Explicit model predictive control (eMPC) has been proposed to reduce the huge computational complexity of MPC while maintaining the performance of MPC. Therefore, this control method has been more widely employed in the automotive industry than MPC. In this paper, an eMPC is designed to perform a double-lane-change (DLC) maneuver. This task has been employed to demonstrate the efficacy of controllers in an autonomous driving situation. In this sense, the proposed controller shows better performance than a driver model designed in CarSim at a high vehicle longitudinal velocity. The main contribution of this paper is to present an eMPC for discrete-time linear time-variant (LTV) systems so that the proposed controller can be robust against parameter variation. In a state-space representation of the vehicle, the longitudinal velocity of the vehicle is assumed to be a constant so that the whole system is linear time-invariant (LTI). However, it is inevitable that this velocity varies in an actual driving situation. Therefore, an eMPC controller is designed using an add-on unit to consider the varying parameter without modification of the eMPC solution. The CarSim simulation results of eMPC show enhanced performance compared to that of eMPC for the LTI system.

1 Introduction

Model predictive control (MPC) has the ability to anticipate future events of systems and fulfill complex constraints by solving open-loop optimization problems at each sampling time [1, 2]. Owing to these advantages, MPC has been employed in various fields. In [3], a MPC scheme has been applied for improving the fuel efficiency in vehicles in a given driving situation, resulting in a significant improvement in fuel consumption. An MPC approach has also been used for charging control of electric vehicles in a smart grid [4], demonstrating that the MPC approach can improve the stability and efficiency of smart grids. In [5], nonlinear MPC has been employed for regenerative braking system of hybrid electric commercial vehicles. The hardware-in-loop test has been conducted to verify improved recovery energy. MPC has been designed in [6] for prevention of a rollover or sideslip due to variable time delays in actual driving situations. As an upper controller, the MPC controller successfully calculated the...
desired tire force, while fulfilling the constraints of the vehicle. Furthermore, MPC has been employed in building cooling systems in [7], where it showed the ability to successfully fulfill such constraints, and in electric circuits to reduce the capacitor voltage ripple and enhance capacitor lifetime of capacitor in [8].

However, because as MPC is based on online optimization, a significant computational complexity is inevitably caused, and this drawback limits the target application of MPC to small and/or slow problems [9]. For this purpose, explicit model predictive control (eMPC) has been suggested to reduce the computational complexity by processing the MPC scheme offline [10]. This offline optimization can be made available with a multi-parametric programming skill, which aims to obtain the solution regions, the so-called critical regions, of an optimization problem by considering state variables as a vector of parameters [11]. Developments of multi-parametric programming can be found in [11, 12].

As eMPC can expand the target applications of MPC to relatively faster problems than that of online MPC, this method has been proposed in many real-time systems. Employed for adaptive cruise control, eMPC showed the suitability of the vehicle to behave in such working conditions in [13]. In [14], eMPC has been developed for real-time control of fully electric vehicles and showed improved energy efficiency by predicting a complex movement of a preceding vehicle. Moreover, eMPC has been used in electric circuits. In [15] an eMPC scheme has been suggested for DC-DC power supplies and the possibility of applying this scheme to industrial microcontrollers by taking advantage of the reduced computational complexity. In addition, an eMPC scheme has been used for active filtering and proven to be suitable for such systems, which require a large bandwidth by providing a fast response [16].

This paper proposes an eMPC controller for a double-lane-change (DLC) maneuver to demonstrate the ability of the proposed controller in an autonomous driving situation. Fig 1 represents a built-in road path for a DLC maneuver in CarSim with a 30 m longitudinal range given for the lane change. The completion of the DLC maneuver can be determined by checking whether a collision with the traffic cones occurs while the vehicle is driving. DLC maneuvers have been widely used to verify the ability of vehicle controllers for path-following control. In [17], a sliding mode controller has been designed to achieve a lane change maneuver in an automated highway system. Also, modeling of a human driver by fuzzy logic has

![Fig 1. Reference road path for DLC maneuver.](https://doi.org/10.1371/journal.pone.0208071.g001)
been done in [18] and verified through a DLC maneuver. For the simulation of DLC maneuvers, CarSim is used in this paper. This tool has been utilized to simulate driving tasks in order to analyze vehicle dynamics and test such vehicle controllers. Many cases using CarSim have been reported in the literature [19–22].

MPC-based controllers are mostly designed based on a discrete-time state space representation of linear time-invariant (LTI) systems. A preliminary version of the present paper, which has been published in proceedings at the international symposium on Industrial Electronics in 2018, also shows the design of an eMPC controller based on an LTI system. However, the parameters assumed to be constant in the LTI system change in an actual driving condition, degrading the performance of the vehicle controllers. To address this issue, we design an eMPC controller that can consider parameter changes through a few computational steps. The main concept of this approach, which has been previously reported in [23], is to compensate for errors caused in state variables due to parameter variation without modification of the critical regions of eMPC, that is, this method does not deteriorate the offline optimization framework.

For not only the steering control, but also other systems in vehicles such as transmission systems or power management, various optimization methods for those systems have been proposed [24–26]. Therefore, the number of electric control units (ECUs) in electric vehicles has increased significantly [27]; correspondingly, the reduction of computational complexity for ECUs has been an important issue to handle. Thanks to the major advantage of eMPC, this control method can serve as a novel approach for ECUs to be capable of performing many types of vehicle control in real time. Furthermore, the eMPC controller for the LTV system in this paper has the ability to maintain the performance of MPC in terms of parameter variation caused by working conditions, aging, or degradation of systems. As a result, it can be shown that the proposed controller can perform MPC optimization offline while ensuring the stability and robustness of the controller by applying a simple add-on unit.

The remainder of this paper is organized as follows. Section 2 reviews the MPC scheme and summarizes a formulation of eMPC. Section 3 shows a state-space representation of the vehicle and the design of eMPC controller for a linear time-variant (LTV) system. The simulation results are discussed in Section 4, and the conclusion and future work are presented in Section 5 and 6.

2 Explicit model predictive control

Based on the receding policy, MPC solves an open-loop optimization problem (1) at each sampling time.

\[
\min_{U \in U_{t-k}, k=0, \ldots, N_u} \{ J(U, x_t) = x_{t+N_f}^T P x_{t+N_f} + \sum_{k=0}^{N_y-1} \left[ (x_{t+k}^T Q x_{t+k} + u_{t+k}^T R u_{t+k} + (y_{t+k} - y_{ref,t+k})^T Q (y_{t+k} - y_{ref,t+k})) \right],
\]

s.t. \( u_{min} \leq u_{t+k} \leq u_{max}, \ k = 0, 1, \ldots, N_u \),
\( x_{min} \leq x_{t+k} \leq x_{max}, \ k = 0, 1, \ldots, N_y \),
\( x_{t+k+1} = A_d x_{t+k} + B_d u_{t+k}, \ k \geq 0 \),
\( y_{t+k} = C_d x_{t+k} \),
\( u_{t+k} = K x_{t+k}, \ N_u \leq k \leq N_y \)
where $A_d$, $B_d$, and $C_d$ are, respectively, the system, input, and output matrix in discrete-time. $k$ is the time index, and $x_{t,k|k}$ indicates the predicted state vector at time $t+k$ from the current time $t$. $N_u$ and $N_y$ denote the prediction horizon for the state vector $x_t$ and input horizon for the input vector $u(t)$. It is assumed that $Q = Q^T \succ 0$, $R = R^T \succ 0$, $QR = QR^T \succ 0$, and $P \succ 0$. The matrix $P$ for the predicted state vector at $N_y$ and the gain matrix $K$, the control law for $N_u < k \leq N_y$, are the solution of the discrete-time algebraic Riccati equation (ARE) as follows:

$$
P = A_d^T P A_d - A_d^T P B_d K + Q = 0,
$$
$$
K = (B_d^T P B_d + R)^{-1} B_d^T P A_d. 
$$

(2)

Based on these weighting matrices, the MPC controller obtains a sequence of optimal vectors $U$, which minimizes the cost function $J$. Then, the controller applies the first input vector $u_t$ to the system while discarding others. As $u_t$ is obtained considering the constraints over $N_u$ and $N_y$, MPC can anticipate future events of the system and fulfill such complex constraints.

Despite these advantages, a considerable computational load due to online optimization of MPC has been regarded as a huge drawback. Therefore, a new MPC scheme, which can reduce the computational burden, has been suggested in [10]. Called explicit MPC, this approach uses a multi-parametric quadratic programming (mp-QP) skill to perform MPC offline.

By using the prediction equation of the state vector as follows

$$
x(t) = \Phi x(t) + \Lambda u(t),
$$

where

$$
x(t) = \begin{bmatrix}
    x_{k+1|k} \\
    \vdots \\
    x_{k+N_y|k}
\end{bmatrix},
\quad
u_t = \begin{bmatrix}
    u_t \\
    \vdots \\
    u(k+N_y-1|k)
\end{bmatrix},
\quad
\Phi = \begin{bmatrix}
    A_d \\
    \vdots \\
    A_d^{N_y}
\end{bmatrix},
\quad
\Lambda = \begin{bmatrix}
    B_d & 0 & \cdots & 0 \\
    A_d B_d & B_d & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    A_d^{N_y-1} B_d & A_d^{N_y-2} B_d & \cdots & B_d
\end{bmatrix}.
$$

(3)
the optimization problem (1) can be reformulated as follows:

\[ J(U, x_t) = U^T H U + 2x_t^T F U + x_t^T G x_t, \]

s.t. \( \Phi_i U \leq B_i + A_i x_t, \)

where

\[ H = \Phi^T \hat{Q} \Phi + \hat{R}, \quad F = \Phi^T \hat{Q} \Lambda, \quad G = \Lambda^T \hat{Q} \Lambda + Q, \]

\[ \Phi = \begin{bmatrix} \Phi_1 \\ \vdots \\ \Phi_N \end{bmatrix}, \quad i = 1, \ldots, N, \quad B_i = \begin{bmatrix} x_{\text{max}} \\ -x_{\text{min}} \end{bmatrix}, \]

\[ A_i = \begin{bmatrix} -A_i' \\ A_i' \end{bmatrix}, \quad i = 1, \ldots, N, \]

\[ \hat{Q} = \begin{bmatrix} Q & \cdots \\ \cdots & Q \\ P & \cdots \end{bmatrix}, \quad \hat{R} = \begin{bmatrix} R & \cdots \\ \cdots & R \end{bmatrix}. \]

The matrices \( H, F, \) and \( G \) can be obtained offline, and \( \Phi_i \) indicates the \( i \)th row of \( \Phi \).

From (4), an mp-QP problem can be formulated by defining \( z \triangleq U + H^{-1} F x \) as follows:

\[ J_z(x_t) = \min \left\{ \frac{1}{2} z^T H z \right\} \]

s.t. \( Wz \leq G + S x_t, \]

where \( J_z(x_t) = J(U, x_t) - \frac{1}{2} x_t^T (Y - FH^{-1} F^T) x_t, \) \( S \triangleq C_i + GH^{-1} F^T, \) and \( H \) is a weighting matrix for \( z \). The inequality constraint in (5), which is defined by \( x_t \), is a polytope, which means the generated critical regions are also polytope. The optimization variable \( z \) is a continuous piecewise affine; correspondingly, the optimal input \( u_t \) is also a continuous piecewise affine function of \( x_t \) [10]. One approach to forming the critical regions has been suggested in [28] as

\[ CR_j = C_j A < C_j b, \quad j = 1, \ldots, q, \]

\[ CR_{j+1} = \{ C_j A > C_j b \text{ and } C_i A \leq C_j b, \forall i < j \}, \quad j = 1, \ldots, q, \]

where \( CR_j, C_j, \) and \( q \) indicate the \( j \)th critical region, inequality constraint, and the number of constraints in the mp-QP problem (5) respectively. The whole space of the parameter vector \( z \) is the sum of the critical regions. In this paper, the parametric optimization (POP) toolbox from the PAROC Platform is used [29]. On the basis of the values of \( z \), the eMPC controller explicitly chooses a critical region among the critical regions. A unique sequence of MPC laws is contained in each critical region so that the controller can obtain a sequence of the optimal vector \( U \) in (1). Then, as MPC processes, only the first input vector \( u_t \) is chosen as the actual control input to the system. Note that in the eMPC scheme, the task of solving the complicated optimization problem (1) at each sampling time turns into a simple computational step of choosing a critical region.
3 Controller design

3.1 Vehicle dynamics

In this subsection, the vehicle dynamics and a state-space representation reflecting a varying parameter will be addressed. Fig 2 represents a vehicle model with longitudinal displacement $x_{igt}(t)$ and lateral displacement $y_{ltr}(t)$. Table 1 lists the vehicle parameters, where its values refer to a vehicle model in CarSim, C-Class Hatchback 2017. $x_{igt}(t)$ and $y_{ltr}(t)$ can be obtained as follows:

\[
\begin{align*}
\dot{x}_{igt}(t) &= V_x \cos \delta_{yaw}(t) - \dot{y}_{ltr}(t) \sin \delta_{yaw}(t) \\
\dot{y}_{ltr}(t) &= V_x \sin \delta_{yaw}(t) + \dot{y}_{ltr}(t) \cos \delta_{yaw}(t),
\end{align*}
\]

(7)

where $\delta_{yaw}(t)$ is the yaw angle of the vehicle. Considering that $\dot{y}_{ltr}(t)$ and $\delta_{yaw}(t)$ in (7) is significantly smaller than $V_x$ in DLC maneuvers [30], (7) can be simply linearized as follows:

\[
\begin{align*}
\dot{x}_{igt}(t) &= V_x \\
\dot{y}_{ltr}(t) &= V_x \delta_{yaw}(t) + \dot{y}_{ltr}(t).
\end{align*}
\]

(8)

Table 1. Parameters of vehicle model.

| Symbol | Description               | Value (units) |
|--------|---------------------------|---------------|
| $V_x$  | longitudinal vehicle velocity | m/s           |
| $m$    | vehicle mass              | 1270 kg       |
| $I_{zz}$ | vehicle yaw inertia     | 1995.78 kg\cdot m^2 |
| $a$    | distance from center to front axis | 1.192 m |
| $b$    | distance from center to rear axis | 1.548 m |
| $C_f$  | front tire stiffness      | 55405 N/rad   |
| $C_r$  | rear tire stiffness       | 55405 N/rad   |
| $G$    | average steering ratio    | 15.29         |

https://doi.org/10.1371/journal.pone.0208071.t001
The motion equation of $y_{ltr}(t)$ and $\delta_{yaw}(t)$ of the vehicle can be presented as follows [30]

$$
my_{ltr}'(t) = -\left(\frac{C_f + C_r}{V_x}\right)y_{ltr}(t)
- \left\{\frac{aC_f + bC_r}{V_x} + mV_x\right\}\delta_{yaw}(t) + \frac{C_f}{G}\delta_{sw}(t),
$$

$$
I_{zz}\delta_{yaw}'(t) = -\left(\frac{aC_f - bC_r}{V_x}\right)y_{ltr}(t)
- \frac{a^2C_f + b^2C_r}{V_x}\delta_{yaw}(t) + \frac{aC_f}{G}\delta_{sw}(t).
$$

In an actual driving situation, it is nearly impossible for the driver to maintain a constant longitudinal velocity $V_x$. Therefore, in this paper, $V_x$ is considered a varying parameter, $V_x(t)$.

Using (9), the state-space representation of the LTV system can be obtained as follows:

$$
\dot{x}_t = A(t)x_t + Bu_t
y_t = Cx_t,
$$

where

$$
A(t) = \begin{bmatrix}
\frac{C_f}{mV_x(t)} & -\frac{aC_f + bC_r}{mV_x(t)} & -V_x(t) & 0 & 0 \\
\frac{aC_f - bC_r}{I_{zz}V_x(t)} & -\frac{a^2C_f + b^2C_r}{I_{zz}V_x(t)} & 0 & 0 \\
1 & 0 & 0 & V_x(t) \\
0 & 1 & 0 & 0
\end{bmatrix},
$$

$$
B = \begin{bmatrix}
\frac{C_f}{m} \\
\frac{aC_f}{I_{zz}G} \\
0 \\
0
\end{bmatrix},
C = [0 \ 0 \ 1 \ 0].
$$

In (10), $x_t = [\dot{y}_{ltr}(t) \ \delta_{yaw}(t) \ y_{ltr}(t) \ \dot{\delta}_{yaw}(t)]^T$ and $u(t) = \delta_{sw}(t)$. The continuous-time state-space representation (10) is discretized with a sampling time of 0.01 s as follows:

$$
x[k+1] = A_d[k]x[k] + B_du[k]
y[k] = C_dx[k],
$$
where

\[
A_d[k] = \begin{bmatrix}
0.9676 & A_1^1[k] & 0 & 0 \\
0.0109 & A_1^2[k] & 0 & 0 \\
0.0098 & 0.0001 & 1 & A_2^2[k] \\
A_1^3[k] & 0.0097 & 0 & 1 \\
\end{bmatrix},
B_d = \begin{bmatrix}
0.0250 \\
0.0234 \\
0.0001 \\
0.0001 \\
\end{bmatrix},
\]

\[C_d = [0 \ 0 \ 1 \ 0],\]

where \(A^i_d[k], i = 1, \ldots, 4\) indicates the varying elements due to \(V_x(t)\), and \(k\) is the sampling time index.

### 3.2 Explicit MPC for LTI system

First, the design of the eMPC controller with the LTI system, which treats the varying parameter \(V_x(t)\) as a constant value, will be presented.

The constraints of the states and input are determined as follows:

\[
\begin{bmatrix}
-0.65 \\
-25.21 \\
-1 \\
-14.32
\end{bmatrix} \leq \begin{bmatrix}
\dot{y}_w(t) \ [\text{km/h}] \\
\dot{\delta}_{yaw}(t) \ [\text{deg/s}] \\
y_w(t) \ [\text{m}] \\
\delta_{yaw}(t) \ [\text{deg}]
\end{bmatrix} \leq \begin{bmatrix}
0.65 \\
25.21 \\
4.5 \\
14.32
\end{bmatrix},
\]

\[-100 \leq \delta_{yaw}(t) \ [\text{deg}] \leq 100.\]

The constraints of \(\dot{y}_w(t)\) and \(\delta_{yaw}(t)\) are set to satisfy the assumption applied in (8). Other constraints are determined based on the boundaries of the driver model states in CarSim, with a longitudinal velocity of 60 km/h and the geometrical constraint of the road path.

The determination of the weighting matrices in (1) is set to

\[
Q = \begin{bmatrix}
0.04 & 0 & 0 & 0 \\
0 & 0.002 & 0 & 0 \\
0 & 0 & 0.001 & 0 \\
0 & 0 & 0 & 125
\end{bmatrix},
R = 1 \times 10^{-6}, \quad QR = 35.
\]

As the value of the weighting factor increases, the associated state or input is more dominantly considered in solving (1). The values in (15) are chosen with respect to this relation through trial and error. The matrix \(P\) in (1), which acts as the weighting matrix for the terminal states, is given by the discrete-time Riccati Eq (2).

The input horizon \(N_y\) is set to 1, and to obtain the range of the prediction horizon \(N_y\), some cases of \(N_y\) are set and simulated as shown in Fig 3. It can be found that the ability of eMPC controllers to predict future events, which is \(y_w(t)\) in this case, is enhanced with a long range of \(N_y\). However, in this simulation, an excessively long range of \(N_y\) results in an unreasonably earlier steering, which can be seen in the case of \(N_y = 60\) as well as increasing the number of critical regions which consumes a large memory size. Because this control action may cause the vehicle to collide with the traffic cones, \(N_y\) is set to 40 in this paper.

A total of 113 critical regions were generated using the POP toolbox and Fig 4 shows several of feasible critical regions where only two parameters, \(z_1\) and \(z_2\), vary in the parameter vector \(z\).

In this figure, \([z_1, \ldots, z_5] = [\dot{y}_w(t) \ \dot{\delta}_{yaw}(t) \ y_w(t) \ \delta_{yaw}(t) \ y_{wtr}^f(t)]\). The range where \(z_1\)
and $z_2$ are bounded by the constraint set (14). Each critical region owns its unique sequence of optimal vectors and a region is chosen based on the values of $z_1$ and $z_2$. The eMPC controller selects the region by simply calculating inequality constraints instead of solving the optimization problem (1) at each sampling time. This reduction in computational load relieves the burden of implementing industrial microcontrollers in real-time situations [31]. For an actual implementation of eMPC controllers, the POP toolbox forms critical regions in the controller design process as in this subsection. Then, eMPC controllers employ the conditional statement written by C language, which represents generated critical regions. Based on the values of the variables, controllers explore this statement to obtain an optimal input vector for the real-time closed-loop control.

In this subsection, an algorithm proposed [23] to compensate the state vector $x[k]$ will be presented. Through this algorithm, the eMPC controller can be made robust against parameter changes.

Fig 5A illustrates the implementation of eMPC for LTI systems. Based on the critical regions, which are formed by LTI systems, and the measured state vector $x[k]$, the eMPC controller explicitly obtains an optimal input $u[k]$. Because the eMPC controller is designed based on the critical regions, it can be made robust against parameter changes.
on $A_d$, the controller is vulnerable to such parameter variation. Whereas Fig 5B shows the implementation of eMPC, where an error compensator, which is a simple linear matrix, is added to compensate for the state vector according to the variation in the system matrix.

The matrix $A_d[k]$ can be determined as

$$A_d[k] = \hat{A}_d + \sum_{i=1}^{N_v} \Delta x_i[k], \quad (16)$$

where $\Delta x_i \in \mathbb{R}^{n \times n}$, $x_i(t) \in \mathbb{R}$, and $N_v$ is the number of varying parameters. We know which elements vary but how the matrix evolves is unknown.

Eq (12) can be rewritten using (16) as follows:

$$x[k] - \hat{A}_d x[k - 1] - B_d u[k - 1] = \sum_{i=1}^{N_v} \Delta x_i[k - 1] x[k - 1],$$

$$= D[k - 1][x_i[k - 1], x_i[k - 1]],$$

$$\ldots, x_i[k - 1]], \quad (17)$$
where $D[k-1] = [\Delta_1 x[k-1], \Delta_2 x[k-1], \ldots, \Delta_N x[k-1]]$ and $\Delta_i x[k-1] \in \mathbb{R}^n$, $i = 1, \ldots, N_v$; thus $D[k-1] \in \mathbb{R}^{n \times m}$.

In (17), the previous states, $x[k-1]$, and input, $u[k-1]$, are known by measurement. Based on the Rouche–Capelli theorem, (17) has a unique solution $\alpha_i[k-1], i = 1, \ldots, N_v$ for all $t \geq 1$.

As a result, by substituting (17) into (16), $A_d[k-1]$ can be obtained based on the recent history of the states and input, and the current state measurement. In order to obtain the gain matrix $L[k]$ for the compensation of the state vector, the following equation can be considered:

$$\hat{x}[k+1] = \hat{A}_d \hat{x}[k] + \hat{B}_d u[k],$$

where $\hat{x}[k]$ denotes a compensated state vector at the time instant $k$. Assume that the initial system matrix $\hat{A}_d$ is invertible, and $\hat{x}[k]$ in (18) is replaced by $L[k] x[k]$, where

$$L[k] = \hat{A}_d^{-1} \hat{A}_d[k].$$

Then, considering (12) and (18), $\hat{x}[k+1] = x[k+1]$, which means the compensated state vector can be identical with the state vector obtained by the time varying system matrix $A_d[k]$ in (12) by employing the gain matrix in (19).

By substituting $A_d[k-1]$ from (17) for $A_d[k]$ in (19), $L[k-1]$ can be calculated. Then, with sufficiently slow time-varying parameters, it can be assumed that $L[k] \equiv L[k-1]$; consequently, the state vector can be compensated as follows:

$$\hat{x}[k] = L[k] x[k] \equiv L[k-1] x[k].$$

Because (20), the compensation of the state vector in Fig 5B, is a simple matrix multiplication based on the previous states and input, this compensator does not need modification of the critical regions of eMPC regarding parameter variation. For the better understanding, a full explanation about this approach can be found in [23].

### 4 Simulation results

Fig 6 shows the simulation results of the eMPC controller for an LTV system at 60 km/h longitudinal velocity, where the driver model is a built-in model in CarSim. The plot of the lateral displacement shows that both eMPC controllers and the driver model succeed in the DLC.

![Fig 5. Frameworks of eMPC for LTI systems (A) and LTV systems (B). In terms of parameter variation, there is no alternative way in Fig 5A to adjust the variation because the critical regions cannot be changed with respect to variation. In contrast, in Fig 5B, compensating for the state vector with an error compensator, enables the controller to be robust against such parameter variation. The main advantage of this approach is that, by simply adding a compensator, where only matrix multiplication is taken in its process, no modification of the critical regions is necessary.](https://doi.org/10.1371/journal.pone.0208071.g005)
Fig 6. Simulation result of eMPC controller for the LTI system at a longitudinal velocity of 60 km/h. This figure shows the fulfillment of the constraint set (14) of the eMPC controller. The vertical range of each window is identical with the constraints of each variable and input.

https://doi.org/10.1371/journal.pone.0208071.g006
maneuver. This figure also demonstrates the fulfillment of the constraints of the eMPC controller by showing that all variables are within the boundaries.

Fig 7 shows the simulation results of eMPC controllers for both LTI and LTV systems at a longitudinal velocity of 100 km/h. The tracking abilities of all controllers, including the driver model, are degraded as a relatively more rapid and sophisticated control of the steering wheel is required than that at a longitudinal velocity of 60 km/h. As we predicted previously, Fig 7 also shows that the longitudinal velocity varies during the driving situation even though this value is assumed to be constant. The eMPC controller designed for the LTI system is inevitable to be vulnerable to this parameter variation. This degradation of the performance of the controller demonstrates the importance of designing the eMPC controller for LTV systems. The result of the lateral displacement error shows the improvement in the eMPC controller for the LTV system. After five seconds of the simulation, where the longitudinal velocity varies apparently enough, the error in the eMPC controller for the LTV system decreases compared to that for the LTI system. A considerable lateral error appears in the simulation of the driver model which nearly causes a collision with a traffic cone, as shown in Fig 8. The driver model and eMPC controller for the LTV system are colored by gray and red, respectively.
Fig 9 shows the state vector of the eMPC controllers for both systems, where the controller for the LTV system employs the compensated state vector. At each sampling time, the gain matrix for compensation $L[k]$ is updated by (19) based on the recent measurement of states and inputs to consider parameter variation. As seen in Fig 9, the variation in the states of the LTV system is relatively restricted than that of the LTI system, resulting in an increase of the tracking ability. In addition, because a rapid lateral acceleration corresponds to a large sprung mass acceleration, which deteriorates the ride comfort [32], the proposed controller can also enhance the ride comfort.

Furthermore, in the actual driving situation of commercial vehicles, it frequently happens that the initial parameters change, for instance, vehicle mass due to passengers on buses or loads on trucks. Hence, robust controllers against parameter variation need to be designed to maintain the performance of controllers for LTI systems.

5 Conclusion

An eMPC controller was designed for the LTV system of the vehicle. eMPC has been proposed to reduce the huge computational complexity of MPC by using an mp-QP technique. In the preliminary version of this paper, the vehicle model for the DLC maneuver is assumed to be LTI. However, such parameters, the longitudinal velocity for instance, are inevitably changed in an actual driving situation.

In this paper, we designed the eMPC controller with a simple add-on unit for the LTV system. The main contribution of this approach is that no modification is required in the critical regions of eMPC. Through the simulation results, the fulfillment of the constraints and tracking ability of the proposed controller were shown. Furthermore, it was found that the proposed controller improved ride comfort by compensating for the state vector and control input.

6 Future works

In the future work, we intend to conduct an actual experiment for a DLC maneuver using an electric power steering test bench to verify the applicability of the eMPC controller for autonomous driving. In addition, a more detailed analysis about the feasibility of the eMPC scheme is
planned for the future work. In this paper, the eMPC controller succeeded in obtaining the control law with respect to the constraints of the vehicle and discrete-time ARE. Therefore, we intend to simulate how the eMPC controller derives the control input when the controller fails to solve the MPC problem due to more complex or harder constraints.

Moreover, because of the increased number of ECUs in electric vehicles, the time delay appeared in transmitting control or state signals has been significantly considered to cope with. Regarding this issue, in [33], a robust sliding mode controller has been designed for the

---

Fig 9. States of the vehicle model during DLC maneuver. In the implementation of the eMPC controller for the LTV system, the states are compensated regarding the parameter variation to improve the robustness of the controller. The proposed controller not only improves the tracking ability of the controller, but also enhances ride comfort as the lateral velocity is more restricted compared to the eMPC controller for the LTI system.

https://doi.org/10.1371/journal.pone.0208071.g009
systems with delay. Therefore, we also intend to employ the algorithm presented in this paper to compensate the time delay in the state and input signals.

**Supporting information**

S1 Dataset. Constraints for partitioning of critical regions when the prediction horizon is 40 and input horizon is 1.

(XLSX)

**Author Contributions**

Conceptualization: Junho Lee, Hyuk-Jun Chang.

Investigation: Junho Lee.

Supervision: Hyuk-Jun Chang.

Writing – original draft: Junho Lee.

Writing – review & editing: Hyuk-Jun Chang.

**References**

1. Lee JH, Morari M, Garcia CE. State-space interpretation of model predictive control. Automatica, vol. 30, no. 4, pp. 707–717, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0005109894901597.

2. Borrelli F. Constrained optimal control of linear and hybrid systems. Springer, 2003, vol. 290.

3. Kamal MAS, Mukai M, Murata J, Kawabe T. Model predictive control of vehicles on urban roads for improved fuel economy. IEEE Transactions on control systems technology, vol. 21, no. 3, pp. 831–841, 2013. https://doi.org/10.1109/TCST.2012.2196478

4. Giorgio AD, Liberati F, Canale S. Electric vehicles charging control in a smart grid: A model predictive control approach. Control Engineering Practice, vol. 22, pp. 147–162, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066113001871.

5. Li L, Zhang Y, Yang C, Yan B, Martinez CM. Model predictive control-based efficient energy recovery control strategy for regenerative braking system of hybrid electric bus. Energy Conversion and Management, vol. 111, pp. 299–314, 2016. https://doi.org/10.1016/j.enconman.2015.12.077

6. Li L, Lu T, Wang R, Chen J. A Three-Dimensional Dynamics Control Framework of Vehicle Lateral Stability and Rollover Prevention via Active Braking With MPC. IEEE Transactions on Industrial Electronics, vol. 64, pp. 3389–3401, 2017.

7. Ma Y, Borrelli F, Hencsey B, Coffey B, Bengbea S, Haves P. Model Predictive Control for the Operation of Building Cooling Systems. IEEE Transactions on Control Systems Technology, vol. 20, no. 3, pp. 796–803, May 2012. https://doi.org/10.1109/TCST.2011.2124461

8. Townsden CD, Summers TJ, Vodden G, Watson AJ, Betz RE, Clare JC. Optimization of Switching Losses and Capacitor Voltage Ripple Using Model Predictive Control of a Cascaded H-Bridge Multilevel StatCom. IEEE Transactions on Power Electronics, vol. 28, no. 7, pp. 3077–3087, July 2013. https://doi.org/10.1109/TPEL.2012.2219593

9. Bengtsson J, Strandh P, Johansson R, Tuneest’al P, Johansson B. Multi-output control of a heavy duty HCCI engine using variable valve actuation and model predictive control. SAE technical paper, Tech. Rep., 2006.

10. Bemporad A, Morari M, Dua V, Pistikopoulos EN. The explicit linear quadratic regulator for constrained systems. Automatica, vol. 38, no. 1, pp. 3–20, 2002. https://doi.org/10.1016/S0005-1098(01)00174-1

11. Pistikopoulos E. Perspectives in multiparametric programming and explicit model predictive control. AIChE journal, vol. 55, no. 8, pp. 1918–1925, 2009. https://doi.org/10.1002/aic.11965

12. Pistikopoulos EN. From multi-parametric programming theory to MPC-on-a-chip multi-scale systems applications. Computers & Chemical Engineering, vol. 47, pp. 57–66, 2012. https://doi.org/10.1016/j.compchemeng.2012.06.031

13. Naus G, van den Bleek R, Ploeg J, Scheepers B, van de Molengraft R, Steinbuch M. Explicit MPC design and performance evaluation of an ACC stop-&-go in 2008 American Control Conference, June 2008, pp. 224–229.
14. Zhang S, Luo Y, Li K, Li V. Real-Time Energy-Efficient Control for Fully Electric Vehicles Based on an Explicit Model Predictive Control Method. IEEE Transactions on Vehicular Technology, vol. 67, no. 6, pp. 4693–4701, June 2018. [https://doi.org/10.1109/TVT.2018.2806400]

15. Beccuti AG, Mariethoz S, Cliquennois S, Wang S, Morari M. Explicit Model Predictive Control of DC-DC Switched-Mode Power Supplies With Extended Kalman Filtering. IEEE Transactions on Industrial Electronics, vol. 56, no. 6, pp. 1864–1874, June 2009. [https://doi.org/10.1109/TIE.2009.2015748]

16. Mariethoz S, Morari M. Explicit Model-Predictive control of a PWM Inverter With an LCL filter. IEEE Transactions on Industrial Electronics, vol. 56, no. 2, pp. 389–399, February 2009. [https://doi.org/10.1109/TIE.2008.2008793]

17. Hatipoglu C, Ozguner U, Redmill KA. Automated lane change controller design. IEEE Transactions on Intelligent Transportation Systems, vol. 4, no. 1, pp. 13–22, March 2003. [https://doi.org/10.1109/TITS.2003.811644]

18. Hajjaji AE, Ouladsine M. Modeling Human Vehicle Driving by fuzzy logic for standardized ISO double lane change maneuver. In Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication. ROMAN 2001 (Cat. No.01THB591), pp. 499–503, 2001.

19. Yao L, Wu J, Wang Y, Liu C. Research on vehicle integrated control algorithm based on matlab and canoe co-simulation. In Transportation Electrification Asia-Pacific (ITEC Asia-Pacific), 2014 IEEE Conference and Expo. IEEE, pp. 1–5, 2014.

20. Kim S, Park K, Song HJ, Hwang YK, Moon SJ, Ahn HS, et al. Development of control logic for hydraulic active roll control system. International Journal of Automotive Technology, vol. 13, no. 1, p. 87, October 2011. [Online]. Available: [https://doi.org/10.1007/s12239-012-0008-5]

21. Yim S. Fault-tolerant yaw moment control with steer—and brake-by-wire devices. International Journal of Automotive Technology, vol. 15, no. 3, pp. 463–468, April 2014. [Online]. Available: [https://doi.org/10.1007/s12239-014-0048-0]

22. Amdouni I, Jeddi N, Amraoui LE. Optimal control approach developed to Four-Wheel Active Steering Vehicles. In 2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO), pp. 1–6, April 2013.

23. Chang H, Krieger A, Astolfi A, Pistikopoulos E. Robust multiparametric model predictive control for LPV systems with application to anaesthesia. Journal of Process Control, vol. 24, no. 10, pp. 1538–1547, 2014. [https://doi.org/10.1016/j.jprocont.2014.07.005]

24. Guo H, Sun Q, Wang C, Wang Q, Lu S. A systematic design and optimization method of transmission system and power management for a plug-in hybrid electric vehicle. Energy, vol. 148, pp. 1006–1017, April 2018. [https://doi.org/10.1016/j.energy.2018.01.152]

25. Xiong R, Cao J, Yu Q. Reinforcement learning-based real-time power management for hybrid energy storage system in the plug-in hybrid electric vehicle. Applied Energy, vol. 211, pp. 538–548, February 2018. [https://doi.org/10.1016/j.apenergy.2017.11.072]

26. Uthaichana K, DeCarlo, R, Bengue, S, Žefran M, Pekarek S. Hybrid optimal theory and predictive control for power management in hybrid electric vehicle. arXiv preprint arXiv:1804.00757, 2018.

27. Xie G, Zeng G, Liu L, Li R, Li K. High performance real-time scheduling of multiple mixed-criticality functions in heterogeneous distributed embedded systems. Journal of Systems Architecture, vol. 70, pp. 3–14, 2016. [https://doi.org/10.1016/j.sysarc.2016.04.008]

28. Dua V, Pistikopoulos EN. An Algorithm for the Solution of Multiparametric Mixed Integer Linear Programming Problems. Annals of Operations Research, vol. 99, no. 1, pp. 123–139, 2000. [https://doi.org/10.1023/A:1019241006836]

29. Oberdieck R, Diangelakis NA, Burnak B, Katz J, Avraamidou S, Pistikopoulos EN. POP User Manual Version 2.0.

30. Rajamani R. Vehicle dynamics and control. Springer Science & Business Media, 2011.

31. Johansen TA, Grancharova A. Approximate explicit constrained linear model predictive control via orthogonal tree search. IEEE Transactions on Automatic Control, vol. 48, no. 5, pp. 810–815, May 2003. [https://doi.org/10.1109/TAC.2003.811259]

32. Nguyen LH, Hong KS, Park S. Road-frequency adaptive control for semi-active suspension systems. International Journal of Control, Automation and Systems, vol. 8, no. 5, pp. 1029–1038, October 2010. [Online]. Available: [https://doi.org/10.1007/s12239-010-0512-1]

33. Caballero-Barragán H, Osuna-Ibarra LP, Loukianov AG, Plestan F. Sliding mode predictive control of linear uncertain systems with delays. Automatica, vol. 94, no. 5, pp. 409–415, August 2018. [https://doi.org/10.1016/j.automatica.2018.04.040]