# Comparison of Event-Triggered Model Predictive Control for Autonomous Vehicle Path Tracking

Jun Chen, *Senior Member, IEEE* and Zonggen Yi, *Member, IEEE*

*Abstract*— **This paper proposes two different event-triggered nonlinear model predictive controls (NMPC) for autonomous vehicle path tracking. The difference between the two event-triggered NMPCs is the determination of control action when an event is not triggered. In the first formulation, the optimal control sequence computed from last triggering event is shifted to determine control action when NMPC is not triggered, while in the second formulation, a time-triggered linear parametric varying MPC (LPV-MPC) with shorter prediction horizon is formulated and solved in between NMPC triggering events to compensate prediction error and disturbance. These two event-triggered NMPCs, together with a time-triggered LPV-MPC and a time-triggered NMPC serving as benchmark, are implemented to track the vehicle path in both longitudinal and lateral directions, with axle driving torque and front steering input as the control variables. Control performance and throughput requirements of different MPCs are then measured and compared, where the advantage of event-triggered formulation is clearly demonstrated.**

## I. Introduction

Model predictive control (MPC) can deal with both linear and nonlinear systems subject to constraints, and has been one of the most popular control methods [1]. MPC solves, for every control loop, a constrained optimization problem incorporating finite horizon predictions based on a dynamical model and the current state estimation. MPC has been extensively studied and many results have been obtained regarding the stability, robustness, and feasibility [2]–[4]. Due to its capability to deal with constraints on both states and inputs, MPC has been widely studied for many applications, such as vehicle systems [5], [6], smart grids [7], and electric motors [8], etc. Recent research and development has been focused on adopting MPC in embedded automotive systems, in order to improve the fuel economy and ride comfort. For example, [9], [10] applied MPC for autonomous vehicle (AV) control, while [11], [12] applied MPC in electric vehicle (EV) for energy optimization. As a real-time optimal control technique, MPC is very suitable for integrated vehicle control, where the control commands of multiple actuators are simultaneously optimized [13], [14]. MPC has also been investigated for possible usage in active safety [15].

Despite the advantage of dealing with system constraints, MPC does require high computation power, which further increases when the system dimension or prediction horizon increases. One way to improve the MPC computational efficiency without degrading the control performance is event-triggered MPC, in which MPC is triggered to formulate and solve the OCP only when it is needed, as opposed to being time-triggered at fixed sampling rate. See for example [16]–[21], and the reference therein. By allowing optimization updates only when it's necessary, the event-triggered MPC can significantly reduce the throughput. Different event-trigger mechanisms have been proposed in literature. For example, an event can be triggered if the deviation of the actual system states from the prediction exceeds a certain threshold, where the prediction is computed when MPC solves the OCP at the last event, [18], [19]. This approach is also called emulation-based event-triggered MPC, and is very advantageous for linear MPC, in which the event-triggering condition can be then translated to conditions on error dynamics, and often can result in guaranteed robustness when the maximum disturbance is assumed to be known. Another widely used method to define the event-triggering mechanism is based on the cost function of the OCP [22], [23], where an event is triggered if the decreasing of cost function is not guaranteed. Finally, an event can also be self-triggered [24], [25]. In this approach the timing of next event is determined when the current event is triggered.

In this paper, we propose two formulations of event-triggered NMPC for AV path tracking, whose difference lies in the determination of control action when an event is not triggered. In the first formulation, the optimal control sequence computed from last triggering event is shifted to determine control action when NMPC is not triggered, while in the second formulation, a time-triggered linear parametric varying MPC (LPV-MPC) with shorter prediction horizon is formulated and solved in between NMPC triggering events to compensate prediction error and disturbance. To compare the aforementioned event-triggered NMPCs, together with a time-triggered LPV-MPC and a time-triggered NMPC serving as benchmark, an AV path tracking problem similar to that of [9] is formulated. In other words, the optimal control problem is formulated to track the vehicle path in both longitudinal and lateral directions, with axle driving torque and front steering input as the control variables. Control performance and throughput requirements of different MPC setups are then measured and compared. The contribution of this paper is two folds: (1) we apply event-triggered NMPC in AV path tracking problem, where the prior work focused on time-triggered MPC; and (2) we propose and investigate the use of time-triggered LPV-MPC in between two NMPC triggering events to allow fast control correction and reduced

This work is supported in part by SECS Faculty Startup Fund and URC Faculty Research Fellowship at Oakland University.

Jun Chen is with the Department of Electrical and Computer Engineering, Oakland University, Rochester, MI 48309, USA (email: junchen@oakland.edu).

Zonggen Yi is with Idaho National Laboratory, Idaho Falls, ID 83415, USA (email: zonggen.yi@inl.gov).

NMPC triggering frequency.

The rest of this paper is organized as follows. Section II presents the event-triggering NMPC formulations together with other MPC setups, while Section III introduces the AV path tracking problem. Section IV presents numerical results and the paper is concluded in Section V.

## II. Time-Triggered and Event-Triggered MPC

Consider the following discrete-time system dynamics

$$\zeta_{t+1} = f(\zeta_t, u_t), \tag{1}$$

where $\zeta_t \in \mathbb{R}^n$ is the system state at discrete time $t$, $u_t \in \mathbb{R}^m$ is the control input. Given a prediction horizon $p$, the MPC aims to find the optimal control sequence $U_t$ and optimal state sequence $Z_t$ by solving an optimal control problem (OCP), where $U_t$ and $Z_t$ are defined as $U_t = \{u_t, u_{t+1}, \ldots, u_{t+p-1}\}$ and $Z_t = \{\zeta_{t+1}, \zeta_{t+2}, \ldots, \zeta_{t+p}\}$.

### A. Time-triggered NMPC

At each sampling time $t$, the conventional time-triggered NMPC solves the following OCP:

$$\begin{aligned}
\min_{Z_t, U_t} \quad & J(Z_t, U_t) && \text{(2a)}\\
\text{s.t.} \quad & \zeta_t = \hat{\zeta}_t && \text{(2b)}\\
& \zeta_{t+k} = f(\zeta_{t+k-1}, u_{t+k-1}), \quad 1 \le k \le p && \text{(2c)}\\
& \zeta_{min} \le \zeta_{t+k} \le \zeta_{max}, \quad 1 \le k \le p && \text{(2d)}\\
& u_{min} \le u_{t+k} \le u_{max}, \quad 0 \le k \le p-1 && \text{(2e)}\\
& \Delta_{min} \le u_{t+k} - u_{t+k-1} \le \Delta_{max}, \\
& \qquad\qquad 0 \le k \le p-1, && \text{(2f)}
\end{aligned}$$

where for (2b), $\hat{\zeta}_t$ denotes the current state estimation, and for (2f), $u_{t-1}$ denotes the control action applied at previous loop. For NMPC, the cost function $J$ in (2a) can be any nonlinear function defined over $U_t$ and $Z_t$. The above OCP is solved for every sampling time $t$, and the first element of $U_t$, i.e., $u_t$ is then applied to actuators as control action, while the remaining of the optimal sequence $U_t$ is abandoned.

### B. Time-triggered LPV-MPC

At each sampling time $t$, one can linearized and discretize (1) around the nominal operating point $(u_0, \zeta_0) = (u_{t-1}, \hat{\zeta}_t)$ to obtain a discrete-time LPV model, as follows:

$$\delta\zeta_{t+1} = f_0 + A_t \delta\zeta_t + B_t \delta u_t. \tag{3}$$

The time-triggered LPV-MPC solves the following OCP:

$$\begin{aligned}
\min_{Z_t, U_t} \quad & J(Z_t, U_t) = \sum_{k=1}^{p} ||\zeta_{t+k} - \zeta_{t+k}^r||_{Q_\zeta}^2 \\
& + \sum_{k=0}^{p-1} \left( ||u_{t+k} - u_{t+k}^r||_{Q_u}^2 + ||u_{t+k} - u_{t+k-1}||_{Q_d}^2 \right) && \text{(4a)}\\
\text{s.t.} \quad & \delta\zeta_t = 0 && \text{(4b)}\\
& \delta\zeta_{t+k} = f_0 + A_t \delta\zeta_{t+k-1} + B_t \delta u_{t+k-1}, \\
& \qquad\qquad 1 \le k \le p && \text{(4c)}\\
& \zeta_{t+k} = \zeta_0 + \delta\zeta_{t+k}, \quad 1 \le k \le p && \text{(4d)}\\
& u_{t+k} = u_0 + \delta u_{t+k}, \quad 0 \le k \le p-1 && \text{(4e)}\\
& \zeta_{min} \le \zeta_{t+k} \le \zeta_{max}, \quad 1 \le k \le p && \text{(4f)}\\
& u_{min} \le u_{t+k} \le u_{max}, \quad 0 \le k \le p-1 && \text{(4g)}\\
& \Delta_{min} \le u_{t+k} - u_{t+k-1} \le \Delta_{max}, \\
& \qquad\qquad 0 \le k \le p-1, && \text{(4h)}
\end{aligned}$$

where $Z_t^r = \{\zeta_{t+1}^r, \ldots, \zeta_{t+p}^r\}$ and $u_t^r = \{u_t^r, \ldots, u_{t+p-1}^r\}$ are the state and input reference for tracking. It is trivial to see that the above OCP can be translated into linearly constrained quadratic programming (QP) problems, and can be solved in real-time by embedded devices [5], [26], [27].

**Algorithm 1** Event-Triggered NMPC

```
procedure eNMPC($\hat{\zeta}_t$, $k$, $U_{t_1}$, $Z_{t_1}$)
    $k \leftarrow k + 1$;
    $e \leftarrow$ computing (5);
    if $e = 1$ then
        $k \leftarrow 0$;
        $(Z_t, U_t) \leftarrow$ Solving OCP (2);
        $u \leftarrow U_t(1)$;
        $U_{t_1} \leftarrow U_t$;
        $Z_{t_1} \leftarrow Z_t$;
    else
        $u \leftarrow U_{t_1}(k+1)$;
    end if
    return $u$, $k$, $U_{t_1}$, $Z_{t_1}$
end procedure
```

### C. Event-triggered NMPC

Unlike time-triggered MPCs presented in previous sections, which solve the OCP at every sampling time, the event-triggered NMPC solves the OCP (2) only when an event is triggered. In this paper, we consider the following event-triggering mechanism, which is adopted from [18]. At sampling time $t$, given the optimal sequence $Z_{t_1}$ computed at last event (at time $t_1$) and the current state estimation $\hat{\zeta}_t$, an event $e$ is defined by

$$e = \begin{cases} 1 & \text{if } ||Z_{t_1}(k) - \hat{\zeta}_t||_Q^\infty > \sigma \text{ or } k > k_{max} \\ 0 & \text{Otherwise} \end{cases}, \tag{5}$$

where $k$ is such that $t = t_1 + kT_s$ with $T_s$ being the sampling time, and $\sigma$ and $k_{max}$ are calibration parameters that influence the frequency of event-triggering. In other words, an event-triggered NMPC solves the OCP (2) only when $e = 1$. When $e = 0$, the control action can be determined using the optimal sequence $U_{t_1}$ computed at last event [16], i.e.,

$$u = \begin{cases} \text{Solution of (2)} & \text{if} \quad e = 1 \\ U_{t_1}(k+1) & \text{Otherwise} \end{cases}. \tag{6}$$

Furthermore, it is trivial to see that $k_{max} < p$, and otherwise (6) can be undefined. Algorithm 1 summarizes the event-triggered NMPC for each sampling time $t$.

**Algorithm 2** Event-Triggered NMPC w. LPV Compensation

```
1:  procedure EMPC_LPV($\hat{\zeta}_t$, $k$, $U_{t_1}$, $Z_{t_1}$)
2:      $k \leftarrow k+1$;
3:      $e \leftarrow$ computing (5);
4:      if $e = 1$ then
5:          $k \leftarrow 0$;
6:          $(Z_t, U_t) \leftarrow$ Solving OCP (2);
7:          $u \leftarrow U_t(1)$;
8:          $U_{t_1} \leftarrow U_t$;
9:          $Z_{t_1} \leftarrow Z_t$;
10:     else
11:         $k_1 \leftarrow \min(k + p_l, p)$;
12:         $u_t^r \leftarrow \{U_{t_1}(k+1), \ldots, U_{t_1}(k_1)\}$;
13:         $Z_t^r \leftarrow \{Z_{t_1}(k+1), \ldots, Z_{t_1}(k_1)\}$;
14:         $u \leftarrow$ Solving OCP (4);
15:     end if
16:     return $u$, $k$, $U_{t_1}$, $Z_{t_1}$
17: end procedure
```

The event-triggered NMPC described above reduces throughput requirement by decreasing the frequency of solving the nonlinear OCP (2). In practice, the control performance can be slightly degraded since event-triggered NMPC can only compensate the model mismatch and/or unmeasured disturbance in a re-active fashion. In this paper, we propose a new event-triggered NMPC that differs from Algorithm 1 by how the control action $u$ is computed when the event is not triggered, i.e., Line 11. Specifically, we propose to use a time-triggered LPV-MPC with shorter prediction horizon to track $Z_{t_1}$ when $e = 0$. By having a closed loop control in between event-triggering, the overall control performance is expected to be better than that of Algorithm 1. The LPV-MPC considered in this case solves the similar OCP problem (4) as described in Section II-B, with prediction horizon $p_l$ being shorter than that of NMPC, and the input and state references $Z_t^r$ and $u_t^r$ computed by $u_t^r = \{U_{t_1}(k+1), \ldots, U_{t_1}(\min(k+p_l, p))\}$ and $Z_t^r = \{Z_{t_1}(k+1), \ldots, Z_{t_1}(\min(k+p_l, p))\}$.

In other words, this LPV-MPC tracks a section of the previous optimal sequences $U_{t_1}$ and $Z_{t_1}$ by utilizing the up-to-date state estimation and model linearization to compensate the prediction error made at last event. Algorithm 2 summarizs the event-triggered NMPC with LPV compensation, which runs each sampling time $t$. Note that the optimal sequences from the OCP (4) is abandoned and not passed to the next control loop.

## III. Autonomous Vehicle Path Tracking

This section presents the bicycle model, as depicted in Fig. 1, which can be used as prediction model for MPC, and formulates the AV path tracking problem.

### A. Vehicle Dynamics Model at CG

The equations for vehicle center of gravity (CG) and wheel dynamics are given by

$$\dot{x} = v_x \cos\psi - v_y \sin\psi \tag{7a}$$

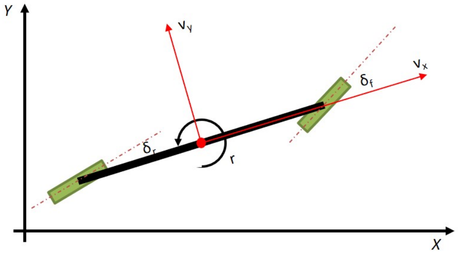

Fig. 1. The bicycle model for vehicle dynamic.

$$\dot{v}_x = v_y r + \frac{2}{m}\sum_{i=f,r} F_{x,i} - g \sin\sigma_g - \frac{1}{m}F_a \tag{7b}$$

$$\dot{y} = v_x \sin\psi + v_y \cos\psi \tag{7c}$$

$$\dot{v}_y = -v_x r + \frac{2}{m}\sum_{i=f,r} F_{y,i} \tag{7d}$$

$$\dot{\psi} = r \tag{7e}$$

$$\dot{r} = \frac{1}{I}\left(2L_{xf}F_{y,f} - 2L_{xr}F_{y,r}\right), \tag{7f}$$

where $x$, $y$ and $\psi$ are the vehicle CG longitudinal position, lateral position, and rotational angle, in *global* inertial frame, and $v_x$, $v_y$, and $r$ are the vehicle longitudinal velocity, lateral velocity, and yaw rate, in *vehicle* frame, $m$ is the vehicle mass, $I$ is the vehicle rotational inertia on yaw dimension, $L_{xf}$ and $L_{xr}$ are the distance from CG to the middle of front and rear axle, respectively.

According to [28], the aerodynamic drag force, $F_a$, can be modeled by, assuming wind speed is 0,

$$F_a = \frac{1}{2}\rho C_d A_F v_x^2, \tag{8}$$

where $\rho$ is the air mass density, $C_d$ is the aerodynamic drag coefficient, $A_F$ is the effective front area. Typical values for $\rho$, $C_d$ and $A_F$ are: 1.225 $kg/m^3$, $0.25 - 0.3$, and $1.6 + 0.00056(m - 756)$. More details regarding vehicle dynamics can be found in [10], [28].

### B. Tire Force Model

The tire force $F_{x,i}$ and $F_{y,i}$ in (7b), (7d), and (7f) are expressed in vehicle frame. Denote $\bar{F}_{x,i}$ and $\bar{F}_{y,i}$ as the tire force in wheel frame, then we have the following relationship, for $i = \{f, r\}$,

$$F_{x,i} = \bar{F}_{x,i}\cos\beta_i - \bar{F}_{y,i}\sin\beta_i \tag{9a}$$

$$F_{y,i} = \bar{F}_{x,i}\sin\beta_i + \bar{F}_{y,i}\cos\beta_i, \tag{9b}$$

where $\beta_i$ is the wheel-road-angle for the wheel $i$. Denote the wheel rotational speed for wheel $i$ as $\omega_i$, then we have

$$\dot{\omega}_i = \frac{T_i/2 - \bar{F}_{x,i}R}{I_w}, \tag{10}$$

where $T_i$ is the propulsion/braking torque at the axle, $R$ is the effective tire radius and $I_w$ is the wheel rotational inertia.

The tire slip ratio $s_i$ and slip angle $\alpha_i$ are defined as (for driving forward)

$$s_i = \frac{\omega_i R - \bar{v}_{x,i}}{\bar{v}_{x,i}} \tag{11a}$$

$$\alpha_i = \arctan \frac{\bar{v}_{y,i}}{\bar{v}_{x,i}}, \tag{11b}$$

where the vehicle corner velocities at wheel frame $\bar{v}_{x,i}$ and $\bar{v}_{y,i}$ are given by

$$\bar{v}_{x,i} = v_{x,i} \cos \beta_i + v_{y,i} \sin \beta_i \tag{12a}$$

$$\bar{v}_{y,i} = -v_{x,i} \sin \beta_i + v_{y,i} \cos \beta_i. \tag{12b}$$

Furthermore, the vehicle corner velocities at vehicle frame $v_{x,i}$ and $v_{y,i}$ then are given by

$$v_{x,f} = v_{x,r} = v_x \tag{13a}$$

$$v_{y,f} = v_y + L_{xf} r \tag{13b}$$

$$v_{y,r} = v_y - L_{xr} r, \tag{13c}$$

Finally, the normal force can be modeled by static load transfer,

$$F_{z,i} = \frac{L_{xr} m g}{2(L_{xf} + L_{xr})} \qquad i = 1, 2 \tag{14a}$$

$$F_{z,i} = \frac{L_{xf} m g}{2(L_{xf} + L_{xr})} \qquad i = 3, 4 \tag{14b}$$

One can assume that during each discretized prediction step the slip is at steady state, hence $\dot{s}_i = 0$. Further assume that the change of $\bar{v}_{x,i}$ is negligible compared to the change of slip ratio, then we have $0 = \dot{s}_i = (\dot{\omega}_i R)/\bar{v}_{x,i} = (T_i/2 - \bar{F}_{x,i} R)R/I_w/\bar{v}_{x.i}$. Therefore $\bar{F}_{x,i} = T_i/2/R$. Also assume that the lateral tire force is within the linear range, and so $\bar{F}_{y,i} = C_i \mu_i F_{z,i} \alpha_i$, where $C_i$ is the tire corner stiffness and $\mu_i$ characterize the road surface. In other words, we adopt the following linear tire model

$$\bar{F}_{x,i} = \frac{T_i}{2R}, \qquad \bar{F}_{y,i} = C_i \mu_i F_{z,i} \alpha_i \tag{15}$$

### C. Complete AV Plant Model

Putting everything together, the nonlinear model for AV path tracking can be compactly represented as

$$\dot{\zeta} = f_c(\zeta, u) \tag{16}$$

where the state vector is $\zeta = [x, v_x, y, v_y, \psi, r]$ and input vector is $u = [T_f, T_r, \beta_f, \beta_r]$. For vehicle that has only front wheel drive and front wheel steering, as is considered in this paper, we have $T_r = 0$ and $\beta_r = 0$. Therefore, we have

$$u = [T_f, \beta_f].$$

The nonlinear function $f_c(\zeta, u)$ can be obtained by assembling (7), (8), (9), (11b), (12), (13), (14), and (15). One can also discretize (16) to obtain a discrete-time model, as follows:

$$\zeta_{t+1} = f(\zeta_t, u_t) \tag{17}$$

for discrete sampling time $t$. The discretization can be done, for example, through Euler's method.

### D. AV Path Tracking

For AV path tracking, a path planner generates desired path over the entire prediction horizon, which is then tracked by the MPC as part of the optimization process. In this paper, we consider the similar driving maneuver that was investigated in [9], where the vehicle tracks a sinusoidal trajectory. As shown in Figure 2, the lateral position is a function of the longitudinal position, as given by

$$y = g(x) = 4 \sin \left( \frac{2\pi}{100} x \right) \tag{18}$$

The vehicle is also tracked to maintain desired longitudinal speed.

Depending on the MPC formulation being used, the actual reference going into MPC would be different. For example, an NMPC set up can explicitly deal with the nonlinear relationship between $x$ and $y$ over the prediction horizon, while a LPV-MPC can only track the lateral position $y$ by assuming the constant $v_x$ and $\psi$. Next section details the controller set up for each MPC formulation under investigation.

## IV. Numerical Results and Discussion

Hereafter we will use tNMPC to indicate the time-triggered NMPC, eNMPC the event-triggered NMPC, eNMPC/LPV the event-triggered NMPC with time-triggered LPV-MPC compensation, and finally LPV-MPC indicates the time-triggered LPV-MPC controller.

### A. Controllers Setup

As mentioned in previous section, the control objective here is to track the AV position according to the reference trajectory defined by (18). Additionally, the longitudinal speed in vehicle frame is tracked to a constant value $v_x^r$. Therefore, the cost function for NMPC (both time-triggered and event-triggered) is defined as follows.

$$\begin{aligned} J_N(Z_t, U_t) = & \sum_{k=1}^{p} ||\zeta_{t+k} - \zeta_{t+k}^r||_{Q_\zeta}^2 \\ & + \sum_{k=0}^{p-1} \left( ||u_{t+k} - u_{t+k}^r||_{Q_u}^2 + ||u_{t+k} - u_{t+k-1}||_{Q_d}^2 \right) \\ & + \sum_{k=1}^{p} \left|\left| \zeta_{t+k}(3) - 4 \sin \left( \frac{2\pi}{100} \zeta_{t+k}(1) \right) \right|\right|_{Q_t}^2 \end{aligned} \tag{19}$$

where the last term penalizes the path tracking error and is nonlinear, and $\zeta_{t+k}^r = [0, v_x^r, 0, 0, 0, 0]^T$ for each $k$.

For LPV-MPC, the cost function is given by (4a), where for each $k$, $\zeta_{t+k}^r = [0, v_x^r, y_k^r, 0, 0, 0]^T$ and

$$y_k^r = 4 \sin \left( \frac{2\pi}{100} (\hat{x} + k \hat{v}_x \cos(\hat{\psi}) T_s) \right) \tag{20}$$

where $\hat{\cdot}$ denotes the estimation of the corresponding variable at the beginning of the prediction horizon.

All controllers are evaluated in simulation environment by using the complete bicycle model with wheel dynamics presented in Section III-C. To provide a more realistic simulation analysis, model mismatch is introduced by differing

| Parameter | MPC | Virtual Vehicle |
|---|---|---|
| $m$ $[kg]$ | 1500 | 1425 |
| $L_{xf}$ $[m]$ | 1.2 | 1.3 |
| $L_{xr}$ $[m]$ | 1.4 | 1.3 |
| $I$ $[kgm^2]$ | 4192 | 4402 |
| $R$ $[m]$ | 0.2159 | 0.2159 |
| $C_i$ [-] | -4.5837 | -4.5837 |
| $\mu_i$ [-] | 1 | 0.95 |

TABLE I

PARAMETERS FOR THE BICYCLE MODEL.

| Cals | tNMPC | eNMPC | eNMPC/LPV | LPVMPC |
|---|---|---|---|---|
| $Q_\zeta(2,2)$ | 1 | 1 | 1 | 1 |
| $Q_\zeta(3,3)$ | 0 | 0 | 0 | 1 |
| $Q_t$ | 2 | 2 | 2 | - |
| $Q_u$ | [10,0;0,19] | [10,0;0,19] | [10,0;0,19] | [10,0;0,40] |
| $Q_d$ | [0,0;0,1] | [0,0;0,1] | [0,0;0,1] | [0,0;0,1] |

TABLE II

MPC CALIBRATIONS.

some key parameters used by the MPC and those used by virtual vehicle plant, as summarized in Table I.

All controllers running at a sampling time of $T_s = 200ms$, with prediction horizon $p = 10$, and upper bounds and lower bounds for input constraints given by

$$u_{max} = \begin{bmatrix} 500 \\ 0.54105 \end{bmatrix} \qquad \Delta_{max} = \begin{bmatrix} 70 \\ 0.034907 \end{bmatrix}$$
$$u_{min} = \begin{bmatrix} -500 \\ -0.54105 \end{bmatrix} \qquad \Delta_{min} = \begin{bmatrix} -200 \\ -0.034907 \end{bmatrix}$$

Table II summarizes the calibration of the 4 controllers.

### B. Numerical results

Figure 2 plots the results when the simulation reaches steady state. Please note that for different controllers, the steady state is reached in different $x$ position, and Figure 2 offsets the plot to align at $x = 0$. As can be seen from Figure 2, tNMPC, eNMPC, and eNMPC/LPV provide comparable tracking performance when $-2 \leq y \leq 2$, i.e., when the vehicle is driving almost straight, while eNMPC has notably performance degradation during high curvature maneuver. Note that the results of LPV-MPC is not reported in Figure 2 in order to maintain the readability of the figure.

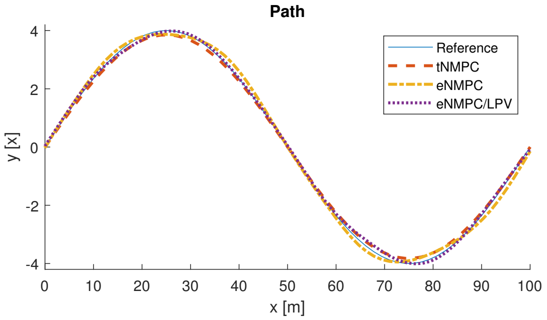

Fig. 2. AV path tracking results.

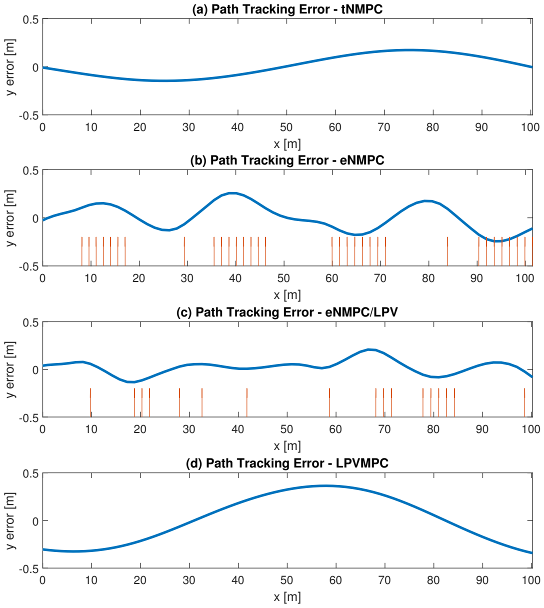

Fig. 3. AV path tracking error.

| Cals | tNMPC | eNMPC | eNMPC/LPV | LPV-MPC |
|---|---|---|---|---|
| Ave. error $[m]$ | 0.111 | 0.133 | 0.077 | 0.252 |
| Max error $[m]$ | 0.173 | 0.256 | 0.208 | 0.364 |
| Ave. $v_x$ $[m/s]$ | 7.98 | 7.86 | 7.86 | 6.75 |
| Ave. $T_s$ $[ms]$ | 200 | 375 | 712 | 200 |

TABLE III

AV PATH TRACKING RESULTS.

Figure 3 plots the path tracking errors, where Figure 3(b) and 3(c) additionally plots the triggering events for eNMPC and eNMPC/LPV. Note that tNMPC, eNMPC, and eNMPC/LPV have comparable worst case tracking error, while eNMPC/LPV has overall best performance. LPV-MPC, on the other hand, has worst control performance in terms of both average and worst case tracking error. Note that the reason eNMPC/LPV outperforms tNMPC in terms of average tracking error is unclear, and additional investigation to confirm this observation is one of the future work. Possible reasons include, for example, different calibrations should be used for time-triggered and event-triggered formulations (in this paper, tNMPC, eNMPC, eNMPC/LPV share the same calibration). The initial guess for tNMPC may also be a cause of local optimum.

Table III summarizes the test results based on Figures 2 and 3. Comparing tNMPC and eNMPC, acceptable performance degradation is observed, while the averaging sampling time is extended from $200ms$ to 375 $ms$, hence relaxing the computation requirement on hardware. Furthermore, eNMPC/LPV triggers nonlinear MPC computation with average interval of $712ms$ between events, while improving average tracking error and degrading maximum error.

Finally, the relative throughputs[1] of tNMPC, eNMPC, and

[1] The throughput data was measured in a desktop computer with Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz CPU and 8.0 GB of RAM.

eNMPC/LPV, scaled by the average computational time of tNMPC throughout the entire sinusoidal period, are compared. Specifically, eNMPC and eNMPC/LPV, when triggered, requires comparable amount of computation time as tNMPC. However, when not triggered, they only require negligible computation. On average, eNMPC and eNMPC/LPV consume 48.88% and 25.66% throughput, respectively, compared to tNMPC.

## V. Conclusion

This paper proposes and compares two different event-triggered nonlinear model predictive controls (NMPC) for autonomous vehicle (AV) path tracking problem. The difference between the two event-triggered NMPCs is, in the first formulation, the last optimal control sequence is used for control action when NMPC is not triggered, while in the second formulation, a linear parametric varying MPC (LPV-MPC) with shorter prediction horizon is used when NMPC is not triggered. The AV path tracking problem considers axle driving torque and front steering input as the control variables. The aforementioned two event-triggered NMPC, together with a time-triggered LPV-MPC and a time-triggered NMPC, are implemented and compared in terms of control performance and throughput requirements, where event-triggered NMPCs are shown to provide benefits of relaxing computational needs while maintaining control performance. Future work includes (1) implementing the proposed eNMPC/LPV in hardware-in-the-loop setting and (2) investigating in energy systems [29]–[31].

## References

[1] D. Q. Mayne, M. M. Seron, and S. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[2] 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.

[3] M. Donkers, W. Heemels, D. Bernardini, A. Bemporad, and V. Shneer, "Stability analysis of stochastic networked control systems," *Automatica*, vol. 48, no. 5, pp. 917–925, 2012.

[4] 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.

[5] A. Bemporad, D. Bernardini, R. Long, and J. Verdejo, "Model predictive control of turbocharged gasoline engines for mass production," SAE Technical Paper, Tech. Rep., 2018.

[6] 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.

[7] Y. Shi, H. D. Tuan, A. V. Savkin, T. Q. Duong, and H. V. Poor, "Model predictive control for smart grids with multiple electric-vehicle charging stations," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 2127–2136, 2018.

[8] B. Cao, B. M. Grainger, X. Wang, Y. Zou, G. F. Reed, and Z.-H. Mao, "Direct torque model predictive control of a five-phase permanent magnet synchronous motor," *IEEE Transactions on Power Electronics*, vol. 36, no. 2, pp. 2346–2360, 2020.

[9] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *2015 IEEE Intelligent Vehicles Symposium*, Seoul, Korea, June 28–July 1, 2015, pp. 1094–1099.

[10] P. Falcone, M. Tufo, F. Borrelli, J. Asgari, and H. E. Tseng, "A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems," in *2007 IEEE Conference on Decision and Control*, New Orleans, LA, Deceber 12–14, 2007, pp. 2980–2985.

[11] F. S. Hoekstra, L. W. Ribelles, H. J. Bergveld, and M. Donkers, "Real-time range maximisation of electric vehicles through active cell balancing using model-predictive control," in *2020 American Control Conference*, Denver, CO, July 1–3, 2020, pp. 2219–2224.

[12] S. Di Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky, "Stochastic MPC with learning for driver-predictive vehicle control and its application to hev energy management," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 3, pp. 1018–1031, 2013.

[13] S. Di Cairano, H. E. Tseng, D. Bernardini, and A. Bemporad, "Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain," *IEEE Trans. Control Syst. Tech.*, vol. 21, no. 4, pp. 1236–1248, 2012.

[14] S. M. M. Jaafari and K. H. Shirazi, "Integrated vehicle dynamics control via torque vectoring differential and electronic stability control to improve vehicle handling and stability performance," *Journal of Dynamic Systems, Measurement, and Control*, vol. 140, no. 7, 2018.

[15] J. Wurts, J. Dallas, J. L. Stein, and T. Ersal, "Adaptive nonlinear model predictive control for collision imminent steering with uncertain coefficient of friction," in *2020 American Control Conference*, Denver, CO, July 1–3, 2020.

[16] 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.

[17] E. Henriksson, D. E. Quevedo, E. G. Peters, H. Sandberg, and K. H. Johansson, "Multiple-loop self-triggered model predictive control for network scheduling and control," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2167–2181, 2015.

[18] 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*, vol. 62, no. 11, pp. 5694–5709, 2017.

[19] H. Li and Y. Shi, "Event-triggered robust model predictive control of continuous-time nonlinear systems," *Automatica*, vol. 50, no. 5, pp. 1507–1513, 2014.

[20] 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*, vol. 63, no. 11, pp. 4008–4015, 2018.

[21] 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*, vol. 65, no. 1, pp. 302–309, 2019.

[22] 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.

[23] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos, "Novel event-triggered strategies for model predictive controllers," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, FL, December 12-15, 2011, pp. 3392–3397.

[24] 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*, vol. 62, no. 1, pp. 177–189, 2016.

[25] F. D. Brunner, M. Heemels, and F. Allgöwer, "Robust self-triggered MPC for constrained linear systems: A tube-based approach," *Automatica*, vol. 72, pp. 73–83, 2016.

[26] G. Cimini and A. Bemporad, "Exact complexity certification of active-set methods for quadratic programming," *IEEE Transactions on Automatic Control*, vol. 62, no. 12, pp. 6094–6109, 2017.

[27] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

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

[29] H. E. Garcia, J. Chen, J. S. Kim, R. B. Vilim, W. R. Binder, S. M. B. Sitton, R. D. Boardman, M. G. McKellar, and C. J. Paredis, "Dynamic performance analysis of two regional nuclear hybrid energy systems," *Energy*, vol. 107, pp. 234–258, 2016.

[30] J. Chen and H. E. Garcia, "Economic optimization of operations for hybrid energy systems under variable markets," *Applied Energy*, vol. 177, pp. 11–24, 2016.

[31] J. S. Kim, J. Chen, and H. E. Garcia, "Modeling, control, and dynamic performance analysis of a reverse osmosis desalination plant integrated within hybrid energy systems," *Energy*, vol. 112, pp. 52–66, 2016.