# Tracking Control Based on Model Predictive Control Using Laguerre Functions with Pole Optimization

Dasol Jeong, Seibum Choi, *Member, IEEE,*

***Abstract*—This paper focused on improving a model predictive control (MPC) using Laguerre functions. This study was conducted to achieve high performance while reducing the computational complexity of MPC for autonomous vehicle tracking control. Previous studies have used a conventional linear time-varying MPC (LTV-CMPC) for the linear time-varying (LTV) vehicle model. For LTV-CMPC, the computational complexity increases exponentially as a predictive horizon and control horizon increase. Real-time implementation of LTV-CMPC with long horizons was difficult due to limited computational resources. For reducing computational complexity, we proposed LTV-MPC using Laguerre functions (LTV-LMPC). Considering the vehicle system, LTV-LMPC used a non-augmented model and described the input rate as Laguerre functions. The proposed LTV-LMPC significantly reduced the number of optimization variables. The number of Laguerre functions and the Laguerre pole determined the performance of the LTV-LMPC. In this study, we derived and proved propositions for analyzing the performance change of LTV-LMPC according to the Laguerre pole. LTV-LMPC with pole optimization (LTV-OLMPC) was proposed based on these propositions. The performance of the proposed algorithm was verified by simulation. The LTV-OLMPC guarantees low computational complexity and high performance.**

***Index Terms*—Autonomous vehicle, Laguerre functions, Laguerre pole optimization, Model predictive control (MPC), Tracking control.**

## I. Introduction

INTELLIGENT transportation systems are attracting attention due to the development of sensors and control technologies [1]. Advanced driver assistance systems, which provide adaptive cruise control [2], [3], and lane-keeping assist [4], [5] assist the driver by reducing some of the tedious parts of driving [6]. Moreover, autonomous vehicle technology has also attracted attention, which improves driving safety, efficiency, and comfort [7]. The core technologies of autonomous driving include perception, decision making, path planning, and tracking control. Among these, only tracking control is directly involved in the vehicle movement. Therefore, a precise tracking control algorithm is essential to ensure the stability of autonomous driving [8].

The primary goal of the tracking control algorithm is to track reference paths. Due to the complexity of vehicle systems, precise tracking control along reference paths is not easy. Vehicle models are essential for accurate tracking control. Vehicle models such as the improved kinematic model [9], path preview model [10], and bicycle model [11] have been used for tracking control. A simple model, a bicycle model, has been used in many previous studies because of its simplicity [12]. For the same reason, the bicycle was used as a vehicle model in this study. The tracking control algorithms have been proposed based on sliding mode control [13], linear quadratic Gaussian control [14], nonlinear adaptive control [15], and model predictive control (MPC) [16]. Among these options, MPC has been used in many tracking control studies because it uses moving horizons and can consider constraints easily [17].

MPC is a well-known and widely used optimal control algorithm. Conventional MPC (CMPC) is robust against model uncertainty and disturbance due to the characteristic of receding horizon control by which it calculates input for each step [18]. It also uses a moving horizon window. Thus, not only regulation control but also tracking control is possible [19]. In the field of autonomous driving, tracking control algorithms have been proposed using CMPC for lane-keeping [20], lane change [21], and collision avoidance [22]. The vehicle model was a time-varying system that changes according to vehicle velocity. Therefore, linear time-varying CMPC (LTV-CMPC) has been proposed and used [8], [23]. For CMPC, the horizon window is determined by the predictive and control horizon. Precise control requires a large prediction horizon and a control horizon. However, it is impossible to use a vaguely large horizon window because of limited computational resources [24]. For this reason, autonomous driving systems using CMPC had no choice but to use a small horizon window. This phenomenon may led to poor control performance.

To overcome this, MPC using Laguerre functions (LMPC) has been proposed. The LMPC had the characteristic that it can drastically reduce the number of optimization variables. Therefore, computational complexity increases in proportion to the square of the horizon in CMPC, whereas it increases in proportion to the horizon in LMPC [19]. These LMPC characteristics have advantages when using a large horizon with limited computational resources. In autonomous driving and vehicle control, LMPC has sometimes been applied [25], [26]. Research on linear time-varying LMPC (LTV-LMPC) has been beginning in various fields [27], [28]. In this study, considering the characteristics of the vehicle model, LTV-LMPC using a non-augmented model and describing the input rate as a Laguerre function was proposed and applied to the tracking control system.

The performance of LTV-LMPC depends on the character-

Dasol Jeong and Seibum Choi are with the Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: oieiaa@kaist.ac.kr; sbchoi@kaist.ac.kr).

istics of the Laguerre functions and the number of Laguerre functions used. LTV-LMPC with many Laguerre functions guaranteed high performance regardless of the Laguerre function used [19]. However, because of the limitation of computational resources, the number of Laguerre functions is also limited. Therefore, for LTV-LMPC with a limited number of Laguerre functions, the tracking control performance varies depending on the characteristics of the Laguerre function utilized [24].

The characteristics of the Laguerre function are determined by the Laguerre pole, which determines the decay rate of the control signal. A low Laguerre pole represents a Laguerre function with a fast decay rate, and a high Laguerre pole represents a Laguerre function with a slow decay rate. Therefore, it is essential to select an appropriate Laguerre pole for the performance of LTV-LMPC. In most previous studies, the Laguerre pole was selected through experimental tuning, and there was no detailed analysis of this. In addition, some previous studies have made efforts to find the optimal Laguerre pole offline, not in real-time [24], [29]. However, the optimal Laguerre pole changes depending on the initial states and references. In the LTV system, the optimal Laguerre pole changes more significantly. Therefore, real-time Laguerre pole optimization is essential to ensure the performance of LTV-LMPC.

Previous studies did not address how to optimize the Laguerre pole in real-time. The complexity of LTV-LMPC made it hard to analyze the effect of the Laguerre pole on LTV-LMPC performance. To overcome this, based on the Laguerre pole characteristics, we derived propositions and corollary that can analyze the performance change of LTV-LMPC according to the Laguerre pole. It enables real-time Laguerre pole optimization. Furthermore, it could be used as a general framework to analyze the performance of LTV-LMPC according to the Laguerre pole.

This study proposed a Laguerre pole optimization algorithm based on the derived propositions and corollary. The gradient descent algorithm was applied as the Laguerre pole adaptation algorithm. The gradient descent algorithm has a local minimum issue. Nevertheless, the gradient descent algorithm has the advantage of being simple to compute. If the computational complexity of the Laguerre pole adaptation algorithm is large, it may be efficient to increase the number of Laguerre functions. In this respect, the gradient descent algorithm was attractive. The local minimum issue using gradient descent was dealt with in detail in Section V. Finally, LTV-LMPC with pole optimization (LTV-OLMPC) was proposed.

The main contributions of this study are as follows: (1) An LTV-LMPC algorithm suitable for the vehicle tracking control using a non-augmented model was proposed. (2) Propositions and corollary to analyze the performance of LTV-LMPC according to Laguerre pole change were derived and proved. (3) A real-time Laguerre pole optimization algorithm and LTV-OLMPC were proposed. The performance was analyzed for the static obstacle avoidance scenario considering the preceding vehicle through CarSim.

The rest of the paper is organized as follows. Section II provides the vehicle dynamics model for tracking control. In Section III, LTV-CMPC for tracking control is introduced. In Section IV, propositions and corollary for analyzing the performance according to the Laguerre pole change are derived and proved. Also, LTV-LMPC and LTV-OLMPC are proposed. The performance validation results by the simulations are provided in Section V. Conclusions and future works are presented in Section VI.

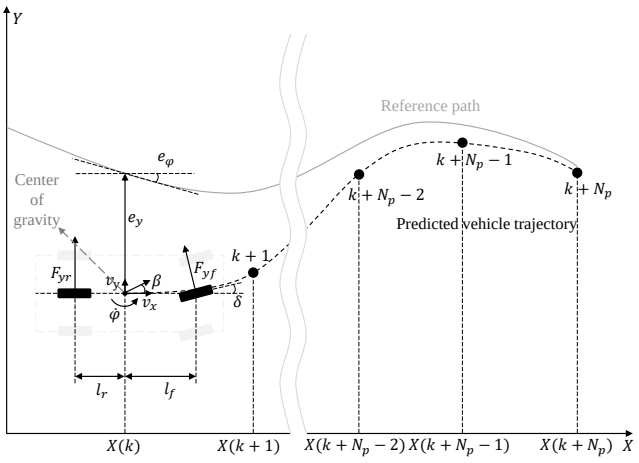


Fig. 1. Model used by model predictive control for tracking control.

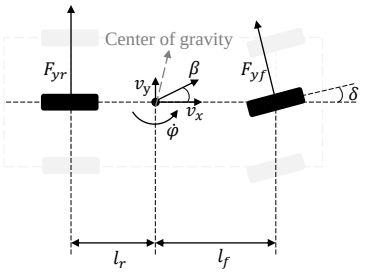


Fig. 2. 2-DOF bicycle model.

## II. Vehicle Dynamic Model for Tracking Control

The vehicle model was used for tracking control. A suitable vehicle model is significant for the design of the tracking control algorithm. Many previous studies which develop control algorithms rather than vehicle modeling have used a bicycle model and a linear tire model [20], [30], [31]. Likewise, this study aimed to develop the advanced model predictive control algorithm. Therefore, the vehicle model consisted of a bicycle model, a linear tire model, and an error dynamics model in this study. The vehicle states and tracking errors are as shown in Fig. 1.

The bicycle model describes the vehicle as a 2-degrees of freedom (DOF) model. The vehicle bicycle models used in this study are as shown in Fig. 2. The vehicle parameters used are as shown in Table I. The force balance equation and the torque balance equation are as follows:

$$\begin{aligned} m_1 a_y &= F_{yf} + F_{yr} \\ I_z \ddot{\varphi} &= l_f F_{yf} - l_r F_{yr} \end{aligned} \tag{1}$$

TABLE I
SPECIFICATIONS OF THE VEHICLE.

| Symbol | Parameter | Value |
|---|---|---|
| $m$ | Mass of the vehicle | $1830\,kg$ |
| $I_z$ | Yaw inertia of the vehicle | $3.77 \times 10^3\,kgm^2$ |
| $l_f$ | Front length of the center of gravity | $1.41\,m$ |
| $l_r$ | Rear length of the center of gravity | $1.64\,m$ |
| $C_f$ | Cornering stiffness of the front axle | $1.47 \times 10^5\,N/rad$ |
| $C_r$ | Cornering stiffness of the rear axle | $1.30 \times 10^5\,N/rad$ |

The kinematic constraints between sideslip angle, yaw rate, velocity, and lateral acceleration are as follows:

$$a_y = \dot{\varphi} v_x + \dot{\beta} v_x \tag{2}$$

The linear tire model was used to describe the tire lateral force. The linear tire model is suitable for describing tire force when the tire slip angle is small. It has advantages in model simplicity and computational complexity because it linearly describes the relationship between tire slip angle and tire lateral force. In this study, the lateral tire force was expressed through cornering stiffness and tire slip angle as follows:

$$\begin{aligned} F_{yf} &= -C_f \alpha_f = -C_f \left( \beta + \frac{l_f}{v_x}\dot{\varphi} - \delta \right) \\ F_{yr} &= -C_r \alpha_r = -C_r \left( \beta - \frac{l_r}{v_x}\dot{\varphi} \right) \end{aligned} \tag{3}$$

Error dynamics were added to describe tracking errors. The tracking errors of the vehicle for the reference path consisted of lateral error and yaw angle error, as shown in Fig. 1. The dynamics of lateral error and yaw angle error are as follows:

$$\begin{aligned} \dot{e}_\varphi &= \dot{\varphi} - \dot{\varphi}_{des} \\ \dot{e}_y &= v_y + v_x e_\varphi \end{aligned} \tag{4}$$

Finally, vehicle dynamics for tracking control, including the bicycle model, the linear tire model, and error dynamics are as follows:

$$\begin{gathered} \text{Let, } \mathbf{x} = \begin{bmatrix} v_y & \dot{\varphi} & e_\varphi & e_y \end{bmatrix}^T, \mathbf{u} = \delta, \mathbf{r} = \dot{\varphi}_{des}, \\ \mathbf{y} = \begin{bmatrix} v_y & \dot{\varphi} & e_\varphi & e_y \end{bmatrix}^T \mathbf{y_{des}} = \begin{bmatrix} 0 & \dot{\varphi}_{des} & 0 & 0 \end{bmatrix}^T \\ \dot{\mathbf{x}} = \mathbf{A_c}\mathbf{x} + \mathbf{B_c}\mathbf{u} + \mathbf{B_{rc}}\mathbf{r}, \\ \mathbf{y} = \mathbf{C_c}\mathbf{x},\ \mathbf{y_{des}} = \mathbf{D_{rc}}\mathbf{r}, \\ \text{where} \end{gathered}$$

$$\mathbf{A_c} = \begin{bmatrix} -\dfrac{C_f + C_r}{m v_x} & \dfrac{-C_f l_f + C_r l_r}{m v_x} - v_x & 0 & 0 \\ \dfrac{-C_f l_f + C_r l_r}{I_z v_x} & -\dfrac{C_f l_f^2 + C_r l_r^2}{I_z v_x} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & v_x & 0 \end{bmatrix},$$

$$\mathbf{B_c} = \begin{bmatrix} \dfrac{C_f}{m} \\ \dfrac{C_f l_f}{I_z} \\ 0 \\ 0 \end{bmatrix}, \mathbf{B_{rc}} = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}, \mathbf{C_c} = \mathbf{I}_{4\times 4}, \mathbf{D_{rc}} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{5}$$

where $v_y$ is the lateral velocity, $\dot{\varphi}$ is the yaw rate, $e_\varphi$ is the yaw angle error, and $e_y$ is the lateral error. Here, $\dot{\varphi}$ is measured directly by the vehicle inertial measurement unit, and it was assumed that $v_y$, $e_\varphi$, and $e_y$ could be estimated. It was also assumed that the cornering stiffness of the front and rear axles are known. The input is the forward steering angle, $\delta$.

When describing vehicle dynamics in this paper, $v_y$ was used instead of sideslip angle. When $v_y$ is used as a state, $\mathbf{B}_c$ can be expressed independently of $v_x$. This has the advantage that $\mathbf{B}_c$ is time-invariant. The vehicle model was converted to a discrete-time system.

$$\begin{gathered} \mathbf{x}(k+1) = \mathbf{A}_k \mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{B_r}\mathbf{r}(k), \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k), \mathbf{y_{des}}(k) = \mathbf{D_r}\mathbf{r}(k), \\ \text{where } \mathbf{A}_k \equiv \mathbf{A}(k) = \mathbf{I}_{4\times 4} + \mathbf{A_c}|_{v_x = v_x(k)} \triangle t \\ \mathbf{B} = \mathbf{B_c}\triangle t,\ \mathbf{B_r} = \mathbf{B_{rc}}\triangle t \\ \mathbf{C} = \mathbf{C_c},\ \mathbf{D_r} = \mathbf{D_{rc}} \end{gathered} \tag{6}$$

where $\triangle t$ is the sampling time. Because $v_x$ varied with time, $\mathbf{A}_k$ was time-varying. And $\mathbf{B}$, $\mathbf{B}_r$, $\mathbf{C}$, and $\mathbf{D}_r$ were time-invariant because they were not affected by $v_x$. Moreover, we assumed that the future velocity profile, $v_x$, is known through path planning [32]. That is, the future system matrix is known.

## III. CONVENTIONAL MODEL PREDICTIVE CONTROL FOR TRACKING CONTROL

In this study, LTV-CMPC was introduced as a path tracking control algorithm. The vehicle model was used to predict future tracking errors. Tracking errors were predicted up to the predictive horizon, and control inputs were optimized up to the control horizon. The cost function consisted of a quadratic form of input and tracking errors. Weight matrices determined the weights between the input and the tracking error. The optimized control inputs over the control horizon satisfied the constraints and minimize the cost function.

### A. Formulation of Path Tracking Control Problem

The purpose of tracking control was to find the optimal control input that minimized the cost function. $\mathbf{Q}$ and $\mathbf{R}$ determined the weights of tracking errors and inputs. A large $\mathbf{Q}$ reduced the tracking error, and a large $\mathbf{R}$ reduced the input rate. The predictive horizon ($N_p$) and control horizon ($N_c$) set the prediction and optimization horizon. In this paper, terminal costs were not considered for algorithm clarity but could easily be included. In path tracking control, the cost function was expressed as in (7).

Model prediction control has the advantage of being able to consider constraints easily [19]. Introducing appropriate constraints is essential for controller implementation and vehicle safety. This study aimed to improve the path tracking algorithm, not to set the constraints precisely. Therefore, commonly used constraints on input rate ($\triangle\delta$), input value ($\delta$), sideslip angle ($\beta = v_y/v_y$), and lateral acceleration ($a_y \approx \dot{\varphi} v_x$) were considered. Path tracking control was transformed into an

optimal control problem using vehicle models, cost functions, and constraints.

$$\begin{aligned}\min_{\mathbf{u}} J = &\sum_{m=1}^{N_p} \|\mathbf{y}(k+m|k) - \mathbf{y_{des}}(k+m)\|_{\mathbf{Q}} \\ &+ \sum_{m=0}^{N_c-1} \|\triangle \mathbf{u}(k+m)\|_{\mathbf{R}} \\ \text{s.t.} \quad &\mathbf{x}(k+1) = \mathbf{A}_k\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{B_r}\mathbf{r}(k) \\ &\triangle\delta_{min} \le \triangle\delta \le \triangle\delta_{max} \\ &\delta_{min} \le \delta \le \delta_{max} \\ &\beta_{min} \le \beta \le \beta_{max} \\ &a_{y,min} \le \dot{\varphi} v_x \le a_{y,max}\end{aligned} \tag{7}$$

Finally, the optimized control input satisfying the above constraints and minimizing the cost function was chosen as the tracking control input.

### *B. Conventional Model Predictive Control for Linear Time-varying System (LTV-CMPC)*

For LTV-CMPC, predicting future tracking errors was done first. A future tracking error that occurred during a predictive horizon is as (8).

Using the prediction model, the cost function of (7) was transformed to (9). The cost function, independent of the input, is expressed as $J_{0C}$.

$$\begin{aligned}&\min_{\mathbf{u}} J = \mathbf{U}^T(\mathbf{\Psi}^T\overline{\mathbf{Q}}\mathbf{\Psi} + \overline{\mathbf{R}})\mathbf{U} \\ &+ \mathbf{2U}^T\left(\mathbf{\Psi}^T\overline{\mathbf{Q}}(\mathbf{F}\mathbf{x}(k) + \mathbf{\Lambda}\mathbf{R_{des}} - \mathbf{Y_{des}}) - \mathbf{G}\mathbf{u}(k-1)\right) \\ &+ J_{0C} \\ &\text{where} \quad J_{0C} = \|\mathbf{F}\mathbf{x}(k) + \mathbf{\Lambda}\mathbf{R_{des}} - \mathbf{Y_{des}}\|_{\overline{\mathbf{Q}}} + \|\mathbf{u}(k-1)\|_{\mathbf{R}} \\ &\mathbf{G} = \begin{bmatrix}\mathbf{R} & \mathbf{0} & \cdots & \mathbf{0}\end{bmatrix}^T, \\ &\overline{\mathbf{Q}} = \begin{bmatrix}\mathbf{Q} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Q}\end{bmatrix}, \overline{\mathbf{R}} = \begin{bmatrix}\mathbf{2R} & \mathbf{-R} & \cdots & \mathbf{0} \\ \mathbf{-R} & \mathbf{2R} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{R}\end{bmatrix}\end{aligned} \tag{9}$$

Constraints can be expressed as (10) through pair of the matrix and vector for inequality constraints. The constraints handling method introduced in [19] was applied to construct $\mathbf{M}_{\text{LTV-CMPC}}$, $\gamma_{\text{LTV-CMPC}}$.

$$\mathbf{M}_{\text{LTV-CMPC}}\mathbf{U} \le \gamma_{\text{LTV-CMPC}} \tag{10}$$

The optimal control input that minimizes the cost function is expressed as (11).

$$\begin{aligned}\mathbf{U} = &-(\mathbf{\Psi}^T\overline{\mathbf{Q}}\mathbf{\Psi} + \overline{\mathbf{R}})^{-1} \\ &\times\left(\mathbf{\Psi}^T\overline{\mathbf{Q}}(\mathbf{F}\mathbf{x}(k) + \mathbf{\Lambda}\mathbf{R_{des}} - \mathbf{Y_{des}}) - \mathbf{G}\mathbf{u}(k-1)\right)\end{aligned} \tag{11}$$

Constraints were satisfied through Hildreth's quadratic programming [19].

$$\hat{\mathbf{U}} = \mathbf{U} + \mathbf{U}_{const} \tag{12}$$

where $\mathbf{U}_{const}$ was modified input to satisfy the constraints. Finally, applying the receding horizon, the optimal control input for each step was given as follows:

$$\mathbf{u}_{\text{LTV-CMPC}}(k) = \hat{\mathbf{U}}(1) \tag{13}$$

## IV. Model Predictive Control Using Laguerre Functions For Tracking Control

The idea behind reducing the computational complexity of LTV-MPC was to describe the future control input as the sum of several orthonormal functions. The $N_c$ future control inputs were described as a linear combination of $N$ orthonormal functions. With this, the optimization variable was reduced from $N_c$ to $N$. Functions, such as Kautz functions [19] and general orthonormal functions [33] have been proposed as orthonormal functions. The Laguerre function has been often used because of its simplicity and good system description. In this study, LTV-LMPC that considered vehicle tracking control characteristics was proposed. The number of Laguerre functions ($N$) and the Laguerre pole ($a$) determine the performance of the LTV-LMPC. However, most studies have excluded analysis of the Laguerre pole. In this study, propositions and corollary about the effect of the Laguerre pole on LTV-LMPC performance were derived and proved. Finally, an LTV-OLMPC was proposed.

$$\mathbf{Y} - \mathbf{Y_{des}} = \mathbf{F}\mathbf{x}(k) + \mathbf{\Psi}\mathbf{U} + \mathbf{\Lambda}\mathbf{R_{des}} - \mathbf{Y_{des}}$$

$$\begin{aligned}\text{where } \mathbf{Y} = \begin{bmatrix}\mathbf{y}(k+1|k) \\ \vdots \\ \mathbf{y}(k+N_p|k)\end{bmatrix} \mathbf{Y_{des}} = \begin{bmatrix}\mathbf{y_{des}}(k+1) \\ \vdots \\ \mathbf{y_{des}}(k+N_p)\end{bmatrix} \mathbf{R_{des}} = \begin{bmatrix}\dot{\varphi}_{des}(k) \\ \vdots \\ \dot{\varphi}_{des}(k+N_p-1)\end{bmatrix} \mathbf{U} = \begin{bmatrix}\mathbf{u}(k) \\ \vdots \\ \mathbf{u}(k+N_c-1)\end{bmatrix} \\ \mathbf{F} = \begin{bmatrix}\mathbf{A}_k \\ \mathbf{A}_{k+1}\mathbf{A}_k \\ \vdots \\ \prod_{m=0}^{N_p-1}\mathbf{A}_{k+m}\end{bmatrix}, \mathbf{\Psi} = \begin{bmatrix}\mathbf{B} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_{k+1}\mathbf{B} & \mathbf{B} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \prod_{m=1}^{N_p-1}\mathbf{A}_{k+m}\mathbf{B} & \prod_{m=2}^{N_p-1}\mathbf{A}_{k+m}\mathbf{B} & \cdots & \prod_{m=N_c}^{N_p-1}\mathbf{A}_{k+m}\mathbf{B}\end{bmatrix}, \\ \mathbf{\Lambda} = \begin{bmatrix}\mathbf{B_r} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_{k+1}\mathbf{B_r} & \mathbf{B_r} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \prod_{m=1}^{N_p-1}\mathbf{A}_{k+m}\mathbf{B_r} & \prod_{m=2}^{N_p-1}\mathbf{A}_{k+m}\mathbf{B_r} & \cdots & \mathbf{B_r}\end{bmatrix}\end{aligned} \tag{8}$$

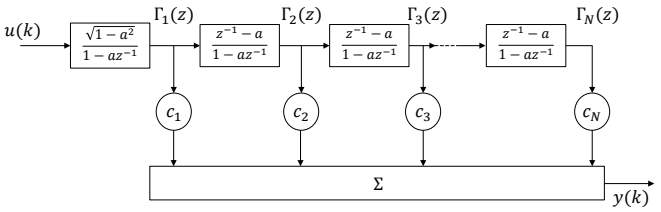


Fig. 3. Laguerre network.

### A. Laguerre Functions

The Laguerre network is determined by the number of functions used ($N$) and the Laguerre pole ($a$). A discrete-time Laguerre network is shown in Fig. 3. The z-transforms of the discrete-time Laguerre networks are as follows:

$$\Gamma_{a,n}(z) = \frac{\sqrt{1-a^2}}{1-az^{-1}} \left( \frac{z^{-1}-a}{1-az^{-1}} \right)^{n-1} \tag{14}$$

where $\Gamma_{a,n}(z)$ is the $n$th Laguerre network. The Laguerre function is defined as the inverse z-transform of the Laguerre network.

$$l_n(k) = Z^{-1}\left(\Gamma_{a,n}(z)\right) \tag{15}$$

The sequence of Laguerre functions can be expressed in the vector form as (16).

$$\mathbf{L}_{a,N}(k) = \begin{bmatrix} l_{a,1}(k) & l_{a,2}(k) & \cdots & l_{a,N}(k) \end{bmatrix}^T \tag{16}$$

The Laguerre function sequence satisfied the following difference equation in state-space form.

$$\begin{gathered} \mathbf{L}_{a,N}(k+1) = \mathbf{Al}_{a,N}\mathbf{L}(k) \\ \text{where } \beta = (1-a^2), \\ \mathbf{L}_{a,N}(0) = \sqrt{\beta}\begin{bmatrix} 1 & -a & \cdots & (-1)^{N-1}a^{N-1} \end{bmatrix}^T \\ \mathbf{Al}_{a,N} = \begin{bmatrix} a & 0 & \cdots & 0 \\ \beta & a & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ (-1)^{N-2}a^{N-2}\beta & (-1)^{N-3}a^{N-3}\beta & \cdots & a \end{bmatrix} \end{gathered} \tag{17}$$

These Laguerre function properties were utilized in the LTV-LMPC configuration and pole optimization.

### B. Model Predictive Control Using Laguerre Functions for Linear Time-Varying System (LTV-LMPC)

In this section, an LTV-LMPC suitable for vehicle tracking control was proposed. Unlike previous studies, this study used the non-augmented model. If the commonly used augmented model is used, the future desired curvature rate ($\ddot{\varphi}_{des}$) is used, which may cause problems such as continuity. Therefore, a non-augmented form was appropriate for the vehicle model. Also, it was hard to describe distant future inputs due to the decaying property of Laguerre functions [19]. Therefore, the future control input rate, not the future control input, was described using Laguerre functions. The proposed LTV-LMPC was analyzed not only for the regulation problem but also for the tracking control. In all formulas, the subscript $N$ represented the result of using $N$ Laguerre functions. The future control input and future control input rates are expressed as (18).

$$\begin{aligned} &\triangle \mathbf{u}_{a,N}(k+m) = \sum_{n=1}^{N} c_{a,n} l_{a,n}(m) = \mathbf{L}_{a,N}(m)^T \eta_{a,N} \\ &\mathbf{u}_{a,N}(k+m) = \mathbf{u}(k-1) + \sum_{i=0}^{m} \mathbf{L}_{a,N}(i)^T \eta_{a,N} \\ &\text{where } \eta_{a,N} = \begin{bmatrix} c_{a,1} & c_{a,2} & \cdots & c_{a,N} \end{bmatrix}^T \end{aligned} \tag{18}$$

where $c_{a,i}$ is the coefficient of the $i$th Laguerre function. The predicted states for the future control input rate are expressed as (19).

$$\begin{aligned} &\mathbf{x}_{a,N}(k+m|k) = \lambda(k,m)^T + \phi_{a,N}(k,m)^T \eta_{a,N} \\ &\text{where} \\ &\lambda(k,m)^T = \prod_{j=0}^{m-1} \mathbf{A}_{k+j} \mathbf{x}(k) \\ &\left[ + \sum_{i=0}^{m-1} \left( \prod_{j=i+1}^{m-1} \mathbf{A}_{k+j} \right) \left( \mathbf{B_r r}(k+i) + \mathbf{Bu}(k-1) \right) \right] \\ &\phi_{a,N}(k,m)^T \\ &= \left\{ \sum_{i=0}^{m-1} \left[ \sum_{p=i+1}^{m} \left( \prod_{j=p}^{m-1} \mathbf{A}_{k+j} \right) \right] \mathbf{B} \mathbf{L}_{a,N}(i)^T \right\} \end{aligned} \tag{19}$$

Also, $\phi_{a,N}(k,m)$ and $\lambda(k,m)$ can be calculated by a convolution sum as (20). These properties reduced computational complexity.

$$\begin{aligned} \phi_{a,N}(k,m)^T =& \mathbf{A}_{k+m-1}\phi_{a,N}(k,m-1)^T + \mathbf{B} \sum_{i=0}^{m-1} \mathbf{L}_{a,N}(i)^T \\ \lambda(k,m)^T =& \mathbf{A}_{k+m-1}\lambda(k,m-1)^T \\ &+ \mathbf{Bu}(k-1) + \mathbf{B_r r}(k+m-1) \end{aligned} \tag{20}$$

Constraints can also be satisfied by LTV-LMPC as follows:

$$\begin{gathered} \mathbf{M}_{a,N}\eta_{a,N} \le \gamma \\ \text{where } m = 0, 1, \cdots, N_p - 1, \\ \mathbf{M}_{a,N} = \begin{bmatrix} \vdots \\ \mathbf{L}_{a,N}(m)^T \\ -\mathbf{L}_{a,N}(m)^T \\ \sum_{i=0}^{m} \mathbf{L}_{a,N}(i)^T \\ -\sum_{i=0}^{m} \mathbf{L}_{a,N}(i)^T \\ \phi_{a,N}(k,m)^T \\ -\phi_{a,N}(k,m)^T \\ \vdots \end{bmatrix}, \gamma = \begin{bmatrix} \vdots \\ \triangle\delta_{max} \\ -\triangle\delta_{min} \\ \delta_{max} - \mathbf{u}(k-1) \\ -\delta_{min} + \mathbf{u}(k-1) \\ \mathbf{x}_{max} - \lambda(k,m)^T \\ -\mathbf{x}_{min} + \lambda(k,m)^T \\ \vdots \end{bmatrix}, \\ \mathbf{x}_{max} = [\beta_{max} v_x(k+m)\, a_{y,max}/v_x(k+m) \infty \infty]^T, \\ \mathbf{x}_{min} = -[\beta_{min} v_x(k+m)\, a_{y,min}/v_x(k+m) \infty \infty]^T \end{gathered} \tag{21}$$

Finally, LTV-LMPC was transformed into quadratic programming (QP) problem, as shown in (22). Furthermore, the

constant cost function independent of the input is expressed as $J_{0L}$.

$$\begin{aligned}
&\min_{\eta_{a,N}} J_{a,N} = \eta_{a,N}^T \mathbf{\Omega}_{a,N} \eta_{a,N} + 2\eta_{a,N}^T \mathbf{\Phi}_{a,N} + J_{0L} \\
&\text{s.t.} \qquad \mathbf{M}_{a,N}\eta_{a,N} \leq \gamma \\
&\text{where } J_{0L} = \sum_{m=1}^{N_p-1} \left(\lambda(k,m)^T - \mathbf{y_{des}}\right)^T \mathbf{Q} \left(\lambda(k,m)^T - \mathbf{y_{des}}\right) \\
&\mathbf{\Omega}_{a,N} = \sum_{m=1}^{N_p-1} \phi_{a,N}(k,m)\mathbf{Q}\phi_{a,N}(k,m)^T + \mathbf{R}_{\mathbf{L},N} \\
&\mathbf{\Phi}_{a,N} = \sum_{m=1}^{N_p-1} \phi_{a,N}(k,m)\mathbf{Q} \left(\lambda(k,m)^T - \mathbf{y_{des}}\right) \\
&\mathbf{R}_{\mathbf{L},N} = \mathbf{R} \times \mathbf{I}_{N\times N}
\end{aligned} \tag{22}$$

The optimal control input that minimizes the cost function is as follows:

$$\eta_{a,N} = -\mathbf{\Omega}_{a,N}^{-1} \mathbf{\Phi}_{a,N} \tag{23}$$

The constraints were satisfied using Hildreth's quadratic programming. The constrained optimal input is expressed as (24). These expressions were used in pole optimization.

$$\begin{aligned}
\hat{\eta}_{a,N} &= \eta_{a,N} - \mathbf{\Omega}_{a,N}^{-1} \mathbf{M}_{a,N,act}^T \lambda_{a,N,act} \\
\text{where } \lambda_{a,N,act} &= -\mathbf{H}_{a,N,act}^{-1} \mathbf{K}_{a,N,act} \\
\mathbf{H}_{a,N,act} &= \mathbf{M}_{a,N,act} \mathbf{\Omega}_{a,N}^{-1} \mathbf{M}_{a,N,act}^T \\
\mathbf{K}_{a,N,act} &= \gamma_{act} + \mathbf{M}_{a,N,act} \mathbf{\Omega}_{a,N}^{-1} \mathbf{\Phi}_{a,N}
\end{aligned} \tag{24}$$

The subscript "act" in (24) means the active constraint set [19]. Finally, the optimal control input through LTV-LMPC is described as follows:

$$\begin{aligned}
\triangle \mathbf{u}_{\text{LTV-LMPC}}(k) &= \mathbf{L}_{a,N}(0)^T \hat{\eta}_{a,N} \\
\mathbf{u}_{\text{LTV-LMPC}}(k) &= \mathbf{u}(k-1) + \mathbf{L}_{a,N}(0)^T \hat{\eta}_{a,N}
\end{aligned} \tag{25}$$

In this section, a tracking control algorithm through LTV-LMPC was proposed. The proposed LTV-LMPC was suitable for the vehicle tracking controller.

### *C. Minimum Cost Function for Various Laguerre Poles*

In this section, the minimum cost function was analyzed according to the Laguerre pole. The optimal Laguerre pole minimizes the minimum cost function. In the LTV system, the optimal Laguerre pole changes according to the initial condition, reference path, and velocity. The minimum cost function analysis was preceded to analyze the effect of the Laguerre pole. The minimum cost function was defined as the value obtained by substituting the optimal control input of (24) into the cost function in (22). The minimum cost function for LTV-LMPC is as follows:

$$\begin{aligned}
&J_{min,a,N} \\
&= -\mathbf{\Phi}_{a,N}^T \mathbf{\Omega}_{a,N}^{-1} \mathbf{\Phi}_{a,N} + \mathbf{K}_{a,N,act}^T \mathbf{H}_{a,N,act}^{-1} \mathbf{K}_{a,N,act} + J_{0L}
\end{aligned} \tag{26}$$

The effect of the Laguerre pole on LTV-LMPC performance was analyzed through simple simulations. The vehicle model, defined in Section II, was used for the simulation. Here, $\triangle t = 20ms$, $N_p = 100$, $\dot{\varphi}_{des} = 0$, $\mathbf{x}(k) = [0,0,0,4]^T$. For two velocities ($v_x = 60$ and $90\,km/h$) and four different numbers of Laguerre functions ($N = 4, 6, 8,$ and $10$), the minimum cost function values were compared according to the Laguerre pole. The simulation results are as shown in Fig. 4.

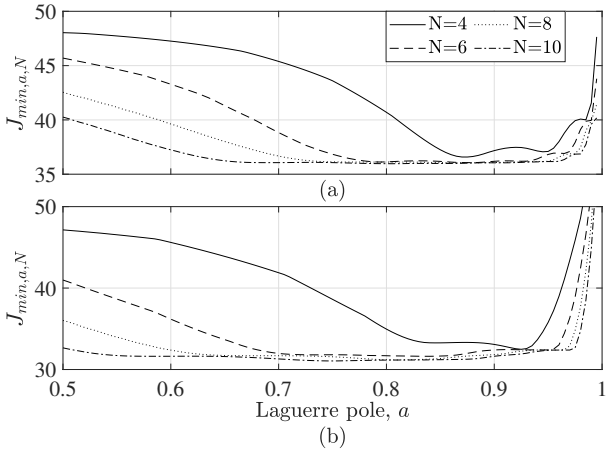


Fig. 4. Minimum cost function for various numbers of Laguerre functions and Laguerre poles. (a) Simulation at $60km/h$. (b) Simulation at $90km/h$.

As shown in Fig. 4, as the number of Laguerre functions increases, the minimum cost function decreases. For a large $N$, the properly tuned Laguerre pole did not degrade LTV-LMPC performance. However, when it was not possible to use a large $N$ due to the limitation of the computational resources, the Laguerre pole affected performance. As shown in Fig. 4, the incorrectly selected Laguerre pole for a small $N$ caused performance degradation. In addition, there were different optimal Laguerre poles depending on the velocity, as shown in Fig. 4(a) and 4(b).

For a small $N$, selection of the optimal Laguerre pole is essential, and this value changes with the velocity. Furthermore, the optimal Laguerre pole changes with the initial condition and the reference path. Therefore, to guarantee the performance of LTV-LMPC using a small $N$, an algorithm that optimizes the Laguerre pole in real-time is required.

### *D. Model Predictive Control Using Laguerre Functions with Pole Optimization for Linear Time-Varying System (LTV-OLMPC)*

This section introduces an adaptation algorithm for optimizing the Laguerre pole in real-time. The partial derivative of the minimum cost function for the Laguerre pole at the optimal Laguerre pole is zero, as shown in (27).

$$\left.\frac{\partial J_{min,a,N}}{\partial a}\right|_{a=a_{opt}} = 0 \tag{27}$$

It's best to find all analytic solutions that make the above equations zero, but it's not easy. Because $J_{min,a,N}$ was a high-order function for $a$. For this reason, in previous studies, the Laguerre pole was experimentally tuned offline. However, if the effect of the Laguerre pole on LTV-LMPC performance is analyzed, it is also possible to optimize the Laguerre

pole in real-time. In this study, propositions and corollary for analyzing LTV-LMPC performance change according to Laguerre pole change were derived and proved. The property of the Laguerre function used for derivation is as shown in Theorem IV.1 [34]. Also, the proposed algorithm uses $N+1$ Laguerre functions. Of these, the $N$ Laguerre functions were used to find the optimal control input, and one Laguerre function was used to optimize the Laguerre pole. The subscript $N+1$ (ex. $\mathbf{\Phi}_{a,N+1}$) indicates a value calculated using $N+1$ Laguerre functions. Also, the subscript $N$ (ex. $\mathbf{\Phi}_{a,N}$) indicates a value calculated using $N$ Laguerre functions.

**Theorem IV.1.** *For the $n$th order Laguerre function with the Laguerre pole, $a$, the following condition holds:*

$$\frac{\partial l_{a,n}(m)}{\partial a} = \frac{1}{1-a^2}[n l_{a,n+1}(m) - (n-1) l_{a,n-1}(m)] \quad (28)$$

The following proposition was derived based on the Laguerre function property, as in Theorem IV.1.

**Proposition 1.** *For any $k$, $m$, the partial derivative of $\phi_{a,N}(k,m)$, derived from $N$ Laguerre functions, for $a$ is expressed as a linear combination of $\phi_{a,N+1}(k,m)$, derived from $N+1$ Laguerre functions, as follows:*

$$\frac{\partial}{\partial a}\phi_{a,N}(k,m)^T = \phi_{a,N+1}(k,m)^T \mathbf{T}_a^T$$
$$\textit{where } \mathbf{T}_a = \frac{1}{1-a^2}\begin{bmatrix} 0 & 1 & \cdots & 0 & 0 & 0 \\ -1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & N-1 & 0 \\ 0 & 0 & \cdots & -(N-1) & 0 & N \end{bmatrix} \quad (29)$$

*Proof.* Please refer to Appendix VI-A. □

A consequence of Proposition 1 is the statement of the following corollary.

**Corollary 1.** *The partial derivatives of $\mathbf{\Omega}_{a,N}$, $\mathbf{\Phi}_{a,N}$, and $\mathbf{M}_{a,N}$, derived by N Laguerre functions, for $a$ are expressed as follows:*

$$\begin{aligned} \frac{\partial}{\partial a}\mathbf{\Omega}_{a,N} =& \mathbf{T}_a(\mathbf{\Omega}_{a,N+1} - \mathbf{R}_{\mathbf{L},N+1})\mathbf{E}^T \\ &+ \mathbf{E}(\mathbf{\Omega}_{a,N+1} - \mathbf{R}_{\mathbf{L},N+1})\mathbf{T}_a^T \\ \frac{\partial}{\partial a}\mathbf{\Phi}_{a,N} =& \mathbf{T}_a\mathbf{\Phi}_{a,N+1} \\ \frac{\partial}{\partial a}\mathbf{M}_{a,N}^T =& \mathbf{T}_a\mathbf{M}_{a,N+1}^T \\ \textit{where } \mathbf{E} =& \begin{bmatrix}\mathbf{I}_{N\times N} & \mathbf{0}_{N\times 1}\end{bmatrix} \end{aligned} \quad (30)$$

Corollary 1 expresses the partial derivative of matrices, derived from LTV-LMPC, for $a$. The $N+1$ Laguerre functions were required for the partial derivative of $N$ Laguerre functions. Using Corollary 1, the partial derivative of $J_{min,a,N}$ for $a$ is expressed as the following proposition:

**Proposition 2.** *For constrained LTV-LMPC, the partial derivatives of $J_{min,a,N}$, the minimum cost function of LTV-LMPC, for $a$ is expressed as follows:*

$$\begin{aligned} \frac{\partial J_{min,a,N}}{\partial a} =& \hat{\eta}_{a,N}^T\left(\mathbf{T}_a\mathbf{\Omega}_{a,N+1}\mathbf{E}^T + \mathbf{E}\mathbf{\Omega}_{a,N+1}\mathbf{T}_a^T\right)\hat{\eta}_{a,N} \\ &+2\hat{\eta}_{a,N}^T\left(\mathbf{T}_a\mathbf{\Phi}_{a,N+1} + \mathbf{T}_a\mathbf{M}_{a,N+1,act}^T\lambda_{a,N,act}\right) \end{aligned} \quad (31)$$

*Proof.* Please refer to Appendix VI-B. □

Such propositions and corollary enable an analysis of the performance of LTV-LMPC according to the Laguerre pole. In previous studies, it has been observed that performance changes as the Laguerre pole changes. However, the analysis was insufficient due to the absence of the above relational expression. The derived propositions and corollary were used in the Laguerre pole optimization algorithm.

The optimal Laguerre pole that satisfies (27) was adapted using the gradient descent method as follows:

$$a_{adap}(k+1) = a_{adap}(k) - \omega\left.\frac{\partial J_{min,N}}{\partial a}\right|_{a=a_{adap}(k)} \quad (32)$$

where $\omega$ is the weighting coefficient and has a positive constant. The gradient descent method has a local minimum issue. Despite this, the gradient descent method has the advantage of having low computational complexity. If the computational complexity of the Laguerre pole optimization algorithm is high, simply increasing the number of Laguerre functions may be efficient, so this advantage is attractive. Also, in the case of the used LTV-LMPC, the global minimum point changes with time, and the cost function of the local minimum and the global minimum do not differ significantly. Therefore, the gradient descent method was applied. The local minimum issue of the gradient descent method was discussed in detail in Section V.

Finally, the control input was optimized using the optimized Laguerre pole as (33).

$$\begin{aligned} \triangle\mathbf{u}_{\text{LTV-OLMPC}}(k) &= \mathbf{L}_{a_{adap},N}(0)^T\eta_{a_{adap},N} \\ \mathbf{u}_{\text{LTV-OLMPC}}(k) &= \mathbf{u}(k-1) + \mathbf{L}_{a_{adap},N}(0)^T\eta_{a_{adap},N} \end{aligned} \quad (33)$$

The complete algorithm of LTV-OLMPC is summarized in Alg. 1.

## V. Simulation Results

### A. Simulation Scenarios

The performance of the proposed algorithm was verified by simulations using CarSim. The driving scenario was the avoidance of static obstacles considering the preceding vehicle. The ego vehicle was chosen as a general passenger vehicle. Velocity profile and reference curve were made based on previous path planning studies based on discrete optimization [35], [32]. The velocity profile and reference curve are shown in Fig. 5. The scenarios consisted of (1) static obstacle avoidance ($\sim 600\,m, \sim 25\,sec$), (2) following the preceding car ($600 \sim 950\,m, 25 \sim 45\,sec$), and (3) lane change to the initial lane ($950\,m \sim, 45\,sec \sim$). Since this study deals with

**Algorithm 1** Algorithm for the proposed LTV-OLMPC

**Input:** $\mathbf{x}(k)$, $v_x(k)$,$\cdots$,$v_x(k+N_p-1)$, $\dot{\varphi}_{des}(k)$,$\cdots$,$\dot{\varphi}_{des}(k+N_p-1)$, $\mathbf{u}(k-1)$, $a_{adap}(k-1)$.
**Output:** $\mathbf{u}(k)$, $a_{adap}(k)$.
*Initialisation* : $\mathbf{Q}, \mathbf{R}, \omega > 0$, $N, N_p \in \mathbf{Z}$, $\mathbf{u}(0) = 0$, $a_{adap}(0) \in (0.7, 1)$, constraints.
$a \leftarrow a_{adap}(k-1)$
compute $\mathbf{A}_k, \cdots, \mathbf{A}_{k+N_p-1}, \mathbf{B}, \mathbf{B_r}$ using (5), (6)
compute $\mathbf{L}_{a,N}(0), \mathbf{Al}_{a,N}$ using (17)
**for** $m = 1$ to $N_p - 1$ **do**
  $\mathbf{L}_{a,N}(k+m) \leftarrow \mathbf{Al}_{a,N}\mathbf{L}(k+m-1)$
  convolution sum: $\phi_{a,N}(k,m), \lambda(k,m)$ using (20)
**end for**
compute using $\mathbf{\Omega}_{a,N}, \mathbf{\Phi}_{a,N}, \mathbf{M}_{a,N}, \gamma$ (21), (22)
$\eta_{a,N} \leftarrow -\mathbf{\Omega}_{a,N}^{-1}\mathbf{\Phi}_{a,N}$
**if** all states and inputs satisfy constraints **then**
  $\hat{\eta}_{a,N} \leftarrow \eta_{a,N}$
  $\mathbf{T}_a\mathbf{M}_{a,N+1,act}^T\lambda_{a,N,act} \leftarrow 0$
**else**
  solve $\lambda_{a,N,act}$ using Hildreth's quadratic programming [19]
  $\hat{\eta}_{a,N} \leftarrow \eta_{a,N} - \mathbf{\Omega}_{a,N}^{-1}\mathbf{M}_{a,N,act}^T\lambda_{a,N,act}$
**end if**
$\mathbf{u}(k) \leftarrow \mathbf{u}(k-1) + \mathbf{L}(0)^T\hat{\eta}_{a,N}$
compute $\partial J_{min,a,N}/\partial a$ using (31)
$a_{adap}(k) \leftarrow a - \omega \times \partial J_{min,a,N}/\partial a$

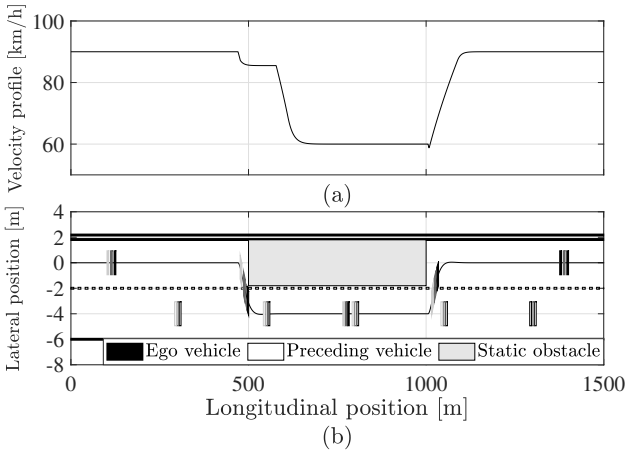


Fig. 5. Simulation scenario. (a) Velocity profile. (b) Reference curve.

the lateral tracking controller, the longitudinal tracking controller for all algorithms was designed through conventional PID control. The vehicle parameters and LTV-MPC tuning parameters are shown in Tables I and II.

The comparison algorithms used were LTV-CMPC, LTV-LMPC, and LTV-OLMPC. Several optimization variables were used for each LTV-MPC algorithm. The number of optimization variables was the control horizon ($N_c$) for LTV-CMPC, and the number of Laguerre functions ($N$) for LTV-LMPC and LTV-OLMPC. The algorithms were evaluated using two benchmarks. The first benchmark evaluated the tracking performance of algorithms. Benchmark 1 was defined as the

TABLE II
TUNING PARAMETERS OF THE TRACKING CONTROL.

| Symbol | Parameter | Value |
|---|---|---|
| $\triangle t$ | Sampling time | $20\,ms$ |
| $\mathbf{Q}$ | Weight matrix for errors | $\mathbf{I_{4\times 4}}$ |
| $\mathbf{R}$ | Weight matrix for input rate | $1/\triangle t^2$ |
| $N_p$ | Predictive horizon | 100 |
| $N_c$ | Control horizon for LTV-CMPC | $20 \sim 100$ |
| $N$ | Number of Laguerre functions | $3 \sim 11$ |
| $a$ | Laguerre pole | $0.7 \sim 1$ |
| $\omega$ | Step size of the pole adaptation | $1.5 \times 10^{-3}$ |
| $\triangle\delta_{max}, \triangle\delta_{min}$ | Constraints of input rate | $\pm 180\,deg/sec$ |
| $\delta_{max}, \delta_{min}$ | Constraints of input value | $\pm 360\,deg$ |
| $\beta_{max}, \beta_{min}$ | Constraints of sideslip angle | $\pm 1\,deg$ |
| $a_{y,max}, a_{y,min}$ | Constraints of lateral acceleration | $\pm 4\,m/s^2$ |

correlation coefficient between the input rate sequence over the simulation of LTV-CMPC with $N_c = N_p = 100$ and the input rate sequence over the simulation of various LTV-MPC algorithms. The input rate sequence through LTV-CMPC with $N_c = N_p = 100$ was the most optimal input rate solution. Because it optimized all control input during the predictive horizon. Therefore, as the correlation value is closer to 1, the input rate sequence of the corresponding LTV-MPC is close to optimal.

$$\text{Benchmark 1: corr}(\overline{\triangle\mathbf{U}}_{\text{LTV-CMPC},N_c=N_p}, \overline{\triangle\mathbf{U}}) \tag{34}$$

where $\overline{\triangle\mathbf{U}}_{\text{LTV-CMPC},N_c=N_p}$ is the input rate sequence vector over the simulation of LTV-CMPC with $N_c = N_p = 100$, $\overline{\triangle\mathbf{U}}$ is the input rate sequence vector over the simulation of each LTV-MPC.

Benchmark 2 was defined as the maximum floating-point operation (FLOP) for each step. FLOP is an objective indicator of computational complexity and has been used to analyze computational complexity [36], [37]. In addition, FLOP can determine whether an algorithm is a real-time implementation. The on-board CPU of autonomous vehicles recently supports 1 Giga FLOP (GLOP) [38]. Assuming that 10% of the computational resource is used by the tracking controller, for a sampling rate of $20\,ms$, it can be calculated that the maximum allowable FLOP is 2 Mega FLOP (2 MFLOP) as follows:

$$\text{Maximum allowed FLOP} = 0.1 \times \frac{1\,\text{GFLOP}}{1\,sec}\frac{1\,sec}{50\,samples} = 2\,\text{MFLOP} \tag{35}$$

The performance-to-computational complexity of the algorithm was analyzed using benchmark 1 and benchmark 2.

### *B. Simulation Results*

The performance of each LTV-MPC algorithm is shown in Table III. The performance of each LTV-MPC algorithm is also shown in Fig. 6, in which the x-axis represents benchmark 1 and the y-axis represents benchmark 2. For LTV-MPC algorithms, it is better to have a large benchmark 1 and a small benchmark 2. In other words, the algorithm on the lower-right corner means better performance-to-computational complexity. As a results, it was verified that LTV-OLMPC, LTV-LMPC, and LTV-CMPC have high performance-to-computational complexity in that order.

TABLE III
PERFORMANCES OF TRACKING CONTROLLERS THROUGH SIMULATION.

| | $N_c/N$ | 20/3 | 30/4 | 40/5 | 50/6 | 60/7 | 70/8 | 80/9 | 90/10 | 100/11 |
|---|---|---|---|---|---|---|---|---|---|---|
| Benchmark 1 | LTV-CMPC | 0.4513 | 0.7166 | 0.9431 | 0.9732 | 0.9854 | 0.9959 | 0.9992 | 0.9998 | 1 |
| [-] | LTV-LMPC | 0.4323 | 0.8111 | 0.9379 | 0.9545 | 0.9685 | 0.9790 | 0.9897 | 0.9955 | 0.9966 |
| | LTV-OLMPC | 0.8607 | 0.9350 | 0.9611 | 0.9739 | 0.9831 | 0.9867 | 0.9949 | 0.9974 | 0.9980 |
| Benchmark 2 | LTV-CMPC | 46.9827 | 66.0467 | 88.1043 | 113.4597 | 142.4167 | 175.2795 | 212.3519 | 253.9381 | 300.0251 |
| [MFLOP] | LTV-LMPC | 1.2900 | 1.7590 | 2.6611 | 4.2734 | 5.6425 | 8.1735 | 12.3389 | 15.3449 | 18.3553 |
| | LTV-OLMPC | 0.9050 | 1.4327 | 3.0412 | 5.3688 | 7.4141 | 10.2388 | 13.3419 | 16.9162 | 20.6802 |

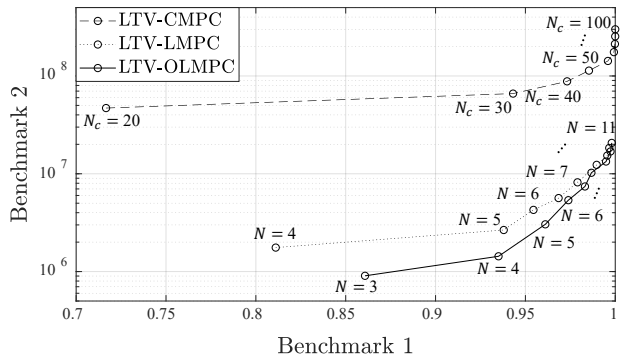


Fig. 6. Performance of each tracking control algorithm with defined benchmarks.

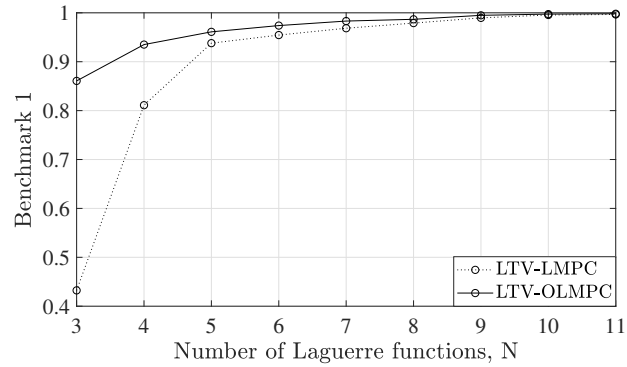


Fig. 7. Tracking control performance of LTV-LMPC and LTV-OLMPC for various numbers of the Laguerre functions.

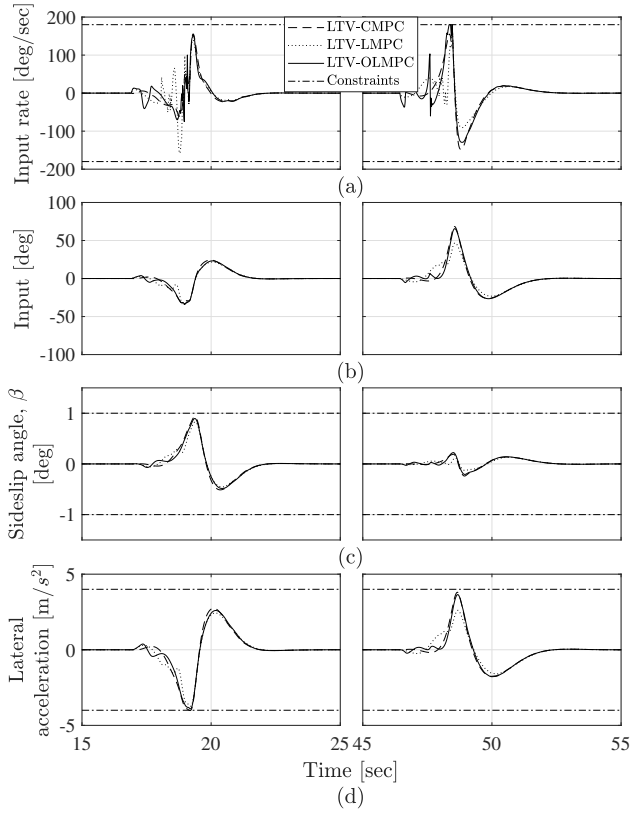


Fig. 8. Simulation results. (a) Optimized input rate, steering wheel angle rate. (b) Optimized input, steering wheel angle. (c) Sideslip angle. (d) Lateral acceleration.

The difference in benchmark 1 for LTV-LMPC and LTV-OLMPC according to N is shown in Fig. 7. For large $N$ ($N =$ 8 to 10), there is no significant difference in performance because the performance has already been saturated. However, for small $N$ ($N = 3$ to 7), there is a difference in performance depending on the pole optimization. Hence, the proposed algorithm has an advantage for systems with limited computational resources.

Next, the tracking control results are analyzed. LTV-CMPC with $N_c = 100$, whose inputs were optimized for all predictive horizons, was used as the first comparison group. Although this gives the best performance, real-time implementation is impossible due to excessive computational complexity. LTV-LMPC and LTV-OLMPC with $N = 4$, which are possible for real-time implementation (benchmark $2 < 2$ MFLOP), were compared and analyzed. The analysis focused on static obstacle avoidance ($15 \sim 25\,sec$) and lane change to initial lane ($45 \sim 55\,sec$) situations. In Figs. 8 - 11, the graph on the left shows the static obstacle avoidance situation, and on the right shows the lane change to the initial lane situation. Moreover, the dashed line indicates LTV-CMPC with $N_c = 100$, the dotted line indicates LTV-LMPC with $N = 4$, and the solid line indicates LTV-OLMPC with $N = 4$.

The input rate, input, sideslip angle, and lateral acceleration are as shown in Fig. 8. The dash-dotted line indicates constraints. As shown in Fig. 8, all inputs and states satisfied the constraints. As shown in Fig. 8(a) and 8(b), for an LTV-LMPC, the control input rate and input were disparate from an LTV-CMPC. On the other hand, an LTV-OLMPC had the control input rate and input similar to an LTV-CMPC. Likewise, as shown in 8(c) and 8(d), an LTV-OLMPC showed a closer sideslip angle and lateral acceleration to an LTV-CMPC.

The longitudinal velocities and trajectories resulting from the tracking control are as shown in Fig. 9. As shown in Fig. 9(a), the longitudinal velocities were the same control result

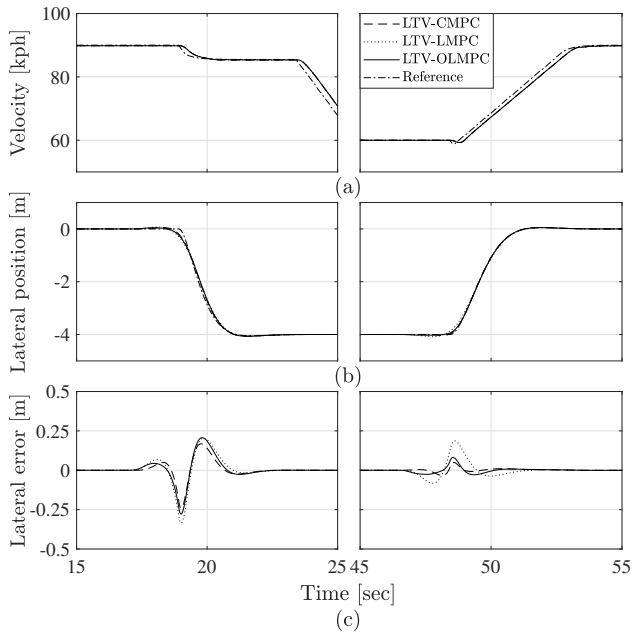


Fig. 9. Longitudinal velocity and trajectory resulting from the tracking control. (a) Longitudinal velocity. (b) Lateral position. (c) Lateral tracking error.

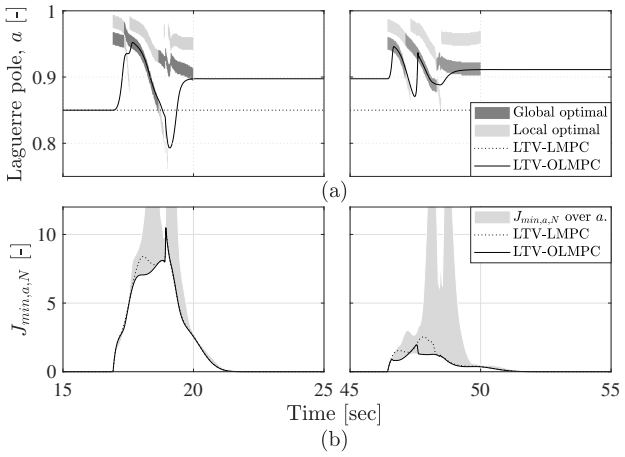


Fig. 10. The Laguerre pole for each step and the corresponding minimum cost function. (a) Laguerre pole ($a$). (b) Minimum cost function ($J_{min,a,N}$).

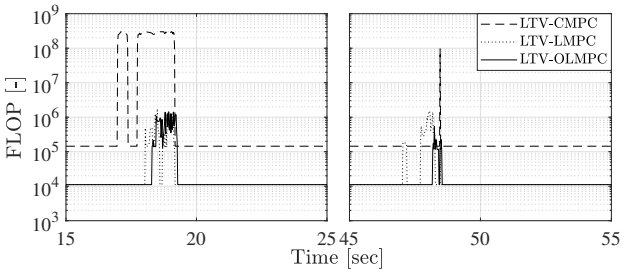


Fig. 11. FLOP of various algorithms for each step.

because the PID controller was the same in all algorithms. The lateral position and tracking error are as shown in Fig. 9(b) and 9(c). The lateral position and tracking error of LTV-LMPC showed some difference from that of LTV-CMPC. On the other hand, the trajectory of LTV-OLMPC was almost similar to that of LTV-CMPC. In other words, the proposed LTV-OLMPC had a similar performance to that of the LTV-CMPC using 100 control horizons, even though only using four Laguerre functions.

Fig. 10(a) shows the optimized Laguerre pole for each step. As shown in Fig. 4, the minimum cost function according to the Laguerre pole had several local minimums, including the global minimum. In Fig. 10(a), the black region indicates the region of global minimum and the gray region indicates the region of local minimum, which is the second minimum. These are the iteratively calculated results of changing the Laguerre pole offline. As shown in Fig. 10(a), the global minimum changed over time. This provided an opportunity for the Laguerre pole to escape from the local minimum to the global minimum. Simulation results show that the Laguerre pole escaped from the local minimum to the global minimum region at about $19, 47$, and $49\,sec$.

Fig. 10(b) shows the minimum cost function ($J_{min,a,N}$) according to the Laguerre pole ($a$). The gray region shows the distribution of the minimum cost function as the Laguerre pole changes from $0.7$ to $1$. For LTV-LMPC with the fixed Laguerre pole, the minimum cost function was much different from the global minimum. On the other hand, for LTV-OLMPC with the optimized Laguerre pole, the minimum cost function was close to the global minimum. Due to these parts, a difference in control performance between LTV-LMPC and LTV-OLMPC occurred. Like this, the proposed Laguerre pole optimization algorithm converged to a global minimum over time due to the characteristics of LTV-LMPC, despite using the gradient descent method. Also, even in the local minimum, there is no significant performance difference from the global minimum.

The computational complexity of each algorithm is shown in Fig. 11. LTV-LMPC and LTV-OLMPC were about 200 times lower computational complexity than LTV-CMPC. Also, the proposed LTV-OLMPC had computational complexity similar to that of LTV-LMPC because it did not require a high computational complexity for the Laguerre pole optimization algorithm.

### C. Discussions

The performance of the proposed tracking control algorithm was verified through simulation using CarSim. The three LTV-MPC algorithms used in the comparison have the following characteristics: LTV-CMPC was a good solution when computational resources were sufficient. However, as the predicted and control horizons increased, the computational complexity increased exponentially. Therefore, high performance could not be guaranteed for a system with limited computation resources. LTV-LMPC had less computational complexity compared to LTV-CMPC. However, when using a small number of Laguerre functions, it was difficult to ensure high performance because the Laguerre pole was not optimized. The proposed LTV-OLMPC optimized the Laguerre pole in real-time. It guaranteed high performance, even when using a small number of Laguerre functions.

However, as shown in Fig. 8, the input rate and input of LTV-LMPC and LTV-OLMPC were noisy and oscillated. The oscillation was caused by minimizing the cost function while satisfying the constraints through a small number of optimization variables. It was seen not only in LTV-LMPC and LTV-OLMPC but also in LTV-CMPC with a small control horizon $(N_c)$. These oscillations of input rates and input could cause mechanical loss of the actual plant. The following ways can reduce oscillations. First, add the term of the second derivative of the input ($\triangle^2\mathbf{u}(k+m)$) in the cost function in (7). Second, increase the number of Lager functions when more computational resources are available. Third, eliminate noise through post-processing such as a low-pass filter. Reduction of oscillation of input or input rate through these methods has advantages when applied to actual plants.

## VI. Conclusion and Future Work

In this paper, we proposed three LTV-MPC algorithms for tracking control. As a prediction model, a general vehicle model incorporating the bicycle model and the linear tire model was used. LTV-CMPC used many optimization variables up to the control horizon. The proposed LTV-LMPC significantly reduced the number of optimization variables by expressing the control input as a sum of Laguerre functions. The Laguerre pole determines the performance of the LTV-LMPC. Propositions and corollary were derived for analyzing the performance change according to the Laguerre pole change. The Laguerre pole optimization algorithm was proposed based on propositions and corollary. Finally, LTV-OLMPC was proposed. The performance of the proposed algorithm was verified using two benchmarks. As a result, LTV-OLMPC, LTV-LMPC, and LTV-CMPC have high performance-to-computational complexity in that order. The proposed algorithm has the following advantages. First, it can significantly reduce the computational complexity of vehicle tracking control. Conversely, longer horizons are available for limited computational resources. Second, the proposed algorithm can be used not only for vehicle systems but also for systems with limited computational resources.

In future studies of autonomous vehicle control, the LTV-OLMPC can apply in the following ways. First, developing combined longitudinal and lateral tracking controllers. The proposed LTV-OLMPC can extend to a multi-input multi-output (MIMO) system. LTV-OLMPC, extended with a MIMO system, can combine longitudinal and lateral tracking controllers. Second, application to nonlinear vehicle models (planar vehicle model or nonlinear tire models). We proposed LTV-OLMPC in the form of linear MPC. Therefore, it is hard to directly apply the LTV-OLMPC to a nonlinear MPC (NMPC) for a nonlinear model. However, some recent studies aimed to study algorithms that convert NMPC to LTV-MPC using a linearization technique. For the converted LTV-MPC problem, the proposed LTV-OLMPC is applicable. In these ways, LTV-OLMPC can apply to various fields of vehicle control as a way to reduce computational complexity.

## Acknowledgment

This research was partly supported by the grant(21TLRP-C152499-03) from Transportation & Logistics Research Program funded by Ministry of Land, Infrastructure and Transport(MOLIT) of Korea government and Korea Agency for Infrastructure Technology Advancement(KAIA)., the BK21 FOUR Program of the National Research Foundation Korea(NRF) grant funded by the Ministry of Education(MOE)., the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No. 2020R1A2B5B01001531)., and the Technology Innovation Program (20010263, Development of innovative design for UX environment improvement and commercialization model of wheelchair electric motorization kit with enhanced portability and convenience) funded By the Ministry of Trade, Industry & Energy(MOTIE, Korea).

## References

[1] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: Review and future perspectives," *IEEE Intelligent transportation systems magazine*, vol. 6, no. 4, pp. 6–22, 2014.

[2] J. Zhou and H. Peng, "Range policy of adaptive cruise control vehicles for improved flow stability and string stability," *IEEE Transactions on intelligent transportation systems*, vol. 6, no. 2, pp. 229–237, 2005.

[3] N. A. Stanton and M. S. Young, "Driver behaviour with adaptive cruise control," *Ergonomics*, vol. 48, no. 10, pp. 1294–1313, 2005.

[4] C. Hu, Z. Wang, Y. Qin, Y. Huang, J. Wang, and R. Wang, "Lane keeping control of autonomous vehicles with prescribed performance considering the rollover prevention and input saturation," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 7, pp. 3091–3103, 2019.

[5] A. Sharmin and R. Wan, "An autonomous lane-keeping ground vehicle control system for highway drive," in *9th International Conference on Robotic, Vision, Signal Processing and Power Applications*. Springer, 2017, pp. 351–361.

[6] M. Hasenjäger and H. Wersing, "Personalization in advanced driver assistance systems and autonomous vehicles: A review," in *2017 ieee 20th international conference on intelligent transportation systems (itsc)*. IEEE, 2017, pp. 1–7.

[7] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2019.

[8] J. Cao, C. Song, S. Peng, S. Song, X. Zhang, and F. Xiao, "Trajectory tracking control algorithm for autonomous vehicle considering cornering characteristics," *IEEE Access*, vol. 8, pp. 59 470–59 484, 2020.

[9] L. Tang, F. Yan, B. Zou, K. Wang, and C. Lv, "An improved kinematic model predictive control for high-speed path tracking of autonomous vehicles," *IEEE Access*, vol. 8, pp. 51 400–51 413, 2020.

[10] S. Xu, H. Peng, and Y. Tang, "Preview path tracking control with delay compensation for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[11] C. Hu, Y. Chen, and J. Wang, "Fuzzy observer-based transitional path-tracking control for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[12] N. H. Amer, H. Zamzuri, K. Hudha, and Z. A. Kadir, "Modelling and control strategies in path tracking control for autonomous ground vehicles: a review of state of the art and challenges," *Journal of intelligent & robotic systems*, vol. 86, no. 2, pp. 225–254, 2017.

[13] R. Solea and U. Nunes, "Trajectory planning and sliding-mode control based trajectory-tracking for cybercars," *Integrated Computer-Aided Engineering*, vol. 14, no. 1, pp. 33–47, 2007.

[14] K. Lee, S. Jeon, H. Kim, and D. Kum, "Optimal path tracking control of autonomous vehicle: Adaptive full-state linear quadratic gaussian (lqg) control," *IEEE Access*, vol. 7, pp. 109 120–109 133, 2019.

[15] P. Petrov and F. Nashashibi, "Modeling and nonlinear adaptive control for autonomous vehicle overtaking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1643–1656, 2014.

[16] C. Sun, X. Zhang, L. Xi, and Y. Tian, "Design of a path-tracking steering controller for autonomous vehicles," *Energies*, vol. 11, no. 6, p. 1451, 2018.

[17] Q. Yao, Y. Tian, Q. Wang, and S. Wang, "Control strategies on path tracking for autonomous vehicle: State of the art and future challenges," *IEEE Access*, vol. 8, pp. 161 211–161 222, 2020.

[18] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 667–682, 1999.

[19] L. Wang, *Model predictive control system design and implementation using MATLAB®*. Springer Science & Business Media, 2009.

[20] Y. Xu, B. Chen, X. Shan, W. Jia, Z. Lu, and G. Xu, "Model predictive control for lane keeping system in autonomous vehicle," in *2017 7th International Conference on Power Electronics Systems and Applications-Smart Mobility, Power Transfer & Security (PESA)*. IEEE, 2017, pp. 1–5.

[21] C. Huang, F. Naghdy, and H. Du, "Model predictive control-based lane change control system for an autonomous vehicle," in *2016 IEEE Region 10 Conference (TENCON)*. IEEE, 2016, pp. 3349–3354.

[22] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 952–964, 2016.

[23] X. Chen, B. Zhou, and X. Wu, "Autonomous vehicle path tracking control considering the stability under lane change," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, p. 0959651821991357, 2021.

[24] M. Muehlebach and R. D'Andrea, "A method for reducing the complexity of model predictive control in robotics applications," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2516–2523, 2019.

[25] B. Zhang, C. Zong, G. Chen, and B. Zhang, "Electrical vehicle path tracking based model predictive control with a laguerre function and exponential weight," *IEEE Access*, vol. 7, pp. 17 082–17 097, 2019.

[26] C. Dogruer, "Constrained model predictive control of a vehicle suspension using laguerre polynomials," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 234, no. 6, pp. 1253–1268, 2020.

[27] M. Hemmasian Ettefagh, M. Naraghi, J. De Dona, and F. Towhidkhah, "Orthonormal function parametrisation of model-predictive control for linear time-varying systems," *International Journal of Systems Science*, vol. 49, no. 4, pp. 868–883, 2018.

[28] M. H. Ettefagh, M. Naraghi, F. Towhidkhah, and J. De Dona, "Model predictive control of linear time varying systems using laguerre functions," in *2016 Australian Control Conference (AuCC)*. IEEE, 2016, pp. 120–125.

[29] M. Muehlebach and R. D'Andrea, "Basis functions design for the approximation of constrained linear quadratic regulator problems encountered in model predictive control," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 6189–6196.

[30] E. Alcalá, V. Puig, J. Quevedo, and U. Rosolia, "Autonomous racing using linear parameter varying-model predictive control (lpv-mpc)," *Control Engineering Practice*, vol. 95, p. 104270, 2020.

[31] F. Yakub and Y. Mori, "Comparative study of autonomous path-following vehicle control via model predictive control and linear quadratic control," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of automobile engineering*, vol. 229, no. 12, pp. 1695–1714, 2015.

[32] X. Hu, L. Chen, B. Tang, D. Cao, and H. He, "Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles," *Mechanical Systems and Signal Processing*, vol. 100, pp. 482–500, 2018.

[33] J. E. M. Sanchez, "Adaptive model predictive control with generalized orthonormal basis functions," Ph.D. dissertation, Carnegie Mellon University, 2017.

[34] N. Tanguy, "La transformation de laguerre discrète," Ph.D. dissertation, Université de Bretagne occidentale-Brest, 1994.

[35] K. Chu, M. Lee, and M. Sunwoo, "Local path planning for off-road autonomous driving with avoidance of static obstacles," *IEEE transactions on intelligent transportation systems*, vol. 13, no. 4, pp. 1599–1616, 2012.

[36] M. N. Zeilinger, C. N. Jones, and M. Morari, "Real-time suboptimal model predictive control using a combination of explicit mpc and online optimization," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1524–1534, 2011.

[37] N. Galioto and A. A. Gorodetsky, "Bayesian system id: Optimal management of parameter, model, and measurement uncertainty," *Nonlinear Dynamics*, vol. 102, no. 1, pp. 241–267, 2020.

[38] F. Pieri, C. Zambelli, A. Nannini, P. Olivo, and S. Saponara, "Is consumer electronics redesigning our cars?: Challenges of integrated technologies for sensing, computing, and storage," *IEEE Consumer Electronics Magazine*, vol. 7, no. 5, pp. 8–17, 2018.


**Dasol Jeong** received the B.S. degree in mechanical engineering from Korea Advanced Institute of Science and Technology (KAIST), Deajeon, Korea, and M.S. in Mechanical Engineering from the Korea Advanced Institute of Science and Technology (KAIST), in 2017 and 2019, respectively. Since 2019, he is currently pursuing the Ph.D. degree in Mechanical engineering at KAIST. His research interests include vehicle dynamics and control, intelligent tires, path planning, and control theory.

**Seibum Choi** (M'09) received the B.S. in mechanical engineering from Seoul National University, Seoul, Korea, the M.S. in mechanical engineering from KAIST, Deajeon, Korea, and the Ph.D. in control from the University of California, Berkeley, CA, USA, in 1993. From 1993 to 1997, he was involved in the development of automated vehicle control systems at the Institute of Transportation Studies, University of California. Through 2006, he was with TRW, Livonia, MI, USA, where he was involved in the development of advanced vehicle control systems. Since 2006, he has been faculty in the Mechanical Engineering Department, KAIST, Korea. His current research interests include fuel-saving technology, vehicle dynamics and control, and active safety systems. Prof. Choi is a Member of the American Society of Mechanical Engineers, the Society of Automotive Engineers, and the Korean Society of Automotive Engineers.

## APPENDIX

### A. Proof of Proposition 1

Using Theorem IV.1, the following state is satisfied:

$$\frac{\partial}{\partial a}\mathbf{L}_{a,N}(i) = \frac{1}{1-a^2}\begin{bmatrix} l_{a,2}(i) \\ -l_{a,1}(i) + 2l_{a,3}(i) \\ \vdots \\ -(N-1)l_{a,N-1}(i) + Nl_{a,N+1}(i) \end{bmatrix} = \frac{1}{1-a^2}\begin{bmatrix} 0 & 1 & \cdots & 0 & 0 & 0 \\ -1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & N-1 & 0 \\ 0 & 0 & \cdots & -(N-1) & 0 & N \end{bmatrix}\begin{bmatrix} l_{a,1}(i) \\ l_{a,2}(i) \\ \vdots \\ l_{a,N}(i) \\ l_{a,N+1}(i) \end{bmatrix} = \mathbf{T}_a\mathbf{L}_{a,N+1}(i) \tag{36}$$

Using (16), (19), and (36) partial derivative $\phi_{a,N}(k,m)^T$ with respect to $a$ as follows:

$$\begin{aligned} &\frac{\partial}{\partial a}\phi_{a,N}(k,m)^T \\ &= \left\{\sum_{i=0}^{m-1}\left[\sum_{p=i+1}^{m}\left(\prod_{j=p}^{m-1}\mathbf{A}_{k+j}\right)\right]\mathbf{B}\frac{\partial}{\partial a}\mathbf{L}_{a,N}(i)^T\right\} \\ &= \left\{\sum_{i=0}^{m-1}\left[\sum_{p=i+1}^{m}\left(\prod_{j=p}^{m-1}\mathbf{A}_{k+j}\right)\right]\mathbf{B}\mathbf{L}_{a,N+1}(i)^T\right\}\mathbf{T}_a^T \\ &= \phi_{a,N+1}(k,m)^T\mathbf{T}_a^T \end{aligned} \tag{37}$$

### B. Proof of Proposition 2

Minimum cost function partial derivative for $a$ is expressed as follows:

$$\begin{aligned} \frac{\partial J_{min,a,N}}{\partial a} &= \eta_{a,N}^T\frac{\partial \mathbf{\Omega}_{a,N}}{\partial a}\eta_{a,N} - 2\eta_{a,N}^T\frac{\partial \mathbf{\Phi}_{a,N}}{\partial a} \\ &+ \lambda_{a,N,act}^T\frac{\partial \mathbf{H}_{a,N,act}}{\partial a}\lambda_{a,N,act} - 2\lambda_{a,N,act}^T\frac{\partial \mathbf{K}_{a,N,act}}{\partial a} \end{aligned} \tag{38}$$

For matrices of the active set, the partial derivative for $a$ is expressed as follows:

$$\begin{aligned} \frac{\partial \mathbf{H}_{a,N,act}}{\partial a} &= -2\mathbf{M}_{a,N+1,act}\mathbf{T}_a^T\zeta_{a,N,act} \\ &\quad - 2\zeta_{a,N,act}^T\mathbf{T}_a\mathbf{\Omega}_{a,N+1}\mathbf{E}^T\zeta_{a,N,act} \\ \frac{\partial \mathbf{K}_{a,N,act}}{\partial a} &= -\mathbf{M}_{a,N+1,act}\mathbf{T}_a^T\eta_{a,N} - \zeta_{a,N,act}^T\mathbf{T_a}\mathbf{\Phi}_{a,N+1} \\ &\quad - 2\zeta_{a,N,act}^T\mathbf{T}_a\mathbf{\Omega}_{a,N+1}\mathbf{E}^T\eta_{a,N} \\ &\text{where } \zeta_{a,N,act} = -\mathbf{\Omega}_{a,N}^{-1}\mathbf{M}_{a,N,act} \end{aligned} \tag{39}$$

As a result of substituting equation (39) into equation (38), the following equation is derived.

$$\begin{aligned} \frac{\partial J_{min,a,N}}{\partial a} &= \hat{\eta}_{a,N}^T\left(\mathbf{T}_a\mathbf{\Omega}_{a,N+1}\mathbf{E}^T + \mathbf{E}\mathbf{\Omega}_{a,N+1}\mathbf{T}_a^T\right)\hat{\eta}_{a,N} \\ &\quad +2\hat{\eta}_{a,N}^T\left(\mathbf{T}_a\mathbf{\Phi}_{a,N+1} + \mathbf{T}_a\mathbf{M}_{a,N+1,act}^T\lambda_{a,N,act}\right) \end{aligned} \tag{40}$$