PAPER

Simultaneous Realization of Decision, Planning and Control for Lane-Changing Behavior Using Nonlinear Model Predictive Control

Hiroyuki OKUDA^{†a)}, Member, Nobuto SUGIE[†], Nonmember, Tatsuya SUZUKI[†], Member, Kentaro HARAGUCHI^{††}, and Zibo KANG^{††}, Nonmembers

SUMMARY Path planning and motion control are fundamental components to realize safe and reliable autonomous driving. The discrimination of the role of these two components, however, is somewhat obscure because of strong mathematical interaction between these two components. This often results in a redundant computation in the implementation. One of attracting idea to overcome this redundancy is a simultaneous path planning and motion control (SPPMC) based on a model predictive control framework. SPPMC finds the optimal control input considering not only the vehicle dynamics but also the various constraints which reflect the physical limitations, safety constraints and so on to achieve the goal of a given behavior. In driving in the real traffic environment, decision making has also strong interaction with planning and control. This is much more emphasized in the case that several tasks are switched in some context to realize higher-level tasks. This paper presents a basic idea to integrate decision making, path planning and motion control which is able to be executed in realtime. In particular, lane-changing behavior together with the decision of its initiation is selected as the target task. The proposed idea is based on the nonlinear model predictive control and appropriate switching of the cost function and constraints in it. As the result, the decision of the initiation, planning, and control of the lane-changing behavior are achieved by solving a single optimization problem under several constraints such as safety. The validity of the proposed method is tested by using a vehicle simulator.
key words: model predictive control, simultaneous path planning and motion control, lane-changing

1. Introduction

Autonomous driving becomes a promising solution for problems in the mobility of future society, such as traffic congestion, accidental injury, environmental pollution, and so on [1]. The autonomous driving is particularly aspired from elderly and/or handicapped people who is losing sufficient driving skills. Even normal drivers sometimes prefer the autonomous driving in the case that they want to spend more time for their jobs or for communications in the car. Although these motivations undoubtedly accelerate the implementation of fully autonomous driving on the traffic society, it may take some decades to complete the transition to autonomous driving society. In the duration of the transition, the autonomous/manual mixed traffic situation must

be considered and be reflected in the design of autonomous driving. From this viewpoint, the autonomous driving system is expected to be in charge of making consensus with surrounding vehicles which may be controlled by a human driver.

Typical components to realize the autonomous driving are the localization, driving environment recognition, decision making, path planning, and motion control. The path planning and motion control are fundamental components to realize safe, reliable and comfortable autonomous driving [2]. Generally speaking, the reference path which achieve the driving objectives such as following the driving lane, following a lead car, avoiding the obstacles and so on is planned [2]–[4] at first, and the path is given to the motion controller to track the created path [5]–[10].

These two functions essentially have strong mathematical interactions, i.e., the planned path must be feasible by the motion controller under various physical constraints. Therefore, the obtained solution (control input) may not be desirable when these two functions are considered independently. Although the increase of the considered constraints for path planning may resolve such problem, this results in the redundant computation in the implementation of the path planning and motion control. In order to solve this problem, the idea of simultaneous path planning and motion control (SPPMC) is considered and attracting great attention [11]–[15]. In SPPMC, a unified optimization problem to find the optimal control input is addressed by considering not only the vehicle dynamics but also the goal of the task and various constraints such as the physical limitations, safety constraints and so on. SPPMC is well studied in the robotics field as shown in the papers [16]–[18]. Although off-line computation is sometimes applied since the SPPMC requires high computational cost, the realtime computation framework is necessary for the automotive application because of time-varying characteristics, uncertainty and openness of the driving environment.

In addition, as the demand for the harmonized driving with surrounding vehicles rises, the importance of the function of decision making is highly regarded [19]–[23]. There also exists a strong interaction between decision making and path planning/motion control. It is highly desired to develop a framework to realize the decision making, path planning and motion control as a single package for a future

Manuscript received February 26, 2020.

Manuscript revised July 21, 2020.

Manuscript publicized August 31, 2020.

[†]The authors are with Nagoya University, Nagoya-shi, 464–8603 Japan.

^{††}The authors are with Toyota Technical Development Corporation, Toyota-shi, 470–0334 Japan.

a) E-mail: h_okuda@nuem.nagoya-u.ac.jp

DOI: 10.1587/transinf.2020EDP7039

autonomous vehicle. This will lead to the development of an explainable driving intelligence package.

Based on these backgrounds, this paper presents how to integrate the functions of path planning, motion control and decision making particularly focusing on lane-changing behavior. Lane changing behavior is one of tasks which require high cognitive workload and the driver have to decide the position to cut in among surrounding cars driving on main lane [24]. Although many literatures have studied the realization of automatic lane changing with decision making [7], [19], [25], [26], those works must execute the path planning and motion control separately with facing the redundancy problem mentioned above. On the other hand, some literatures which tried SPPMC based on the system control technique could not handle the decision making in the framework because the decision making part is usually discrete time event and handled out of the SPPMC. [12]–[14], [20], [27]–[29]. The proposed architecture is based on a nonlinear model predictive control (MPC) [6], [9], [30]–[32] with a switched cost function, and it can find not only smooth path and control input of the vehicle but also the instant of initiating the lane-changing behavior and the place to merge on the target lane. In the proposed MPC, some weighting parameters and the reference position (not path) on the lane, which is specified in the cost function, are switched depending on the driving situation. The validity of the proposed method is tested in the simulation experiment using a vehicle dynamics simulator with targeting a simple lane-changing behavior focusing on the proof of controller concept.

2. Problem Setting

2.1 Target Task

A lane-changing behavior, which is shown in Fig. 1, is addressed in this study. The control target is Car E which is driving in the right lane in Fig. 1. The Car E tries to change the driving lane after the Car E reaches at the point $x = 100$ (lane-changing behavior is initiated after a certain condition is satisfied). There are two cars, Car 1 and Car 2, on the left lane and they drive straightly without changing their driving lane. Car E can select where to cut-in ahead of Car 1 (point 3), between Car 1 and Car 2 (point 2) or behind of Car 2 (point 1, in Fig. 1) depending on the situation. The driving speed of Car E, Car 1 and Car 2 are assumed to be constant in this study for the simplicity of the problem, however, this limitation can be relaxed by modifying the behavior model of cars and the constraints.

Note that the addressed task in this paper is not a general lane-changing behavior but one specified lane-changing situation of them [24]. More general lane-changing tasks must be considered in the real application, such as the lane-changing situation considering leader and follower cars of ego car, sudden driving speed change of surrounding cars, number of driving lanes, double lane-change situation and so on. This paper, however, is focusing on establishing how

Fig. 1 Considered lane-changing behavior

to handle the ‘decision making’ aspect in the SPPMC framework and the targeting task is chosen to make the problem simple for the proof of its concept. The validation with various lane-changing scenario is necessary to confirm the usefulness and the applicability of the proposed method to car control application in a realistic driving environment.

2.2 Features of Lane-Changing Behavior

Generally speaking, lane-changing behavior is executed by the following procedure:

1. cruising current driving lane,
2. selecting where to cut-in,
3. decision making about the timing of lane change
4. generating the reference path to change the driving lane
5. following the reference path.

The lane cruising and the reference path following are usually executed by the vehicle controller regarding the center of the lane and/or the planned path. The reference path generation is sometimes executed by the continuous curve generation method [3], [4], [28]. On the other hand, the cut-in position and the timing to change the driving lane are decided according to the simple logic or machine learning considering driving safety. [19], [20], [24], [25] In Fig. 1, there are four possible location for Car E to cut-in as follows:

- entering at front of Car 1,
- entering between Car 1 and Car 2,
- entering behind Car 2,
- no lane changing.

The choice of the target location strongly affects on the path planning and motion control. Such interaction among decision making, path planning and motion control makes the design of the lane-changing behavior difficult.

3. Simultaneous Decision Making, Path Planning and Motion Control Platform

3.1 Outline of Proposed System

To overcome the difficulty mentioned in the previous section, this paper proposes a new framework for lane-changing behavior based on unified optimization and implemented by receding horizon manner. Figure 2 shows the architecture of the proposed system. The proposed control system can be regarded as one of the non-linear model predictive control (MPC).

Fig. 2 Outline of proposed system

MPC is a general framework which can accept various objective functions and constraints depending on the goal of the control system. As shown in Fig. 2, selected task by the driver's intention or the higher-level planner gives the set of the reference state (e.g. the lateral position of the center of desired driving lane) and the constraints (e.g. constraints for collision avoidance) to the optimizer. One of the key idea to realize the unification is to use the common MPC across the various tasks as long as possible by switching the reference state and the constraints instead of switching the MPC controller itself. This makes easy to ensure the continuity of the control input thanks to the consideration of the dynamics and the physical limitation of the cars.

Because of the nature of MPC as the regulation problem, the proposed system can accept not only the reference path but the reference state which means the goal of the selected task to determine the optimal state trajectory and the control input. Other interesting features of lane-changing behavior realized by the proposed system are the decision on the timing and the location to cut-in. The timing to depart the driving lane and the place to cut-in must be decided to determine the reference state or to switch the controller in advance usually according to the safety discriminant in conventional MPC. On the other hand, the proposed system applies the switched cost function and the switched constraints to realize the path planning and motion control together with the decision making in single optimizer without switching neither the reference nor the controller itself. Note that the timing and the selection of the place to enter are not included explicitly as the variable but decided implicitly in the resulting optimal state trajectory. This may help to consider the stability of the controller from the viewpoint of Lyapunov function since the reference tracking problem is sometimes more difficult to consider its stability than the regulation problem. The discussion about the stability is beyond the scope of this paper.

3.2 Outline of Mathematical Formulation

Although the proposed system is a general framework which can be applied to various practical tasks in driving situations, the lane-keeping task and the lane-changing task are

Table 1 Parameters of vehicle model

	symbol	value	unit
Mass of the vehicle	M	1370	kg
Yaw moment of inertia	I_z	2870	$\text{kg} \cdot \text{m}^2$
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

Table 2 Definition of variables

	variable	unit
Position of Car E	p_x, p_y	[m]
Speed of Car E	v_x, v_y	[m/s]
Yaw angle of Car E	θ	[rad]
Yaw rate of Car E	r	[rad/s]
Lateral deviation from center of lane of Car E	y_e	[m]
Driving speed in local coordinate of Car E	V	[m/s]
Curvature of center of lane	ρ	[1/m]
Yaw angle of tangent of driving lane	θ_d	[rad]
Tire angle input for Car E	δ	[rad]
Index of cars	$c \in \{1, 2, E\}$	-
Positions of Car c	p_x^c, p_y^c	[m]

considered as the examples in this section too. The goal of the optimization problem is to find the optimal input profiles and the state trajectory in several seconds future. The constraints and objective function considered in this problem are changed depending on the task as mentioned in the previous section. The optimization problem for the proposed system is formulated in the following sections. The C/GMRES (Continuation / Generalized Minimum RESidual) method [30] which is the fast computation method for the non-linear model predictive control explained later is utilized in this paper to obtain the optimal input in each control step in real-time. Thanks to the capability of the method, the objective function, the vehicle dynamics, and the other constraints can be formulated as nonlinear functions.

3.3 Input and State

Firstly, input and state variables are defined. The control input vector u_t at the time of t is defined as follow:

$$u_t = \{\delta_t\}, \quad (1)$$

where δ_t is the tire angle of ego car (Car E). Generally, not only the tire angle but also the speed or the acceleration of the car should be considered in SPPMC. However, the constant driving speed is assumed for all cars from the viewpoint of computational complexity while this extension of the problem would be straightforward.

The state vector x_t is defined as

$$x = [p_y, \dot{p}_y, \theta, \dot{\theta}, p_x]^T, \quad (2)$$

where p_x and p_y are the longitudinal and the lateral position of Car E, θ is the yaw angle of Car E, respectively, as shown in Table 2. The definition of the input vector and the state vector should be kept as consistent as possible among the target tasks.

Fig. 3 Equivalent bicycle model for four wheeled car

3.4 Behavior Models

Secondly, the behavior models of the Car E is introduced. The predicted positions of Car 1 and Car 2 over the prediction horizon are given to the problem by applying the linear prediction instead of considering the behavior models of other cars explicitly in the state equation. The behavior models of other cars must be included in state equation if there exists any interaction among Car E, Car 1 and Car 2.

The equivalent bicycle model [5] shown in Fig. 3 is considered as the behavior model of the Car E. According to the equivalent bicycle model, the time derivatives of the lateral velocity and yaw rate of the vehicle are expressed as follows:

$$\begin{bmatrix} \dot{v}_y \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\frac{a_{11}}{v_x} & \frac{a_{12}}{v_x} - v_x \\ -\frac{a_{21}}{v_x} & \frac{a_{22}}{v_x} \end{bmatrix} \begin{bmatrix} v_y \\ r \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \delta, \quad (3)$$

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

$$a_{11} = (C_f + C_r)/M, \quad (4)$$

$$a_{12} = -(l_f C_f - l_r C_r)/M, \quad (5)$$

$$a_{21} = (l_f C_f - l_r C_r)/I_z, \quad (6)$$

$$a_{22} = -(l_f^2 C_f + l_r^2 C_r)/I_z, \quad (7)$$

$$b_1 = C_f/M, \quad (8)$$

$$b_2 = l_f C_f/I_z, \quad (9)$$

with setting each physical parameters as listed in Table 1.

The driving speed of the ego car, V , is assumed to be constant in this paper as mentioned in previous sections. Assuming the slip angle β is small enough, the first and second order time derivatives of the lateral deviation from the center of lane y_e can be computed as follows:

$$\begin{aligned} \dot{y}_e &= V \sin(\beta + \theta_e) \approx V(\beta + \theta_e) \\ &= v_y + V\theta_e, \end{aligned} \quad (10)$$

$$\ddot{y}_e = \dot{v}_y + V\dot{\theta}_e \quad (11)$$

where θ_e , the difference of the angle between the vehicle heading and the tangent of the driving lane, is also assumed

to be small. The benefit to consider the error system form the center of the driving lane is that the obtained formula can be applied not only to the straight road but also to the curvy road by considering the curvature of the road. Since $\theta = \theta_e + \theta_d$, the yaw rate of the vehicle r is expressed as follows:

$$r = \dot{\theta}_e + \dot{\theta}_d = \dot{\theta}_e + \rho V, \quad (12)$$

$$\dot{r} = \ddot{\theta}_e \quad (13)$$

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

$$\dot{p}_x = V \cos \theta, \quad \dot{p}_y = V \sin \theta. \quad (14)$$

Note that the $\rho = 0$ is always set to be zero since the straight road is considered in this paper.

Finally, the behavior model of Car E formulated as the state equation is given as follows:

$$\begin{aligned} \frac{\partial}{\partial t} x &= f(x, u), \\ &= \begin{bmatrix} -\frac{a_{11}}{V} \dot{p}_y + a_{11}\theta + \frac{a_{12}}{V} \dot{\theta} + b_1\delta + (a_{12} - V^2)\rho \\ \dot{\theta} \\ -\frac{a_{21}}{V} \dot{p}_y + a_{21}\theta + \frac{a_{22}}{V} \dot{\theta} + b_2\delta + a_{22}\rho \\ V \cos \theta \end{bmatrix}. \end{aligned} \quad (15)$$

This state equation is linear if the constant speed is assumed. However, this state equation turns into a nonlinear equation when the driving speed is treated as the control variable together with the tire angle.

3.5 Constraints for Safety Driving

One benefit to solve the unified SPPMC problem is to consider the feasibility of the optimal trajectory regarding the physical limitations and/or the other various constraints. For examples, the physical limitations of the car, the constraints to achieving the given task (e.g. driving the certain point at a certain speed) and the safety constraint, which can be regarded as one of the most important issue on the motion control as shown by a number of studies [1], can be included.

The safety constraints,

$$p_y \leq C_y \quad \text{if } \exists c \in \{1, 2\}, p_x^c - D \leq p_x \leq p_x^c + D, \quad (16)$$

where c is the index of the surrounding cars and p_x^c is the longitudinal position of the Car c as shown in Table 2, is introduced to realize the collision avoidance to the other cars in this paper.

Here $D[m]$ is the length of the overlapped area between Car E and the others in the longitudinal direction and C_y is the lateral position of the border of the prohibited area not to collide other cars in the lateral direction. The parameters C_y and D can be decided not only from the physical size of

Fig. 4 Enabled area of log barrier function

cars in the driving situation but also to take a margin for the safety and fearless driving. Note that the parameter D is particularly set to larger value than its physical size in order to take a enough safety margin for the case when surrounding cars change their driving speed suddenly. $D = 20[\text{m}]$ used in this paper, for example, means $1.8[\text{s}]$ of safety margin in time head way.

Generally speaking, the gradient-based optimization algorithm such as the C/GMRES method can not handle the hard inequality constraint directly. One of an idea to implement the inequality constraints is to use the slack variable [33]. The detail of how to use inequality as the hard constraints is directed in the references [30], [33], however, another idea, which introduces a barrier function as the soft constraint, is used in this paper. The barrier function applied in this paper is as follows:

$$J_b = \sum_{c=1}^2 J_b^c \quad (17)$$

$$J_b^c = \begin{cases} -k \log(C_y - p_y) & \text{if } p_x^c - D < p_x \wedge \\ & p_x < p_x^c + D \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

This barrier function is switched depending on the relative distance between the Car E and the other cars in longitudinal direction. These functions take a non-zero value only when the ego car and another car drives side by side as shown in Fig. 4. It is desirable if it is possible to consider the constraint with 'if' condition and the logical operations from the viewpoint of the desirability than using the approximated simple shape to express the prohibited area as shown in [27].

3.6 Cost Function

As usual MPC, the deviation from the reference state is evaluated in the cost function. In addition, the cost function also includes the terms for soft constraints denoted in the previous section.

The cost function J computed from the input series $\hat{\mathbf{u}}(t) = [\hat{u}_0, \dots, \hat{u}_{N-1}]$ at the time of t is given as follows:

$$J(\hat{\mathbf{u}}(t)) = \frac{1}{2} \phi(\hat{\mathbf{x}}(N|t)) + \frac{1}{2} \sum_{k=0}^{N-1} \{L(\hat{\mathbf{x}}(k|t), \hat{\mathbf{u}}(k|t)) + J_b(k|t)\}h \quad (19)$$

$$\phi(\hat{\mathbf{x}}(k|t)) = (\hat{\mathbf{x}}(k|t) - \mathbf{x}_{ref}(k|t))^T Q_k (\hat{\mathbf{x}}(k|t) - \mathbf{x}_{ref}(k|t)) \quad (20)$$

$$L(\hat{\mathbf{x}}(k|t), \hat{\mathbf{u}}(k|t)) = \phi(\hat{\mathbf{x}}(k|t)) + \hat{\mathbf{u}}(k|t)^T R_k \hat{\mathbf{u}}(k|t), \quad (21)$$

where h is the control interval and Q_k and R_k are weights for the state error and the input, respectively. The ϕ_k evaluates the penalty for the state error and the L_k is the penalties evaluating both of the state error and the magnitude of the input. $\hat{\mathbf{x}}_k$ is the predicted state and $\mathbf{x}_{ref}$ is given reference state. While the reference $\mathbf{x}_{ref}$ is switched according to the selected task, the parameters Q_k is also switched in a lane-changing task as explained in the following sections. $R_k = R_\delta$ is assumed to be constant and set by cut and try.

3.7 Switching of Reference State Depending on Task

In contrast to that the some conventional path tracking controller requires the planned path and/or the driving lane information from the map, the proposed SPPMC framework requires only the desired terminal state depending on the selected task. The reference state used in this paper is defined as $\mathbf{x}_{ref}(t) = [p_y^{\text{ref}}(t), \dot{p}_y^{\text{ref}}, \theta^{\text{ref}}, \dot{\theta}^{\text{ref}}, p_x^{\text{ref}}]$ and is set to be $[p_y^{\text{ref}}(t), 0, 0, 0, 0]$ to keep driving straight on the desired lateral position $p_y^{\text{ref}}(t)$ in this paper. The reference lateral position is switched when the Car E approaches at the initiation point $x = 100[\text{m}]$ and the selected task is changed. The $p_y^{\text{ref}}(t)$ can be simply specified as follows:

$$p_y^{\text{ref}}(t) = \begin{cases} 0 & \text{if } p_x(t) < 100 \text{ (right lane)} \\ 3 & \text{if } p_x(t) \geq 100 \text{ (left lane)} \end{cases}, \quad (22)$$

from the physical meaning of the tasks. This switching can be regarded as the change of the intention of the autonomous system or the human driver to try the lane-changing.

3.8 Design of Switched Cost Function for Safety Lane-Changing

In the proposed method, the weight parameters in the cost function are switched depending on the predicted state in the prediction horizon. Note that only the lane-changing task after the reference is changed is discussed in this section since no switching is necessary in the lane-keeping task.

While the lane-change task is started by changing the reference state, the proposed system does not initiate the lane-changing behavior immediately. The initiation timing and the position to cut-in is determined implicitly by the optimization process of the control input and the state trajectory with considering the safety margin thanks to the switched cost function. The weight parameter for the state error $Q(k|t)$ at the k th step of prediction horizon at the time t is written as follows:

$$Q(k|t) = \begin{cases} Q_A & \text{if } \hat{d}^{\min}(k|t) < L_m, \\ Q_B & \text{if } \hat{d}^{\min}(k|t) \geq L_m, \end{cases} \quad (23)$$

$$\text{where } \hat{d}^{\min}(k|t) = \min_{c \in \{1,2\}} |\hat{p}_x(k|t) - \hat{p}_x^c(k|t)| \quad (24)$$

Table 3 Parameters used in simulation

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

(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. 5 Switching scheme of objective function weights for safety lane-changing

where $\hat{p}_x(k|t)$ and $\hat{p}_x^c(k|x)$ are the predicted longitudinal position of the ego car and the other car, respectively.

An example of the Q_A and Q_B are shown in Table 3, and the illustrative example of the weight parameter switching scheme is shown in Fig. 5. This switched Q function means that the deviation of the lateral position and the yaw angle from the reference state are considered by using Q_B when the enough margin is predicted among the predicted positions of Car E ($\hat{p}_x(k|t)$) and Car c ($\hat{p}_x^c(k|t)$) (section B in Fig. 5) while they are ignored otherwise (section A in Fig. 5.) The section A which is depicted as the blue area in Fig. 5 can be considered as the transient section to wait the initiation of the lane-changing behavior to keep the enough margin.

The Car E may start lane-changing behavior immediately and drives with almost contacting the other cars after changing the reference if the switching cost function is not introduced. This behavior is quite unnatural even if the collision-free is ensured by the safety constraints. Although the safety margin L_m must be designed to determine the characteristics of the lane-changing behavior, the system designer can decide L_m without being too conservative since the collision-free is guaranteed by the safety constraint explicitly.

There can be another idea to switch the reference state depending on the predicted state instead of switching the parameter. It is not guaranteed anymore whether the solution

will be converged because there exist strange loop between the state and the reference state if the reference state is the function of the predicted state. Section A and B must be determined by another planner in advance in that case, and it may not be simultaneous optimization anymore.

3.9 Formulation of Input Series Optimization Problem

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

for given:

$$Q_A, Q_B, R_\delta, \hat{x}(0|t) = x(t), (\text{Table 1}), (\text{Table 3}),$$

$$x_{ref}(t), p_x^1(k|t), p_y^1(k|t), p_x^2(k|t), p_x^2(k|t)$$

$$\forall k \in \{0, \dots, N-1\} \quad (25)$$

find:

$$\hat{x}(k|t), \hat{u}(k|t),$$

$$(\forall k \in \{1, \dots, N-1\}), \quad (26)$$

which minimize:

$$(\text{Eq. (19)}),$$

subject to:

$$(\text{Eq. (14)}), (\text{Eq. (15)}), (\text{Eq. (18)})$$

This input optimization must be solved in realtime by using the C/GMRES method [30], [33] summarized in next section.

3.10 Fast Computation for Non-Linear MPC Based on Continuation

The optimization problem defined in the previous section must be solved every control cycle in realtime in non-linear MPC framework. In [30], [33], a fast computation method, which is called C/GMRES method (Continuation / Generalized Minimum RESidual), for non-linear MPC that is based on the continuation, technique is presented. The key idea of continuation is to track the necessary optimality condition by feed-backing the deviation from the Karush-Kuhn-Tucker (KKT) condition in realtime. Here the outline of C/GMRES method as the general MPC (without inequality constraints) is introduced. In MPC, the series of control input $\{u_k^*\}_{0:N-1}$ are the finding variables with assuming a discrete-time nonlinear state equation $x_{k+1} = x_k + f(x_k, u_k)h$ where x_k is the state vector at the time step k and h is the time interval, respectively. $\{u_k^*\}_{0:N-1}$ is found as the optimal input which minimizes the following objective function

$$J(\{\hat{u}_k\}_{0:N-1}) = \Phi(\hat{x}_N) + \sum_{k=0}^{N-1} L(\hat{x}_k, \hat{u}_k)h. \quad (27)$$

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 defined as follows:

$$H(\hat{x}_k, \hat{u}_k, \lambda_k, \mu_k) = L(\hat{x}_k, \hat{u}_k) + \lambda_k^T f(\hat{x}_k, \hat{u}_k) + \mu_k^T C(\hat{x}_k, \hat{u}_k). \quad (28)$$

Here the condition,

$$\begin{cases} \hat{x}_0 = x_t, \lambda_N = \Phi(\hat{x}_N)^T, \\ \hat{x}_{k+1} = \hat{x}_k + f(\hat{x}_k, \hat{u}_k)h, \\ \lambda_k = \lambda_{k+1} + H_x(\hat{x}_k, \hat{u}_k, \lambda_{k+1}, \mu_{k+1})^T h, \\ H_u(\hat{x}_k, \hat{u}_k, \lambda_{k+1}, \mu_k) = 0 \\ C(\hat{x}_k, \hat{u}_k) = 0 \end{cases}, \quad (29)$$

gives the sufficient condition for optimality for $\forall k \in \{0 : N-1\}$ where the x_t denotes the measurement at the time of t . Then, by letting $U = [\hat{u}_0, \mu_0, \hat{u}_1, \mu_1, \dots, \hat{u}_{N-1}, \mu_{N-1}]^T$ be a vector composed of all control input and Lagrange multiplier in prediction horizon, (Eq. (29)) can be seen as a non-linear equation system $F(U, x_t) = 0$ of U when the x_t is given.

The key idea of the continuation is using the optimal control input obtained in the previous control cycle instead of recomputing U at every control cycle. 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 respecting that the U and the F_x are the partial derivatives of F with respect to x . The value of F can be converged to 0 gradually when the condition $\delta F = -\alpha F$ ($0 < \alpha < 1$) is hold every control cycle. To realize this behavior, δU is obtained as the solution of the following linear equations:

$$F_U \delta U = -(\alpha F + F_x \delta x). \quad (30)$$

This process can be regarded as a kind of feedback to regulate the F to be zero. 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 the control input sequence can be done by solving a linear equations system (Eq. (30)) instead of computing a nonlinear optimization problem. In addition, (Eq. (30)) can be approximately solved by GMRES method which is known as the fast computation technique to solve a large-scale linear equations with limited computation time. Thus the computational burden is kept small and that computation can be completed before the 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. Note that the Hamiltonian function in the proposed system in this paper is not differentiable at the border of the switching. The gradient near the discontinuity is replaced by the gradient of

the nearest continuous point. Although this approximation is a rough approximation, the obtained problem is enough practical as shown in the latter simulation experiment.

4. Simulation Results

4.1 Setting of the Simulation

Here the simulation setting is explained in this section. The Car E starts from $x_E = 0$ and initiate the lane-changing by setting the reference of lateral position of the cost function to be $p_y^{ref}(t) = 3$ when the Car E reaches to $p_x(t) > 100$ [m]. The other parameters are set as shown in Table 1 and Table 3. The control interval h and the number of prediction steps N must be chosen considering the computational speed and the preferable prediction time. Five second is chosen for the prediction time arbitrary regarding the time duration needed for accomplishment of lane change at first, and the $N = 500$ with $h = 0.01$ [s] are selected in order to finish the computation in each control frame. $L_m = 50$ [m] which is used to the condition to switch the weights in cost function was decided with considering the driving distance over the prediction horizon which is almost equal to 55.5[m] since the driving speed of ego car is approximately 11.1[m/s]. Other parameters, i.e. weight parameters in cost functions, are decided by cut and try with looking the resulting control performance. The initial positions of the Car 1 and Car 2 are varied in each simulation to test various driving situations. In this paper, five cases shown in Table 4 are tested in this paper.

4.2 Real-Time Computation Capability

At the first of discussion, the time spent for the simulation by proposed method are listed in Table 5. The result is obtained with a desktop computer with Intel Core i7 3.4GHz CPU and 8GB of main memory. Table 5 shows the mean,

Table 4 Settings of driving situations

case	V^E [km/h]	X_0^1 [m]	v_0^1 [km/h]	X_0^2 [m]	v_0^2 [km/h]
(A)	40	150	20	120	20
(B)	40	150	50	120	28.8
(C)	40	150	50	120	43.2
(D)	40	150	50	120	40
(E)	40	150	57.6	120	18

*note that $L_m = 0$ in the case (E)

Table 5 Computation time (millisecond)

N	avg.	1%	99%
100	0.424	0.393	0.589
200	1.57	1.49	2.06
300	3.42	3.26	4.09
400	5.97	5.74	6.56
500	9.19	8.91	9.77
600	13.5	12.9	15.6
800	23.2	22.6	24.7
1000	36.4	35.6	37.7

Fig. 6 Obtained trajectory ($V_1 = 20$ [km/h], $V_2 = 20$ [km/h])

Fig. 8 Obtained trajectory ($V_1 = 43.2$ [km/h], $V_2 = 50$ [km/h])

Fig. 7 Obtained trajectory ($V_1 = 26.7$ [km/h], $V_2 = 50$ [km/h])

Fig. 9 Obtained trajectory ($V_1 = 40.0$ [km/h], $V_2 = 50$ [km/h])

1 percentile and 99 percentile of the computation time used for the optimization problem in one control step. The result shows that the 99 percentile of the computation time with $N = 500$ is less than 10[msec] which is the control interval in this paper. Here the real-time computation ability of the proposed method is confirmed when $N \leq 500$.

4.3 Discussion on Obtained Trajectories

The resulting trajectories by the simulation computed with the proposed method are shown in Fig. 6 to Fig. 9. The horizontal axis of all figures show the longitudinal position and the vertical axes show the lateral positions of cars. In those figures, the Car Es are indicated by a red icon that starts from $x_E = 0$. The time t in the figure shows the time instances of each snapshot. The times, $t^l = 24.14$ in Fig. 6, $t^l = 25.48$ in Fig. 7 and $t^l = 26.93$ in Fig. 8 which are called the time to leave (TTL) are the time instances when the Car E start to leave its driving lane ($y_E > 0.1$). On the other hand, the t^{init} is defined as the time instance when the car changes the reference lateral position in the cost function to initiate the lane-changing, when the Car E passes the point $e = 100$ [m] in other words, for the convenience.

Figure 6 shows the result which the Car E cut-in into ahead of the Car 1 since the Car E drives faster than other cars. The Car E does not leave its driving lane immediately after t^{init} and keeps driving straightly until passing the Car 1. Actual lane-changing is started around $t = 24.14$ [s] since the enough margin is predicted in future between the Car E and the Car 1 by model prediction.

Figure 7 shows the result that the Car E cut-in into the gap between the Car 1 and the Car 2. In this case, the Car E drives faster than Car 2 but slower than Car 1. So the Car E waits for the leaving of Car 1 and waits until the Car E passes Car 2. After the prediction of the both gaps between the Car E and Car 1 and Car E and Car 2 become larger than L_m , the Car E started to leave its driving lane and finished the lane-change smoothly.

Figure 8 shows the another result that the Car E waits for the leaving of other cars. Car E can not leave its driving lane immediately after t^{init} because both of the Car 1 and Car 2 drives close to the Car E. After a while, Car E leaves its driving lane when it is predicted the Car E can follow the Car 2 with keeping more margin to L_m after the accomplishment of lane-changing.

Figure 9 shows the result that the Car E does not

Fig. 10 Validation of safety constraint ($V_1 = 57.6[\text{km/h}]$, $V_2 = 18[\text{km/h}]$, $L_m = 0$)

change its driving lane. The Car E and Car 2 drives same speed with shorter distance than L_m . So the Car E can not keep the safety margin to satisfy the condition to change the parameter of cost function Q . Although the decision was made successfully to realize the safe driving, however, it is necessary to consider not only the steering but also the driving speed as the input to accomplish the lane-changing in this case. This problem will be tackled in our future work.

As a result, the proposed algorithm realize not only the motion control but also the decision of the timing to start the lane-changing by applying the proposed SPPMC method in real-time without two-stage optimization for path planning and motion control. Although this paper focused on verification of the applicability of a non-linear MPC based on C/GMRES method to a SPPMC framework, the performance in computational speed, control accuracy and so on are not compared with other state-of-the-art methods. Those comparison are our next step in addition to the validation with various and realistic driving situation.

4.4 Validity of Safety Constraint

Finally, the validity of safety constraints is tested.

The margin for the safety, L_m should be kept enough long in order to avoid the collision, usually. Here the collision between Car E and Car 2 are tested by setting the safety margin, L_m , to be zero. Figure 10 shows the simulated trajectory with $v_2 = 18[\text{km/h}]$ while $v_E = 40[\text{km/h}]$. Car E starts lane-changing immediately when it passes the starting point $x = 100[\text{m}]$ because the margin L_m is always less than the distance between Car E and Car 2.

It is confirmed that the lateral position y_E is kept less than $2[\text{m}]$ thanks to the safety constraint, $y_E \geq 2$. The ego car completed the lane-changing smoothly after passing Car 2 without any collision. This means the designer of the controller can choose L_m according to the driver's characteristics or preference freer since the implemented constraints can ensure the safety at the critical moment. In addition, needless to say, various types of constraints, such as for the

safety, for the physical limitations and/or for the achievement of the aimed task can be also included as long as those constraints can be handled by C/GMRES method.

5. Conclusion

This paper proposed a simultaneous motion planning and control method realizing a simple decision making in the real-time optimization utilizing a non-linear model predictive control. A lane-change behavior was chosen as the targeting task. Smooth and safe lane change was realized without making a detailed reference path in advance by applying a switched cost function depending on the predicted state in real-time optimization. The non-linear model predictive control (MPC) framework is used to obtain the optimal control input in real-time dealing with these equality and inequality constraints. The validity of the proposed system was checked through the simulation with a realistic vehicle dynamics simulator, and it is confirmed that the suitable decision making of the timing of lane-change and the gap to cut-in besides the smooth and safe path planning and control in the simulation.

Generally speaking, the computational load should get large if the targeting car has more than two surrounding cars, and the applicability of the proposed system in such a case should be checked in future work. The addressed lane-changing task in this paper is also relatively simple case regarding the realistic driving situation. Validation of the proposed method with more realistic driving situation and with the real car are also other future issues to confirm the robustness and usefulness of the proposed method. In addition, the proposed method must be compared with other conventional state-of-the-art methods for path planning and motion control in future work.

References

- [1] L. Li, D. Wen, N. Zheng, and L. Shen, "Cognitive Cars: A New Frontier for ADAS Research," *IEEE Trans. Intell. Transp. Syst.*, vol.13, no.1, pp.395–407, 2012.
- [2] B. Paden, M. Čáp, 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.
- [3] J.-P. Laumond, P.E. Jacobs, M. Taix, and R.M. Murray, "A motion planner for nonholonomic mobile robots," *IEEE Trans. Robot. Autom.*, vol.10, no.5, pp.577–593, 1994.
- [4] L. Han, H. Yashiro, H. Tehrani Nik Nejad, Q.H. Do, and S. Mita, "Bézier curve based path planning for autonomous vehicle in urban environment," *2010 IEEE Intelligent Vehicles Symposium*, pp.1036–1042, 2010.
- [5] M. Abe, *Automotive Vehicle Dynamics*, Tokyo Denki University Press, 2008.
- [6] P. Falcone, F. Borrelli, J. Asgari, H.E. Tseng, and D. Hrovat, "Predictive Active Steering Control for Autonomous Vehicle Systems," *IEEE Trans. Control Syst. Technol.*, vol.15, no.3, pp.566–580, 2007.
- [7] S.A. Arogeti and N. Berman, "Path Following of Autonomous Vehicles in the Presence of Sliding Effects," *IEEE Trans. Veh. Technol.*, vol.61, no.4, pp.1481–1492, 2012.
- [8] Snider and M. Jarrod, "Automatic steering methods for autonomous automobile path tracking," Robotics Institute, Pittsburgh, PA, Tech.

- Rep. CMU-RITR-09-08, 2009.
- [9] A. Koga, H. Okuda, Y. Tazaki, T. Suzuki, B. Levedahl, K. Haraguchi, and Z. Kang, "Autonomous Lane Tracking Reflecting Skilled/Un-skilled Driving Characteristics," 42th Annu. Conf. of the IEEE Industrial Ele. Soc., pp.3175–3180, 2015.
 - [10] J.E. Naranjo, C. Gonzalez, R. Garcia, and T. de Pedro, "Lane-Change Fuzzy Control in Autonomous Vehicles for the Overtaking Maneuver," IEEE Trans. Intell. Transp. Syst., vol.9, no.3, pp.438–450, Sept. 2008.
 - [11] 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," 2015 IEEE International Conference on Vehicular Electronics and Safety (ICVES), pp.69–74, 2015.
 - [12] Y. Rasekhipour, A. Khajepour, S. Chen, and B. Litkouhi, "A Potential Field-Based Model Predictive Path-Planning Controller for Autonomous Road Vehicles," IEEE Trans. Intell. Transp. Syst., vol.18, no.5, pp.1255–1267, May 2017, doi: 10.1109/TITS.2016.2604240.
 - [13] Q. Wang, B. Ayalew, and T. Weiskircher, "Predictive Maneuver Planning for an Autonomous Vehicle in Public Highway Traffic," IEEE Trans. Intell. Transp. Syst., vol.20, no.4, pp.1303–1315, April 2019, doi: 10.1109/TITS.2018.2848472.
 - [14] G. Cesari, G. Schildbach, A. Carvalho, and F. Borrelli, "Scenario Model Predictive Control for Lane Change Assistance and Autonomous Driving on Highways," IEEE Intell. Transp. Syst. Mag., vol.9, no.3, pp.23–35, Fall 2017, doi: 10.1109/ITS.2017.2709782.
 - [15] M. Neunert, C. de Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli, "Fast nonlinear model predictive control for unified trajectory optimization and tracking," Proc. IEEE International Conference on Robotics and Automation (ICRA), 2016.
 - [16] M.A. Gutiérrez-Limón, A. Flores-Tlacuahuac, and I.E. Grossmann, "A reactive optimization strategy for the simultaneous planning, scheduling and control of short-period continuous reactors," Computers & Chemical Engineering, vol.84, pp.507–515, 2016. ISSN 0098-1354.
 - [17] C.V. Caldwell, D.D. Dunlap, and E.G. Collins, "Motion planning for an autonomous underwater vehicle via sampling based model predictive control," IEEE OCEANS, 2010.
 - [18] Y. Rasekhipour, A. Khajepour, S. Chen, and B. Litkouhi, "A Potential Field-Based Model Predictive Path-Planning Controller for Autonomous Road Vehicles," IEEE Trans. Intell. Transp. Syst., vol.18, no.5, pp.1255–1267, May 2017.
 - [19] S. Ulbrich and M. Maurer, "Probabilistic online POMDP decision making for lane changes in fully automated driving," IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013), pp.2063–2067, Oct. 2013.
 - [20] D.D. Salvucci, H.M. Mandalia, N. Kuge, and T. Yamamura, "Lane-Change Detection Using a Computational Driver Model," HUMAN FACTORS, vol.49, no.3, pp.532–542, June 2007.
 - [21] W. Schakel, V. Knoop, and B. Van Arem, "Integrated lane change model with relaxation and synchronization," Transportation Research Record, vol.2316, pp.47–57, 2012.
 - [22] H. Okuda, K. Harada, T. Suzuki, S. Saigo, and S. Inoue, "Design of Automated Merging Control by Minimizing Decision Entropy of Drivers on Main Lane," IEEE Intelligent Vehicles Symposium (IV), 2017.
 - [23] S. Ueda and T. Wada, "Modeling of drivers' decision making of merging space on expressways," Industrial Electronics Society, IECON 2015.
 - [24] M. Keyvan-Ekbatani, V.L. Knoop, and W. Daamen, "Categorization of the lane change decision process on freeways," Transportation Research Part C: Emerging Technologies, vol.69, pp.515–526, 2016.
 - [25] T. Bandyopadhyay, K.S. Won, E. Frazzoli, D. Hsu, W.S. Lee, and D. Rus, "Intention-aware motion planning," International Workshop on the Algorithmic Foundations of Robotics, 2012.
 - [26] Y. Feng, R. Zhang, G. Lie, H. Wang, H. Wen, and J. Xu, "Trajectory planning and tracking control for autonomous lane change maneuver based on the cooperative vehicle infrastructure system," Expert Systems with Applications, vol.42, no.14, pp.5932–5946, 2015.
 - [27] H. Okuda, N. Sugie, and T. Suzuki, "Real-Time Collision Avoidance Control Based on Continuation Method for Nonlinear Model Predictive Control with Safety Constraint," 2017 Asian Control Conf., pp.1086–1091, 2017.
 - [28] J. Jia, A. Khajepour, W. Melek, and Y. Huang, "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints," IEEE Trans. on Vehicular Technology, vol.66, no.2, pp.952–964, IEEE Transactions, 2017.
 - [29] T. Kawabe, H. Nishira, and T. Ohtsuka, "An Optimal Path Generator Using a Receding Horizon Control Scheme for Intelligent Automobiles," Proc. 2004 IEEE Conf. on Cont. App., pp.1597–1602, 2004.
 - [30] M. Otsuka, "On the calculation method of nonlinear Receding Horizon control," J. Society of Instrument and Control Engineers, vol.41, no.5, pp.367–371, 2002 (in Japanese).
 - [31] 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, pp.286–293, 2002.
 - [32] 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, pp.418–424, 2002 (in Japanese).
 - [33] T. Otsuka, "A continuation/GMRES method for fast computation of nonlinear receding horizon control," Automatica, vol.40, no.4, pp.563–574, 2004.

Hiroyuki Okuda was born in Gifu, Japan, in 1982. He received B.E. and M.E. degrees in Advanced Science and Technology from Toyota Technological Institute, Japan in 2005 and 2007, respectively, and he received a Ph.D. degree in Mechanical Science and Engineering from Nagoya University, Japan in 2010. He was a PD researcher with the CREST, JST from 2010 to 2012, and an assistant professor at Nagoya University's Green Mobility Collaborative Research Center from 2012 to 2016. He was a visiting researcher in the Mechanical Engineering Department of U. C. Berkeley in 2018. Currently, he is an assistant professor in the Department of Mechanical Science and Engineering at Nagoya University. His research interests are in the areas of system identification of hybrid dynamical system and its application to the modeling and analysis of human behavior and human-centered system design of autonomous/human-machine cooperative system. Dr. Okuda is a member of the IEEE, IEEEJ, SICE, and JSME.

Nobuto Sugie was born in Aichi, JAPAN, in 1995. He received the B.S. degree in Electronic Mechanical Engineering from Nagoya University, JAPAN in 2017. His research interests are in the areas of modeling driver decision characteristics and its application to vehicle control.

Tatsuya Suzuki was born in Aichi, JAPAN, in 1964. He received B.S., M.S., and Ph.D. degrees in Electronic Mechanical Engineering from Nagoya University, Japan in 1986, 1988, and 1991, respectively. From 1998 to 1999, he was a visiting researcher in the Mechanical Engineering Department of U. C. Berkeley. Currently, he is a Professor in the Department of Mechanical Systems Engineering and Executive Director of Global Research Institute for Mobility in Society, Nagoya University. He is also a

Principal Investigator in JST, CREST. He won the best paper award at the International Conference on Autonomic and Autonomous Systems 2017 and the outstanding paper award from the International Conference on Control Automation and Systems 2008. He also won the journal paper award from IEEJ, SICE, and JSAE in 1995, 2009, and 2010, respectively. His current research interests are in the areas of analysis and design of human-centric mobility systems and integrated design of transportation and energy management systems. Dr. Suzuki is a member of the IEICE, SICE, ISCIE, JSAE, RSJ, JSME, IEEJ, and IEEE.

Kentaro Haraguchi was born in Aichi, JAPAN, in 1981. He received the B.S., M.S. and Ph.D. degrees in Science from Nagoya University, JAPAN in 2004, 2006 and 2011, respectively. He joined Toyota Technical Development Corporation in 2011, and engaged in control development of a robotic auto-operating system. His research interests include modeling and simulation of human behavior.

Zibo Kang received the B.S. and Ph.D. degrees in Industrial Automation Engineering from Northeastern University, China, in 1986 and 1996, respectively. From 1993 to 1995, he was a graduate research student in Nagoya University. He is currently a senior general manager in Toyota Technical Development Corporation (TTDC), Japan. Before joining TTDC, he was an assistant professor at Northeastern University of China, then a visiting researcher at Daido University, Japan. He won the Technical Award

from the SICE in 2009. His research interests include the modeling and simulation of vehicles, human behavior and transportation environment, and system verification & validation. Dr. Kang is a member of the SICE and JSAE.