. Worthmann, "Stability and Feasibility of State Constrained MPC Without Stabilizing Terminal Constraints," *Systems & Control Letters* 72 (2014): 14–21.

12. S. Gros and M. Zanon, “Learning for MPC With Stability & Safety Guarantees,” *Automatica* 146 (2022): 110598.

13. D. L. Marruedo, T. Alamo, and E. F. Camacho, “Stability Analysis of Systems With Bounded Additive Uncertainties Based on Invariant Sets: Stability and Feasibility of MPC,” in *2002 American Control Conference* (2002), 364–369.

14. W. P. Heemels, K. H. Johansson, and P. Tabuada, “An Introduction to Event-Triggered and Self-Triggered Control,” in *IEEE Conference on Decision and Control* (IEEE, 2012), 3270–3285.

15. Q. Hou and J. Dong, “Distributed Dynamic Event-Triggered Consensus Control for Multiagent Systems With Guaranteed l_2 Performance and Positive Inter-Event Times,” *IEEE Transactions on Automation Science and Engineering* 21, no. 1 (2022): 746–757.

16. D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, “Distributed Event-Triggered Control for Multi-Agent Systems,” *IEEE Transactions on Automatic Control* 57, no. 5 (2011): 1291–1297.

17. H. Zhang, D. Yue, X. Yin, and J. Chen, “Adaptive Model-Based Event-Triggered Control of Networked Control System With External Disturbance,” *IET Control Theory & Applications* 10, no. 15 (2016): 1956–1962.

18. X.-M. Zhang and Q.-L. Han, “Event-Triggered Dynamic Output Feedback Control for Networked Control Systems,” *IET Control Theory & Applications* 8, no. 4 (2014): 226–234.

19. X. Chen and F. Hao, “Event-Triggered Average Consensus Control for Discrete-Time Multi-Agent Systems,” *IET Control Theory & Applications* 6, no. 16 (2012): 2493–2498.

20. 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* 62, no. 10 (2015): 2555–2564.

21. S. Huang and J. Chen, *Event-Triggered Model Predictive Control for Autonomous Vehicle With Rear Steering* (SAE World Congress, Detroit, MI, 2022), https://doi.org/10.4271/2022-01-0877.

22. J. Yoo and K. H. Johansson, “Event-Triggered Model Predictive Control With A Statistical Learning,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51, no. 4 (2021): 2571–2581.

23. Y. Wang, Y. Yan, T. Shen, S. Bai, J. Hu, L. Xu, and G. Yin, “An Event-Triggered Scheme for State Estimation of Preceding Vehicles Under Connected Vehicle Environment,” *IEEE Transactions on Intelligent Vehicles* 8, no. 1 (2023): 583–593.

24. F. D. Brunner, W. Heemels, and F. Allgöwer, “Robust Event-Triggered Mpc With Guaranteed Asymptotic Bound and Average Sampling Rate,” *IEEE Transactions on Automatic Control* 62, no. 11 (2017): 5694–5709.

25. A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos, “Novel Event-Triggered Strategies for Model Predictive Controllers,” in *IEEE Conf. Decision and Control* (IEEE, 2011), 3392–3397.

26. K. Hashimoto, S. Adachi, and D. V. Dimarogonas, “Self-Triggered Model Predictive Control for Nonlinear Input-Affine Dynamical Systems Via Adaptive Control Samples Selection,” *IEEE Transactions on Automatic Control* 62, no. 1 (2016): 177–189.

27. F. D. Brunner, M. Heemels, and F. Allgöwer, “Robust Self-Triggered Mpc for Constrained Linear Systems: A Tube-Based Approach,” *Automatica* 72 (2016): 73–83.

28. H. Li and Y. Shi, “Event-Triggered Robust Model Predictive Control of Continuous-Time Nonlinear Systems,” *Automatica* 50, no. 5 (2014): 1507–1513.

29. C. Liu, H. Li, Y. Shi, and D. Xu, “Codesign of Event Trigger and Feedback Policy in Robust Model Predictive Control,” *IEEE Transactions on Automatic Control* 65, no. 1 (2019): 302–309.

30. H. Li, W. Yan, and Y. Shi, “Triggering and Control Codesign in Self-Triggered Model Predictive Control of Constrained Systems: With Guaranteed Performance,” *IEEE Transactions on Automatic Control* 63, no. 11 (2018): 4008–4015.

31. 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), 342–347.

32. J. Chen and Z. Yi, “Comparison of Event-Triggered Model Predictive Control for Autonomous Vehicle Path Tracking,” in *IEEE Conference on Control Technology and Applications* (2021), 808–813.

33. 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* 8, no. 6 (June, 2023): 3547–3555.

34. 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* 31, no. 11 (2021): 5476–5494.

35. C. Rother, Z. Zhou, and J. Chen, “Development of A Four-Wheel Steering Scale Vehicle for Research and Education on Autonomous Vehicle Motion Control,” *IEEE Robotics and Automation Letters* 8, no. 8 (August, 2023): 5015–5022.

36. T. Wakasa, K. Sawada, and S. Shin, “Event-Triggered Switched Pinning Control for Merging or Splitting Vehicle Platoons,” *IFAC-PapersOnLine* 53, no. 2 (2020): 15 134–15 139.

37. 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* 30, no. 7 (2020): 2644–2670.

38. N. T. Hung and A. M. Pascoal, “Cooperative Path Following of Autonomous Vehicles With Model Predictive Control and Event Triggered Communications,” *IFAC-PapersOnLine* 51, no. 20 (2018): 562–567.

39. J. Chen, X. Meng, and Z. Li, “Reinforcement Learning-Based Event-Triggered Model Predictive Control for Autonomous Vehicle Path Following,” in *American Control Conference* (IEEE, 2022), 3342–3347.

40. F. Dang, D. Chen, J. Chen, and Z. Li, “Event-Triggered Model Predictive Control With Deep Reinforcement Learning,” *IEEE Transactions on Intelligent Vehicles* 9, no. 1 (2024): 459–468.

41. R. Rajamani, *Vehicle Dynamics and Control* (Springer, 2011).