2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

Simultaneous Realization of Planning and Control for Lane-change Behavior Using Nonlinear Model Predictive Control

Nobuto Sugie[1], Hiroyuki Okuda[1], Tatsuya Suzuki[1], Kentaro Haraguchi[2]
and Zibo Kang[2]

Abstract— **The demand for smart control technology is glowing according to the increase of expectation for autonomous driving. This paper presents an integrated planning and control scheme for automated lane-change behavior based on real-time nonlinear model predictive control (NMPC). In the proposed method, switched weighting parameters are used in the cost function over the prediction horizon. This enables us to reduce the number of reference points for control compared with the conventional framework. In addition, some safety constraints, such as collision avoidance and/or speed limitation are considered in a consistent manner. Although the NMPC usually requires huge computational cost, by applying the Continuation/GMRES (C/GMRES) method, it can be drastically reduced and become possible to be implemented in real time. Finally, validity of the proposed method is demonstrated by simulation study.**

I. Introduction

The expectation for the autonomous driving is rapidly glowing. At the same time, development of fundamental framework for the design of dependable autonomous driving becomes more important. Path planning and vehicle control are basic and essential functions to realize the autonomous driving. There are huge number of previous researches of path planning and vehicle control [1]–[5]. Vehicle controller which realizes the planned path requires the vehicle dynamics model to achieve accurate path tracking. Path planner also have to consider some physical characteristics and constraints of the vehicle dynamics in order to plan the safe, comfortable and feasible path. The standard path planner, however, generates the reference path based on the smooth curve, such as the spline curve, clothoid curve and so on, to design the path that the vehicle can easily follow. In this framework, it is sometimes difficult to determine the feasible parameters of the curve because of lacking the information on the physical characteristics, such as nonholonomic dynamical feature of the vehicle. In order to improve the feasibility and reliability of the path planning the vehicle dynamics must be explicitly considered together with other safety constraints. This implies that vehicle dynamics must be considered in both path planning and control, and same design effort must be paid for the planning and control twice. To overcome this ‘double effort’ problem, ’simultaneous motion planning and control’(SMPC) [6] which tries to solve the path planning and vehicle control as the single problem in unified framework. In the SMPC, the control objective(goal) and constraints in order to achieve the desired task are considered without explicitly considering the driving path. This drastically reduces the total design cost for the autonomous vehicle. This is also beneficial for the developers of autonomous driving system since they can be implemented as a single unified module in the overall system architecture.

In this paper, a new simultaneous planning and control, which can find control inputs without executing an explicit path planning, is proposed and applied to lane-change task. Proposed method exploits the nonlinear model predictive control(NMPC) considering the vehicle dynamics and the constraints to reflect the physical limitation of the vehicle and the safety requirements. In the proposed NMPC, the parameters of cost function is switched depending on the predicted state variables in order to secure the safety margin to surrounding vehicles without specifying many reference points. In other words, the proposed method requires only the information on control objectives and constraints, and does not need to consider the path planning explicitly. Although the NMPC usually requires huge computational cost, by applying the Continuation/GMRES (C/GMRES) method [11], [13], it can be drastically reduced and the NMPC becomes possible to be executed in real time. In the C/GMRES method, instead of finding optimal control input at every control cycle, the difference from the last control input is optimized and implemented. By this mechanism, The required computational cost can be one hundredth of the case that the standard nonlinear optimization is implemented.

Finally, Proposed method is implemented on the driving simulator, and its usefulness is validated for various lane-changing situation.

II. Problem setting

A lane-changing task is addressed in this paper assuming that the all positions the driving speeds of the surrounding vehicles, and the tire angle of the ego vehicle are measurable. Straight express way with two lanes is used as the driving environment as shown in Fig.1. Width of each lanes is 3[m]. X axis is along with the longitudinal direction of the road, and Y axis, lateral position from the center of right lane. The ego vehicle drives on the right lane and starts from origin (p_x,p_y)=(0,0) where p_x and p_y are the positions of the ego vehicle. Another vehicle drives on the left lane as an surrounding vehicle to be avoided from collision. The Y position of surrounding vehicle O_y is 3[m] during the experiment. There are supposed to be only two vehicles

*This work was not supported by any organization

[1] N. Sugie, H. Okuda and T. Suzuki are with Nagoya University, Furo-cho, Chikusa, Nagoya, Aichi, Japan h_okuda@nuem.nagoya-u.ac.jp

[2] K. Haraguchi and Z. Kang are with Toyota Technical Development Corporation, 1-9 Imae, Hanamoto-cho, Toyota, Aichi, Japan

978-1-7281-0322-8/18/$31.00 ©2018 IEEE

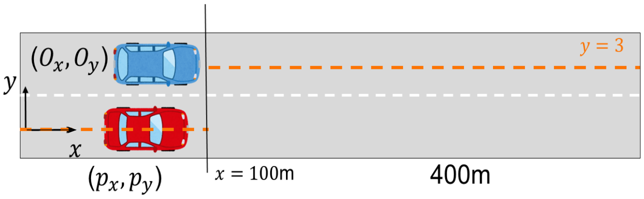


Fig. 1. Driving environment

in this paper for the simplicity. It is also assumed that the driver intend the lane changing after passing the point $x = 100$(100[m] ahead of start point.) This paper focus only on the lateral motion control of the ego vehicle in the NMPC framework.

III. Model predictive planning and control for lane-change behavior

A. Overview of proposed method

In order to achieve the planning and control simultaneously, non-linear model predictive control(NMPC) is used as the basic framework. MPC is well known as the promising technique to realize a real-time optimal control under some practical constraints [4], [8]–[10], [13].

The plant model used in the proposed NMPC is a so-called bicycle model, which has a time varying parameter, that is the driving speed, and is known to be a good approximated model to represent the lateral dynamical characteristics. The control input for the vehicle is the front tire angle δ.

In the standard control framework, the vehicle controller computes the control input so as to follow the reference path planned by the path planner in advance. In the proposed method, simple point reference, which denotes the final value of the state at the end of the lane changing task, is used in the NMPC framework together with the safety constraints.

Since the NMPC allows to consider nonlinear dynamics and constraints, various kinds of practical constraints for vehicle motion can be considered in the single framework. The C/GMRES method [11] is exploited to reduce the computational burden for finding the optimal control input in real-time.

B. State equations for NMPC

The equivalent bicycle model [12] depicted in Fig.2 is considered as the dynamics model of the ego vehicle. According to the equivalent bicycle model, the time derivatives of the lateral velocity and yaw rate of the vehicle are expressed as follows:

$$\begin{bmatrix} \dot{v_y} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\dfrac{a_{11}}{v_x} & \dfrac{a_{12}}{v_x} - v_x \\ -\dfrac{a_{21}}{v_x} & \dfrac{a_{22}}{v_x} \end{bmatrix} \begin{bmatrix} v_y \\ r \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \delta, \tag{1}$$

where, a_{ij} and b_k are the constant parameters of the vehicle $(i, j, k = 1, 2.)$ These parameters are computed from original physical parameters as follows:

$$a_{11} = (C_f + C_r)/M, \tag{2}$$
$$a_{12} = -(l_f C_f - l_r C_r)/M, \tag{3}$$
$$a_{21} = (l_f C_f - l_r C_r)/I_z, \tag{4}$$
$$a_{22} = -(l_f^2 C_f + l_r^2 C_r)/I_z, \tag{5}$$
$$b_1 = C_f/M, \quad b_2 = l_f C_f/I_z. \tag{6}$$

where the definition of each parameters are listed in Table I.

Here assuming that the vehicle speed V is constant and the slip angle β is small enough, the first and second order time derivatives of the lateral deviation from the reference path y_e are computed as follows:

$$\begin{aligned} \dot{y_e} &= V \sin(\beta + \theta_e) \simeq V(\beta + \theta_e) \\ &= v_y + V\theta_e, \end{aligned} \tag{7}$$
$$\ddot{y_e} = \dot{v_y} + V\dot{\theta}_e \tag{8}$$

where θ_e, the difference of the angle between the vehicle heading and the tangent of the reference path at the nearest point on the path is also assumed to be small. Since $\theta = \theta_e + \theta_d$, the yaw rate of the vehicle r is expressed as follows:

$$r = \dot{\theta}_e + \dot{\theta}_d = \dot{\theta}_e + \rho V, \tag{9}$$
$$\dot{r} = \ddot{\theta}_e \tag{10}$$

where θ_d is the yaw angle of the tangent of the reference path, and $\dot{\theta}_d$ is equivalent to ρV. In addition, the position of ego vehicle, (p_x, p_y), and the driving speed V are easily computed by

$$\dot{p_x} = V \cos\theta, \quad \dot{p_y} = V \sin\theta. \tag{11}$$

Note that although the tracking path is specified by the center of the straight road ($p_y = 0$), and ρ is always equal to be 0 (straight road) in this paper, this formulation can also deal with the lane-change behavior on the curvy road.

Now the state vector x and the input vector u can be defined from above equations as follows:

$$x = \begin{bmatrix} p_y, & \dot{p_y}, & \theta, & \dot{\theta}, & p_x \end{bmatrix}^{\mathrm{T}}, \quad u = \delta. \tag{12}$$

Finally, the nonlinear state equation is given by

$$\frac{\partial}{\partial t} x = f(x, u), \tag{13}$$
$$= \begin{pmatrix} \dot{p_y} \\ -\frac{a_{11}}{V}\dot{p_y} + a_{11}\theta + \frac{a_{12}}{V}\dot{\theta} + b_1\delta + (a_{12} - V^2)\rho \\ \dot{\theta} \\ -\frac{a_{21}}{V}\dot{p_y} + a_{21}\theta + \frac{a_{22}}{V}\dot{\theta} + b_2\delta + a_{22}\rho \\ V\cos\theta \end{pmatrix}. \tag{14}$$

C. Cost function for lane change task

For automated vehicle control, the cost function requires the reference state since the deviation from the reference state is evaluated as the cost. In independent motion planning and control scheme, the generated path planned by the path planner is given as the reference state in the cost function.

TABLE I

Parameters of vehicle model

	symbol	value	unit
Mass of the vehicle	M	1370	kg
Yaw moment of inertia	I_z	2870	$\mathrm{kg}\cdot\mathrm{m}^2$
Distance from gravity to front axis	l_f	1.11	m
Distance from gravity to rear axis	l_r	2.66	m
Cornering stiffness - Front tire	C_f	60000	N/rad
Cornering stiffness - Rear tire	C_r	30000	N/rad

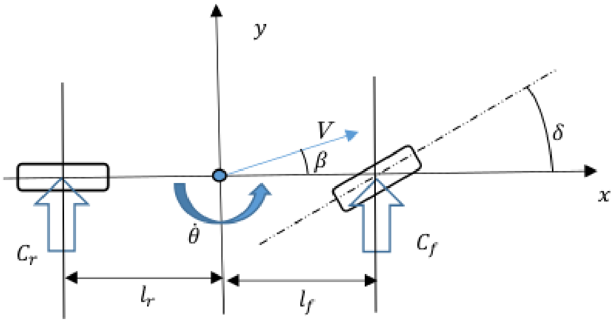


Fig. 2. Equivalent bicycle model for four wheeled car

In the proposed simultaneous design, only desired terminal state is required as the reference state instead of specifying detail of the reference path. This can be realized by switching the weight parameters in the cost function.

The cost function J for the input series $\hat{\mathbf{u}} = [\hat{u}_0, \cdots, \hat{u}_{N-1}]$ used in the proposed NMPC is given as follows:

$$J(\hat{\mathbf{u}}) = \frac{1}{2}\phi_N(\hat{x}_N) + \frac{1}{2}\sum_{k=0}^{N-1} L(\hat{x}_k, \hat{u}_k)h \qquad (15)$$

$$\phi_k(\hat{x}_k) = (\hat{x}_k - x_{ref})^{\mathrm{T}} Q_k (\hat{x}_k - x_{ref}) \qquad (16)$$

$$L_k(\hat{x}_k, \hat{u}_k) = \phi_k(\hat{x}_k) + \hat{u}_k^{\mathrm{T}} R_k \hat{u}_k, \qquad (17)$$

where h is the control interval and Q_k and R_k are weights for the state error and the input respectively. ϕ_k evaluates the penalty for the state error. L_k is the penalties evaluating both of the state error and the magnitude of input. $\hat{x}_k$ is the predicted state and x_{ref} is given reference state in the prediction horizon. Note that weights Q_k and R_k may not be constant but variables depending on the prediction state as explained in the section-E.

D. Setting of reference state for lane changing

As described in the previous section, the reference state and the weights must be set appropriately to achieve the desired task. In this paper, the reference state, $x_{ref}(t) = [p_y^{\mathrm{ref}}(t),\ \dot{p_y}^{\mathrm{ref}},\ \theta^{\mathrm{ref}},\ \dot{\theta}^{\mathrm{ref}},\ p_x^{\mathrm{ref}}\ ]$ is set to be $[p_y^{\mathrm{ref}}(t),\ 0,\ 0,\ 0,\ 0\ ]$ to keep driving straight on the desired lateral position $p_y^{\mathrm{ref}}(t)$. Since the lane change intention is supposed to appear when the ego vehicle arrives at $x = 100$ as denoted in section II, $p_y^{\mathrm{ref}}(t)$ is specified as follows:

$$p_y^{\mathrm{ref}}(t) = \begin{cases} 0 & (p_x(t) < 100) \\ 3 & (p_x(t) \geq 100) \end{cases} \qquad (18)$$

This state reference is extremely simple compared to setting the reference state trajectory based on the path planning.

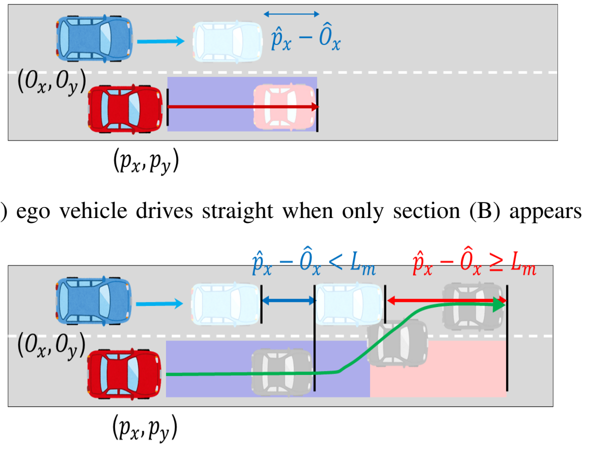
E. Setting of weight parameters for safety lane changing

In the proposed method, the weight parameters in the cost function are switched depending on the predicted state. This enables us to be free from specifying the complex reference state trajectory. The weight parameter for the state error, Q_k at the kth step of predicted states is determined by:

$$Q_k = \begin{cases} Q_A & (\|\hat{p}_{x_k} - \hat{O}_{x_k}\| \geq L_m) \\ Q_B & (\|\hat{p}_{x_k} - \hat{O}_{x_k}\| < L_m), \end{cases} \qquad (19)$$

where $\hat{p_{x_k}}$ and $\hat{O_{x_k}}$ are the predicted longitudinal position of the ego vehicle and the surrounding vehicle, respectively. This equation classifies each predicted steps into the two sections, section (A) and (B), according to the margin between the ego vehicle and surrounding vehicle. An example of the Q_A and Q_B are denoted in Table II, and the illustrative example of the weight parameter switching scheme is shown in Fig.3. Section (A) which is depicted as a red area in Fig.3 is the duration when the ego vehicle is expected to possible to change its driving lane with safe margin. Thise section is regarded as the section to accomplish the lane changing. The reference state for the lateral position p_y^{ref}, which is regarded as the intention of lane changing, is considered and the state error for the lateral position is penalized in this section in order to achieve the lane changing task. On the other hand, the section (B) which is depicted as the blue area in Fig.3 is considered as the transient section. The reference state for the lateral position p_y^{ref} and the yaw angle error θ_{e_k} are ignored in the cost function during this section. This implies that if the section (A) does not appear in later steps in the prediction horizon, ego vehicle drives more flexibly depending only on the constraints.

The section (A) is not appeared if the ego vehicle can not take enough margin to the surrounding vehicle over the prediction horizon as shown in Fig.3(a). In this case, ego vehicle drives straightly because of the weight parameters in cost function. If the section (A) appears after the section (B) as shown in Fig.3(b), ego vehicle tries to change its lane since the ego vehicle track the reference path in section (B) with considering the constraints. If only the section (B) appears in prediction horizon, the ego vehicle follows the reference path like standard path tracking controller. Note that section (A) also appears when the surrounding vehicle passes the ego vehicle and drives ahead with enough margin. By applying the weight parameter switching strategy, safe lane-change behavior is achieved without specifying complex state trajectory. This scheme is closer to the human drivers' natural lane-change behavior.



(a) ego vehicle drives straight when only section (B) appears

(b) ego vehicle change its driving lane when section (A) appeared after section(B)

Fig. 3. Switching scheme of objective function weights for safety lane changing

TABLE II
PARAMETERS USED IN SIMULATION

parameter	value
h	0.01
N	500
L_m	50
Q_A	diag[100 100 1 10000 0]
Q_B	diag[0 100 0 10000 0]
S_{fA}	diag[100 100 1 10000 0]
S_{fB}	diag[0 100 0 10000 0]
R_δ	2000

F. Input and state constraints

For practical use, proposed framework is required to consider the constraints in the input optimization process. Usually, the physical limitation of the vehicle such as a limitation on the control input, maximum acceleration and speed and so on must be considered to obtain feasible control input. At the same time, the constraints to achieve the desired logical condition, e.g. avoiding the prohibited area constraints for collision avoidance, have to be addressed to ensure the driving safety.

As one of the ideas to consider the collision avoidance constraint, an ellipsoid area is considered around the surrounding vehicle as prohibited area as shown in Fig.4. The constraints to prohibit the vehicle to enter the prohibited area can be written as:

$$(\frac{\hat{p}_{x_k} - \hat{O}_{x_k}}{r_x})^2 + (\frac{\hat{p}_{y_k} - \hat{O}_{y_k}}{r_y})^2 > 1, \tag{20}$$

for $^\forall k \in \{0, \cdots, N-1\}$ where the predicted position of ego vehicle is $(\hat{p}_{x_k}, \hat{p}_{y_k})$, the predicted position of surrounding car is $(\hat{O}_{x_k}, \hat{O}_{y_k})$ and the diameters of the prohibited area along x and y directions are r_x and r_y, respectively. Since this inequality constraints is sometimes difficult to be considered directly in gradient based nonlinear optimization method, this constraints are expressed as the equality constraints [7] by introducing slack variables $\hat{w}_k$ as a new input.

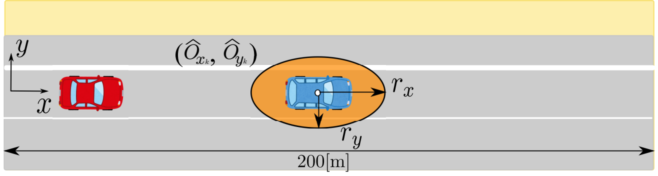


Fig. 4. Prohibition area for collision avoidance constraint

Please refer the references for the detail of implementation of inequality constraints [11].

Another approach to realize the constraints is to consider them as the soft constraints with introducing barrier function. Any kind of monotonic function can be used to generate the input which does not intrude prohibited area.

G. Formulation of input series optimization problem

Finally, the optimization problem at time t for NMPC is described as follows:

for given:

$$\begin{aligned} &Q_k, R_k, \hat{x}_0 = x(t), (\text{Table I}), (\text{Table II}), \\ &x_{ref}(t), O_{x_0}, O_{y_0} \end{aligned} \tag{21}$$

find:

$$\begin{aligned} &\hat{x}_k, \quad \hat{u}_k, \\ &(^\forall\ k \in \{1, \cdots, N-1\}), \end{aligned} \tag{22}$$

which minimize:

(Eq. 23),

subject to:

(Eq. 1) to (Eq. 11)

This input optimization must be solved in real-time in model predictive control framework.

H. Fast computation for non-linear MPC based on continuation

The optimization problem defined in previous section is solved at every control cycle in real-time in MPC framework. Since the problem is defined in nonlinear form, it requires the special treat to solve in real-time.

In [7], [11], a fast computation method for NMPC that is based on the continuation technique is presented. The essential idea of continuation is to track the optimality condition of the problem by feed-backing the deviation from the Karush-Kuhn-Tucker(KKT) condition in real-time.

A discrete-time nonlinear state equation $x_{t+1} = x_t + f(x_t, u_t)h$ where x is state vector and h is the time interval is assumed at the beginning. Control input series $\{u_k^*\}_{0:N-1}$ at the time t are the finding variables. $\{u_k^*\}_{0:N-1}$ is found as the optimal input which minimizes the following objective function

$$J(\{\hat{u}_k\}_{0:N-1}) = \Phi(\hat{x}_N) + \sum_{k=0}^{N-1} L(\hat{x}_k, \hat{u}_k)h. \tag{23}$$

where $\hat{x}_k$ and $\hat{u}_k$ denote the predicted state and optimized control input at k-th step of the prediction horizon. Then the Hamiltonian function using the co-state λ_k and the Lagrange multiplier for the constraint μ_k are the defined as follows:

$$H(\hat{x}_k, \hat{u}_k, \lambda_k, \mu_k) = \\ L(\hat{x}_k, \hat{u}_k) + \lambda_k^{\mathsf{T}} f(\hat{x}_k, \hat{u}_k) + \mu_k^{\mathsf{T}} C(\hat{x}_k, \hat{u}_k) \tag{24}$$

Then, a sufficient condition for optimality is given by

$$\hat{x}_0 = x_t,\ \lambda_N = \Phi(\hat{x}_N)^{\mathsf{T}}, \\ \begin{cases} \hat{x}_{k+1} = \hat{x}_k + f(\hat{x}_k, \hat{u}_k)h, \\ \lambda_k = \lambda_{k+1} + H_x(\hat{x}_k, \hat{u}_k, \lambda_{k+1}, \mu_{k+1})^{\mathsf{T}} h, \\ H_u(\hat{x}_k, \hat{u}_k, \lambda_{k+1}, \mu_k) = 0 \\ C(\hat{x}_k, \hat{u}_k) = 0 \end{cases} \tag{25}$$

for $\forall k \in \{0 : N-1\}$. Let $U = [\hat{u}_0,\ \mu_0,\ \hat{u}_1,\ \mu_1,\ \ldots,\ \hat{u}_{N-1},\ \mu_{N-1}]^{\mathrm{T}}$ be a vector obtained by concatenating all control input and Lagrange multiplier in prediction horizon. Here Eq. (25) can be seen as a nonlinear equation system $F(U, x_t) = 0$ of U.

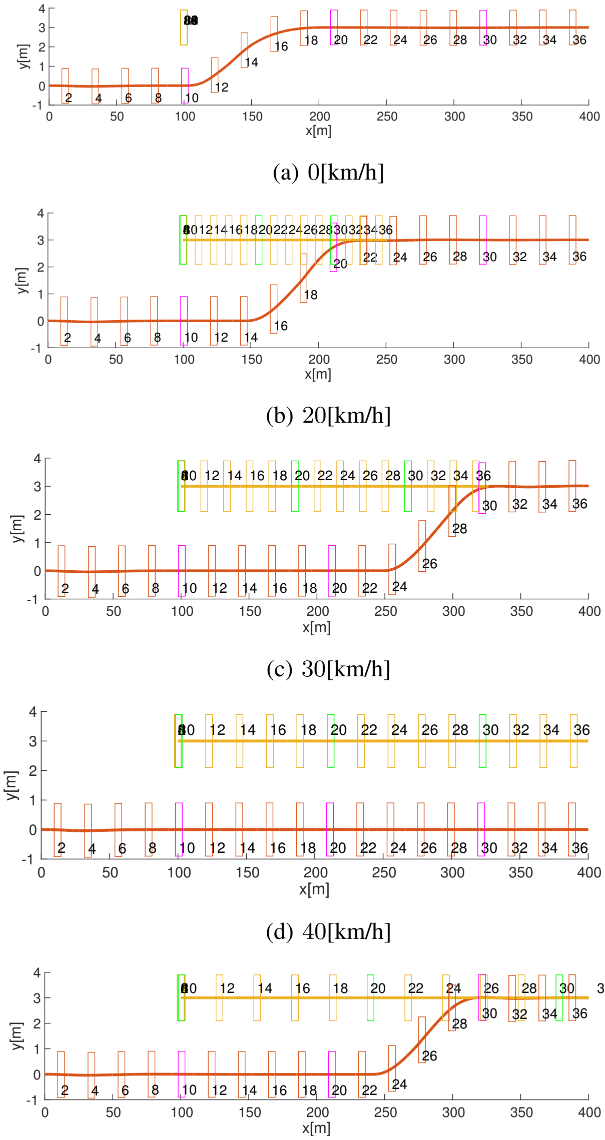
The essential idea of continuation is to fully utilize the optimization result of the previous control cycle instead of recomputing U at every control cycle. By supposing the optimal control input sequence U_{t-1} is obtained at time $t-1$, he optimal input at time t, U_t, is computed by utilizing U_{t-1}. Let $\delta x = x_t - x_{t-1}$ be the time difference of the state vector of the plant and δU be the amount of correction for the input sequence to track the optimal condition. Then the change of F(simple notation of $F(U, x_t)$) is given by

$$\delta F = F_U \delta U + F_x \delta x.$$

where the F_U is the partial derivatives of F with respect to U and F_x is the partial derivatives of F with respect to x. If the condition $\delta F = -\alpha F \quad (0 < \alpha < 1)$ is hold every control cycle, the value of F will converge to 0 gradually. To realize this behavior, δU is obtained as the solution of the following linear equations:

$$F_U \delta U = -(\alpha F + F_x \delta x). \tag{26}$$

Once δU is obtained, U_t is obtained by integrating δU as $U_t = U_{t-1} + \delta U$. In this manner, a single update of control input sequence can be done by solving a linear equations system (26) instead of computing a nonlinear optimization problem. In addition, Eq.26 can be approximately solved by GMRES method which is known as the fast computation technique to solve a large scaled linear equations with limited computation time. Thus the computational burden is kept small and that computation can be completed before next control step. The partial derivatives F_x and F_U can be calculated efficiently by the forward difference approximation. Other partial derivatives, H_x and H_u are obtained analytically once the cost function is given as C^2 continuous function.



(a) 0[km/h]

(b) 20[km/h]

(c) 30[km/h]

(d) 40[km/h]

(e) 50[km/h]

Fig. 5. Trajectory differences depending on the other car's velocity

IV. Simulator Experiment Results

Proposed method is implemented and tested in a driving simulator. The car dynamics of ego vehicle is computed with the vehicle dynamics simulator Carsim by Mechanical Simulation Corporation.

Figure 5 shows the resulting profiles of simulator experiments with proposed method. Horizontal axis of each figure is the longitudinal position and vertical axis shows the lateral position of the vehicles. The speed of the ego vehicle V is constant, 40[km/h]. The surrounding car departs from $O_x = 100$[m] and $O_y = 3$[m] when the targeting car arrives at $x = 100$[m]. The surrounding car goes straight at a constant speed without changing its driving lane nor driving speed. In Fig.5, the lane changing profiles with $0, 20, 30, 40, 50$[km/h] of surrounding vehicle speed are shown, respectively. Other

TABLE III

DISTANCE TO SURROUNDING CAR WHEN LANE CHANGING STARTS

V_O[km/h]	0	20	30	40	50
$p_x - O_x$[m]	0	22.2	36.1	- (failed)	-36.1

parameters used in simulation are shown in Table II.

In Fig.5, the positions of the ego vehicle and the surrounding vehicle are shown every 2 seconds by rectangles. The ego vehicle positions are shown by red rectangles and the surrounding vehicle positions are shown by yellow rectangles. Red and yellow solid line shows the trajectories of ego vehicle and surrounding vehicle, respectively. The color of rectangles changes to pink or green every 10 seconds. The numbers located near rectangles represent the elapsed time from the simulation started.

Table III shows the distance between two cars when the ego vehicle starts lane changing as the result of the appearance of section (A) in prediction horizon. In Fig.5-(d) which shows the resulting profile with setting the surrounding car speed 40[km/h], the ego vehicle can not change its driving lane since the ego vehicle and the surrounding car driving side-by-side with same driving speed. The ego vehicle drives faster than surrounding vehicle in Fig.5-(a), (b) and (c) and the lane change is done by cutting in into the front of the surrounding car. The surrounding car goes ahead of the ego car in Fig.5-(e) and the task is done by changing its driving lane on behind of the surrounding car. In Fig.5-(a), the ego vehicle starts lane changing just after arriving $p_x(t) = 100$. We can also see that the ego vehicle starts lane changing later in Fig.5-(c) compared to Fig.5-(b) since it takes more time to have enough margin to the surrounding car in the case Fig.5-(c). On the other hand, the ego vehicle could have the enough margin to the surrounding car behind it after the surrounding car leaves in Fig.5-(e). It is confirmed that the proposed method can achieve the safety lane changing with considering safety margin in future. Proposed method can decide the lane changing start time automatically with just setting preferable margin but without explicit reference path to follow. This is the most big advantage of using proposed simultaneous motion planning and control framework.

Finally the fast computation characteristics of proposed method is checked. The statistics of computation time required for the 1 frame computation in the simulation experiment is shown in Table IV. This statistics is from the experiment of the case (a) in Fig.5 but it is not affected by surrounding car setting. It was found that the computation time required to update the input with C/GMRES method is just 6.84[msec] in average. In addition, since the total computation time is less than the total simulation time as shown in Table IV, it is confirmed that the proposed method can achieve real-time control. Now the fast computation availability of the proposed method is confirmed.

V. CONCLUSION

The simultaneous planning and control framework based on the nonlinear model predictive control is proposed and applied to lane-change task. In the proposed NMPC, the vehicle dynamics, the physical limitation and the safety requirements are considered as the constraints. In addition, the parameters of cost function is switched depending on the predicted state variables instead of changing the reference. This means the proposed method requires only the information on control objectives and constraints, and does not need to consider the path planning explicitly. The proposed method was implemented and the validity is checked in simulator experiment. Although the NMPC usually requires huge computational cost, it is confirmed that the proposed method achieve real time computation by applying the (C/GMRES) method.

TABLE IV

STATISTICS ON COMPUTATION TIME

	time per step[ms]
maximum	8.01
minimum	6.05
average	6.84
total computation time	25,386
(total simulation time)	37,090

Application for more complex driving environment such as curvy road, real vehicle experiment and to consider multiple surrounding vehicle are future works.

REFERENCES

[1] B. Paden, M. Cap, S. Z. Yong, D. Yershov and E. Frazzoli, “A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles,” *IEEE Trans. on Intelligent Vehicles*, Vol. 1, No. 1, pp. 33–55, 2016.

[2] J. P. Laumond, P. E. Jacobs, M. Taix and R. M. Murray, ”A motion planner for nonholonomic mobile robots,” *IEEE Trans. on Robo. and Auto.*, Vol. 10, No. 5, pp. 577–593, 1994.

[3] L. Han, H. Yashiro, H. Tehrani Nik Nejad, Q. H. Do and S. Mita, ”Bezier curve based path planning for autonomous vehicle in urban environment,” in *2010 IEEE Intelligent Vehicles Symposium*, pp. 1036-1042, 2010

[4] A. Koga, H. Okuda, Y. Tazaki, et al.: “Autonomous Lane Tracking Reflecting Skilled/Un-skilled Driving Characteristics,” in *42th Annu. Conf. of the IEEE Industrial Ele. Soc.*, pp.3175–3180, 2015.

[5] S. A. Arogeti and N. Berman, ”Path Following of Autonomous Vehicles in the Presence of Sliding Effects,” *IEEE Trans. on Vehicular Tech.*, vol. 61, no. 4, pp. 1481-1492, 2012.

[6] C. Gotte, M. Keller, C. Hass, K. H. Glander, A. Seewald and T. Bertram, ”A model predictive combined planning and control approach for guidance of automated vehicles,” in *2015 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pp. 69–74, 2015

[7] M.Otsuka, Realtime collision avoidance control based on continuation method for nonlinear model predictive control with safety constraint, in *2017 11th Asian Control Conference (ASCC)*, pp. 1086-1091, 2017

[8] M.Oshima and M.Ogawa, “Model Predictive Control-I : Basic Principle : history & present status,” *Trans. on the Institute of Sys., Cont. and Info. Eng.*, vol.46, NO.5, 2002

[9] M.Kano and M.Oshima, “Model Predictive Control-II : Linear Model Predictive Control,” *Trans. on the Institute of Sys., Cont. and Info. Eng.*, vol.46, NO.7, 2002

[10] P. Falcone, F. Borrelli, J. Asgari, et al.: “Predictive Active Steering Control for Autonomous Vehicle Systems,” *IEEE Trans. on Control Sys. Tech.*, Vol.15, No.3, 2007.

[11] M.Otsuka, A continuation/GMRES method for fast computation of nonlinear receding horizon control, *Automatica*, Vol. 40, No. 4, pp. 563–574, 2004

[12] M.Abe, “Automotive Vehicle Dynamics,” *Tokyo Denki University Press*, 2008

[13] T. Kawabe, H. Nishira and T. Ohtsuka, “An Optimal Path Generator Using a Receding Horizon Control Scheme for Intelligent Automobiles,” in *2004 IEEE Conf. on Cont. App.*, pp. 1597–1602, 2004