Trajectory Generation Using Model Predictive Control for Automated Vehicles

-Maximize Usage of Road Width Like A Skilled Human Driver-

Yoshiaki Irie ¹⁾ Daisuke Akasaka ²⁾

*1) Toyota Motor Corporation, Automated Driving & Advanced Safety System Development Div.
2-3-18, Kudanminami, Chiyoda-ku, Tokyo, 102-0074, Japan (E-mail: yoshiaki_irie@mail.toyota.co.jp)*

*2) MathWorks Japan
7/F Akasaka Garden City, 4-15-1 Akasaka, Minato-ku, Tokyo, 107-0052, Japan*

Received on November 9 ,2020

ABSTRACT: Recently, many companies have been working on developing technologies for automated driving. For automotive companies, one of the challenges is to realize the automated driving with safety and reliability. This paper addresses a trajectory generation method using model predictive control (MPC) for the control development of the automated driving. Using the MPC, the design method of a steering control will be proposed in order to generate trajectory and develop a practical algorithm considering a product implementation. More concretely, we propose the design method of 1) online trajectory generation to maximize use of an entire road width like a skilled driver that has a significant experience in driving and 2) smooth trajectory correction whenever the vehicle slips away from an appropriate trajectory, which are useful for actual product development.

KEY WORDS: vehicle dynamics, autonomous driving system / electronic stability control, model predictive control, automated vehicle, vehicle trajectory control, path planning, driving model [B1]

1. Introduction

Recently, many companies have been working on developing new technologies for ADAS (Advanced Driver Assistance System) and automated vehicle. Many ADAS vehicles have already launched in the global market today. For both of the ADAS and the automated vehicle, one of the technological challenges is to achieve an excellent driving performance like a skilled driver that has a significant experience beyond a machine. In other words, this indicates that passenger comfort is improved with a natural driving.

In the current ADAS, the center line of a road is often set as a desired trajectory which is tracked by using some feedback control algorithm such as LQR (Linear Quadratic Regulator). However, the center line is not necessarily a good trajectory to reduce a lateral acceleration or jerk that directly effects riding comfort on curved roads, that is, strictly following the center line may expose passengers to discomfort or even stress. It is important to find a good trajectory in an entire road width to increase the driving performance.

The objective of this research is to develop a new practical method of a trajectory generation using the model predictive control (MPC) for automated vehicle to maximize use of a road width. In this paper, we propose an affordable and practical approach by taking into consideration a product implementation. A lateral control, i.e., steering control is focused on by separating a longitudinal control and a lateral control independently for practicality and simplicity. We can avoid a high-spec CPU because the number of control parameters can be reduced and the complex nonlinearity of a vehicle dynamics model can be avoided for the

MPC. The steering control command is calculated with a fixed longitudinal velocity at each time step.

This paper proposes the design method of 1) online trajectory generation within an entire road width, i.e., not necessarily follow a center line and 2) smooth trajectory correction whenever the vehicle slips away from an appropriate trajectory due to unexpected disturbance or environmental change. The result of numerical simulations will show that the MPC has a capability to perform like a skilled driver with high control performance. In general, the skilled drivers look far ahead while driving, and hence it is expected that we can generate the good trajectory using the look-ahead information such as a road alignment. The MPC can calculate an optimal manipulation using the look-ahead information in a prediction interval.

LQR is a popular feedback control method for vehicle dynamics control. However, the MPC combines the advantage of feedforward and feedback control. The feedforward part is updated at every control interval by a prediction using a dynamical model and preview information such as road alignment. In addition, the MPC can explicitly deal with signal constraints more easily than the LQR. Therefore, the MPC is expected to maximize the control performance compared with the LQR.

Many trajectory planning and control algorithms have been actively researched for autonomous driving [1]. MPC-based control approaches [2-5] have been proposed in lateral control applications such as a lane keeping, an obstacle avoidance and a path following. Recently, the MPC has also been become a popular approach for path planning or trajectory generation [6-9]. The point of this paper is that the algorithm and calibration are discussed from

the perspective of 1. increasing the driving performance by making full use of the road width and 2. product merchandising.

The paper is organized as follows. Section 2 describes vehicle dynamics model for both of the simulation and prediction of the MPC. Section 3 introduces the MPC and sets up an optimization problem for the lateral control. Section 4 to 5 discusses how to simplify a cost function and design the MPC parameters via numerical simulations. Finally, Section 6 concludes with a brief summary.

2. Vehicle System

2.1. Vehicle Dynamics Model

A vehicle dynamics is simply described by an equivalent bicycle model which can be represented by a linear state-space equation as below:

$$\dot{x} = Ax + Bu \quad (1)$$

where $x \in \mathbb{R}^5$ and $u \in \mathbb{R}$ are the state and the control input respectively. x is defined as $x = [\dot{\theta} \ \theta \ \dot{y} \ y \ \delta]^T$ and u is the steering angle command. A and B matrices are defined as follows. The meaning of each notation is described in Appendix A.1.

$$A = \begin{bmatrix} -\frac{2l_f^2 C_f + 2l_r^2 C_r}{IV} & 0 & -\frac{2l_f C_f - 2l_r C_r}{IV} & 0 & \frac{2l_f C_f}{IN} \\ 1 & 0 & 0 & 0 & 0 \\ -\frac{2l_f C_f - 2l_r C_r}{mV} - V & 0 & -\frac{2C_f + 2C_r}{mV} & 0 & \frac{2C_f}{mN} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{T} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{T} \end{bmatrix}^T$$

2.2. System Hardware

The hardware configuration of a vehicle system under consideration is based on general ADAS and automated vehicle systems. As shown in Fig.1, we assume that a camera and a positioning system with GNSS and map-matching are equipped. The vehicle system has an inertial measurement unit (IMU) with which a yaw rate of the vehicle can be detected. In addition, an electric power steering (EPS) system is mounted to control the vehicle behavior.

To preview the road alignment, the camera is used to accurately get a view closer along the road. The map is used to roughly get a view far ahead of the road. This is a minimum hardware configuration for the automated vehicle system with the proposed algorithm of this paper. Other devices may also be equipped to increase sensing accuracy and reliability.

Fig. 1 System hardware configuration.

3. Model Predictive Control

Model predictive control (MPC) [10] is a feedback control method that predicts and optimizes the future plant response at each control interval using a dynamical model of the plant. The MPC solves an optimization problem online and can explicitly handle signal constraints due to physical or safety limitation. Moreover, the MPC can deal with multivariable plants in a similar manner. Therefore, the MPC has attracted a lot of attentions from a wide range of industries to achieve high control system performance for complex systems.

With the MPC, the human-like trajectory generation problem must be formulated as an optimization problem, i.e., a cost function with inequality constraints. The cost function means a required control performance to be minimized. Reducing a lateral acceleration/jerk and a steering angular rate should be considered in the cost function. The vehicle is subject to some physical constraints such as road width and steering angle limitation.

3.1. Prediction Model

A prediction model is an important part of the MPC to optimize the future vehicle behavior. To create the prediction model, we assume that a vehicle longitudinal velocity is constant during prediction range. This is because the model needs to be simplified to avoid increasing the computational cost of numerical optimization. Here, a road center line is considered as a base line for the coordination of the prediction model. More precisely, as shown in Fig.2, the state of the prediction model is defined as a deviation from the road center line [11]. However, note that the road center line is not necessarily a desired vehicle trajectory.

Fig. 2 Coordination of prediction model.

Let x_e be the state of the prediction model defined as

$$x_e := \begin{bmatrix} \dot{\theta}_e \\ \theta_e \\ \dot{y}_e \\ y_e \\ \delta \\ z \end{bmatrix} = \begin{bmatrix} \dot{\theta} - \dot{\theta}_{ref} \\ \theta - \theta_{ref} \\ \dot{y} + V\theta_e \\ y - y_{ref} \\ \delta \\ \int y_e \end{bmatrix} \quad (2)$$

in which $\dot{\theta}_{ref}$ is the yaw rate reference and θ_{ref} is the heading angle reference based on the road center line. The current and future information of $\dot{\theta}_{ref}$ and θ_{ref} can be obtained from the camera and/or the map. In (2), $\dot{\theta}_e$, θ_e , $\dot{y}_e$ and y_e mean the deviation from the road center line. z is a newly added state that is the time integration of y_e to increase robustness against disturbance.

By substituting the new state (2) into (1), we can derive the state space model below (See Appendix A.3 for further details). For simplicity, we assumed that $\dot{\theta}_{ref} = 0$ to avoid a possible noisy input and reduce the computational cost.

$$\begin{cases} \dot{x}_e = A_e x_e + B_e u + G_e \dot{\theta}_{ref} \\ y_m = C_e x_e + H_e \dot{\theta}_{ref} \end{cases} \quad (3)$$

where $y_m \in \mathbb{R}^7$ denotes the measurement output which consists of the state x_e and the yaw rate $\dot{\theta}$, i.e.,

$$y_m = [x_e^T \dot{\theta}]^T = [\dot{\theta}_e \ \theta_e \ \dot{y}_e \ y_e \ \delta \ z \ \dot{\theta}]^T. \quad (4)$$

We can directly measure $\dot{\theta}$ using the vehicle equipped yaw rate sensor. $\dot{\theta}$ will be used in the cost function to reduce the lateral motion change. Each coefficient matrices in (3) are defined as follows:

$$A_e = \begin{bmatrix} \frac{2l_f^2 C_f + 2l_r^2 C_r}{IV} & \frac{2l_f C_f - 2l_r C_r}{I} & -\frac{2l_f C_f - 2l_r C_r}{IV} & 0 & \frac{2l_f C_f}{IN} & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{2l_f C_f - 2l_r C_r}{mV} & \frac{2C_f + 2C_r}{m} & -\frac{2C_f + 2C_r}{mV} & 0 & \frac{2C_f}{mN} & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{T} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$B_e = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{T} & 0 \end{bmatrix}^T$$

$$G_e = \begin{bmatrix} -\frac{2l_f^2 C_f + 2l_r^2 C_r}{IV} & 0 & -\frac{2l_f C_f - 2l_r C_r}{mV} - V & 0 & 0 & 0 \end{bmatrix}^T$$

$$C_e = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_e = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]^T$$

3.2. Optimization Problem

The steering angle command u is calculated at each control interval by solving the optimization problem formulated as below:

$$\min_u \int_t^{t+T_p} (\sum_{i=1}^7 q_i y_{mi}^2 + r_{\dot{u}} \dot{u}^2 + r_u u^2) d\tau \quad (5)$$

s.t.

$$\begin{aligned} y_{emin} &\leq y_e \leq y_{emax} \\ u_{min} &\leq u \leq u_{max} \\ \dot{u}_{min} &\leq \dot{u} \leq \dot{u}_{max} \end{aligned}$$

where T_p is the prediction horizon of time range, q_i ($i = 1, \dots, 7$), $r_{\dot{u}}$ and r_u are the non-negative scalar weight for each measurement

outputs (error), the change rate and magnitude of the steering angle command, respectively. The suffix “min” and “max” in the inequality constraints denotes lower bound and upper bound. The inequality of y_e means the road width constraint. The control input constraints limit the steering angle and angular velocity.

The linear MPC optimization problem above can be converted to a discrete-time form and thus results in Quadratic Programming (QP) problem. We can solve the problem in real-time using fast QP solvers. Once an optimal solution $u^* = [u_t^* \ u_{t+1}^* \ \dots \ u_{t+p-1}^*]^T$ along the prediction horizon is obtained at each control interval, the first element u_t^* of the sequence is only used as an actual control input.

Using the obtained steering angle command, the EPS system works to realize the desired lateral behavior. The feedback control is performed to measure the current state x_e at each time step by using the camera and the yaw rate sensor.

Fig. 3 Control system with MPC.

Fig. 4 Optimization problem with preview information.

4. Performance Design

The basic principle of our performance design are two-folds:

1. Realize high driving performance like a skilled driver.
2. Generate a trajectory making use of the full width of a road, i.e., not necessarily follow a center line

Based on the above, firstly the cost function in (5) will be simplified. Secondly, we will propose how to design the controller parameters including the weight and the prediction horizon to meet desired performance requirements.

4.1. Redesigned Cost Function

To simplify the cost function, we need to understand the role of each term in (5) for the vehicle behavior. The role will be explained as follows:

For the ease of explanation, the cost function in (5) is described again below.

$$\int_t^{t+T_p} (q_1 \dot{\theta}_e^2 + q_2 \theta_e^2 + q_3 \dot{y}_e^2 + q_4 y_e^2 + q_5 \delta^2 + q_6 z^2 + q_7 \dot{\theta}^2 + r_u \dot{u}^2 + r_u u^2) d\tau$$

- The yaw rate deviation term $q_1 \dot{\theta}_e^2$ has an effect to move the vehicle closer to the center line while turning on curved roads, i.e., correct the lateral position toward the center line.
- The heading angle deviation term $q_2 \theta_e^2$ has an effect to align the vehicle heading to the center line. This means that the time integral of $\dot{\theta}_e$ is regulated.
- The lateral velocity deviation term $q_3 \dot{y}_e^2$ has an effect to reduce a sudden change of the lateral behavior. The lateral displacement term $q_4 y_e^2$ moves the vehicle lateral position closer to the center line on both curved and straight roads.
- The steering related terms $q_5 \delta^2$, $r_u \dot{u}^2$ and $r_u u^2$ penalize a steering motion and hence increase the smoothness of the vehicle behavior.
- The time integral state term $q_6 z^2$ is expected to increase the robustness against disturbances such as lateral wind and road cant. The paper focuses on a nominal operation without the disturbance. However, this is an important part for the performance design in actual product development.
- The yaw rate $\dot{\theta}$ does not appear in the state of the prediction model (2). However, $\dot{\theta}$ is newly added to the measurement output (4). $q_7 \dot{\theta}^2$ has an effect to penalize a sudden change of the yaw motion. This contributes to reducing the lateral acceleration.

Table 1 The role of each term in the cost function.

Role	Symbol	Effect
Control yaw behavior	$\dot{\theta}_e$	Moving the vehicle closer to the center line while turning driving
	θ_e	Aligning the vehicle heading to the center line
	$\dot{\theta}$	Suppressing the generation of the yaw rate, which contributing to reducing the lateral acceleration
Control lateral displacement	$\dot{y}_e$	Suppressing the sudden change of the lateral displacement
	y_e	Moving the vehicle closer to the center line while turning and straight driving
Smoothness	δ	Increasing the smoothness of the vehicle behavior
Disturbance rejection	z	Suppressing the effect of the external disturbance
Preview range	T_p	Optimizing the cost function during the time range

In what follows, the cost function will be simplified. It is desirable to generate a trajectory with a smaller lateral acceleration in order to increase riding comfort. Since the longitudinal velocity is assumed to be constant in the vehicle model, reducing the yaw rate is almost equivalent to reducing the lateral acceleration. Furthermore, in general, the yaw rate can be measured more accurately than the lateral acceleration. Therefore, the yaw rate $\dot{\theta}$ is used in the cost function. To avoid many calibration parameters, several terms in (5) are deleted and substituted by other terms with the similar effect, e.g., $q_4 y_e^2$ can be substituted by $q_1 \dot{\theta}_e^2$ in that the vehicle is moved closer to the center line while turning. For the smooth behavior, we will adopt the steering angle rate $\dot{u}$, not the

steering angle δ and u , to focus on avoiding the sudden change of the steering motion like a skilled driver manipulation.

As a result, the cost function is redefined as below:

$$\min_u \int_t^{t+T_p} (w_1 \dot{\theta}_e^2 + w_2 \dot{\theta}^2 + r \dot{u}^2) d\tau \quad (6)$$

s.t.

$$\begin{aligned} y_{emin} &\leq y_e \leq y_{emax} \\ \dot{u}_{min} &\leq \dot{u} \leq \dot{u}_{max} \end{aligned}$$

where the weight w_i ($i = 1, 2$) are redefined instead of q_i .

The parameters in the optimization are narrowed down to the yaw rate deviation, the yaw rate, and steering angle command rate. The limit of the road width and the steering angle command rate are considered as the constraints.

5. Parameter Design

In this section, we will show how to design the MPC parameters, that is the weight and the prediction horizon in (6) through the numerical simulation of a closed-loop system.

5.1. Simulation Conditions

In the simulation, the vehicle specification is set up as in Appendix A.2. The longitudinal velocity V is 60km/h ($\cong 16.7$ m/s) and the control interval T_s is 0.1sec. The prediction horizon step is set to 35 step which means the prediction time range T_p is 3.5sec, i.e., the prediction distance is approximately 58.3m.

MATLAB®, Simulink® and Model Predictive Control Toolbox™ [12] are used for the closed-loop simulation. The QP problem (6) is solved by an active-set solver called the KWIK algorithm based on [13]. In a Simulink model, the controller is modeled using “MPC Controller” block in the toolbox. The block can deal with a linear time-invariant system as a prediction model. The Simulink model is composed of the vehicle dynamics, the MPC controller, the calculation of the measurement output y_m and the road center line data. If the vehicle velocity V is variable, the state-space model (3) becomes a linear parameter varying (LPV) system. In such a case, “Adaptive MPC Controller” block can be used in the Simulink model. The block supports the linear prediction model with parameter change at each control interval.

5.2. Weight of Cost Function

Fig.5 shows the generated trajectories by two different combination of the weight on a L-shaped curved road. Case 1 focuses on reducing the yaw rate deviation $\dot{\theta}_e$ with a larger w_1 and a smaller w_2 . Case 2 focuses on reducing the yaw rate $\dot{\theta}$ with a smaller w_1 and a larger w_2 . Each value of the weight is listed in Table 2. Fig.6 shows the vehicle lateral behaviors with respect to the horizontal axis of the road distance. In Fig.6, from the above, the lateral position deviation, the curvature of the road, the lateral acceleration and the lateral jerk are depicted.

In theory, an optimal trajectory follows a geometric line, which is the largest possible turning radius through a corner. For example, the apex is often chosen as the geometric center of the corner. The apex is the innermost point of the path taken through a curve. The vehicle enters from the outside edge, touches the apex, and then runs to the outside edge of the road on the exit.

As seen in the figures, Case 1 shows that the vehicle moves in closer to the center line while turning. Meanwhile, Case 2 takes near the geometric line. The lateral jerk is clearly reduced in the

corner entry and exit compared with Case 1. This result indicates that by reducing the yaw rate $\dot{\theta}$, the MPC can generate a trajectory with less jerk making use of the entire road width like a skilled driver. We can choose the characteristics of the vehicle trajectory by adjusting w_1 and w_2 . In practice, we should use not only w_2 , but also w_1 to move the vehicle to the inside of the road for robustness and reliability.

Fig.7, 8 demonstrates the effect of the weight of r in (6). As in Table 2, Case 3 takes a larger weight ($r = 100$) compared with Case 4 ($r = 0$). The simulation result shows that Case 3 results in increasing the smoothness of the lateral acceleration and jerk. In practice, the weight value needs to be selected carefully, considering the actuator specification how it quickly responds to the steering angle command.

Table 2 Weight values of Case 1~4.

Case	w_1	w_2	r
1	100	0	100
2	0	100	100
3	0	100	100
4	0	100	0

Fig. 5 Case 1 vs Case 2: Generated trajectories.

Fig. 6 Case 1 vs Case 2: Lateral behaviors.

Fig. 7 Case 3 vs Case 4: Generated trajectories.

Fig. 8 Case 3 vs Case 4: Lateral behaviors.

It is concluded that the weight parameters are designed based on the following direction:

- Generate a trajectory using the weight w_2 to reduce the yaw rate $\dot{\theta}$ contributes to reducing the lateral acceleration and jerk.
- Increase the smoothness of the vehicle behavior using the weight r for the steering angle rate.

Note that this simplified direction of the parameter choice is derived from the simulation with minimum basic parameters. However, actually, it is necessary to sophisticate the direction through testing cycle between a simulation test and an actual vehicle test.

5.2. Prediction Horizon

Prediction horizon is an important MPC parameter which represents a predicted time range from a current time. The prediction horizon should be carefully selected because it affects a control system performance and a computational cost to solve an optimization problem.

Since the prediction horizon affects the driving performance with the generated trajectory, it is important to find a direction how to calibrate the prediction horizon. It is desirable that we have a calibration concept to makes it easier for us to find an appropriate

horizon efficiently. This paper proposes an example of the calibration concept as shown in Fig.9.

Fig. 9 An example of prediction horizon setting concept.

In Fig.9, the arrow at the top (from left to right) indicates the preview distance from the current vehicle position. The step number means the prediction horizon step. In this case, the prediction horizon step depends on the vehicle velocity 60km/h ($\approx 16.7\text{m/s}$) and the control interval 0.1sec. In general, the camera is available from several 10m to 100m at a maximum. The map data is used outside the camera's available range.

Fig. 10 Case 5~7: Trajectory difference by prediction horizon

Fig.10 shows that the difference of the generated trajectory for three prediction horizons 15, 35 and 50 steps, i.e., the prediction distances are approximately 25.0, 58.3 and 83.3m. In Case 5, for the shorter horizon, i.e., the number of step is 15, the MPC recognizes the road ahead as almost a straight line at each time step. Therefore, the MPC is delayed to recognize the curve ahead and then operate the steering action. As a result, it is difficult to obtain the desired trajectory with comfortable driving. On the other hand, in Case 6 and 7, for the longer horizon, we can see almost the similar trajectory between 35 and 50 steps. In this case, 35 steps is suitable provided that the computational cost is lower. Furthermore, the long prediction distance may include multiple curvatures on the road ahead. In such case, the generated trajectory near ahead in front of the vehicle may not be suitable depending on the road alignment due to far ahead of the prediction being possibly optimized intensively. Therefore, it is necessary to determine an appropriate length of the prediction horizon. We need to find a well-balanced horizon length among driving environment, system operating range, target control performance, camera capability of measuring a distance and CPU specification.

5.3. Effect of Variable Velocity Driving on MPC with Constant Velocity Prediction

In the above discussion, we assumed that the vehicle velocity is constant in the prediction model to simplify the algorithm to able to be capable of productization. However, the velocity is naturally changed during driving. In this section, we will discuss the effect of the variable velocity on the trajectory generation using the proposed algorithm.

In a general driving practice, it is known that the velocity should be adjusted as “slow-in, fast-out”, which means that the cornering velocity is reduced before apex and then increased after apex while turning. A travel time needs to be minimized to obtain a fastest travel time trajectory. However, the velocity is assumed to be constant and the travel time cannot be easily evaluated in the cost function. Hence, it does not guarantee that the generated trajectory corresponds to the fastest travel time trajectory.

In the simulation, a measured variable velocity is entered into the MPC at each sampling time. Then, the MPC predicts a future vehicle behavior with a fixed velocity at the sampling time. Fig.11 shows three trajectories with different given velocities in the L-shaped curved road. The top graph in Fig.12 describes how the velocity is changed. All cases are set as the same entry velocity 60 km/h at the entry point P1 of the corner. In Case 8, the vehicle run at the fixed 60 km/h. In Case 9 and 10, the velocity is linearly decreased/increased from P1 to the clip point on the inside P2. Then, the vehicle accelerates/decelerates to 60 km/h again toward the exit P3. Note that the purpose of this subsection is to evaluate the effect of the variable velocity and therefore the strictness of the clip point position is not discussed here.

From the figures, all three cases follow almost the same trajectory from P1 to the clip point even though the velocity is changed while turning. In this example, the velocity at P1 is set to 60m/h in all cases. This mainly determines the trajectory at the entry of the corner to the clip point. The velocity change while turning has a small effect on the trajectory.

Fig. 11 Case 8~10: Generated trajectories by variable velocity.

Fig. 12 Case 8~10: Lateral behaviors by variable velocity.

Next, let us discuss the trajectory from P1 to P2. As shown in Fig.11 and 12, the trajectories are clearly different among each case because the prediction distances are different at P2. For example, in Case 9, the trajectory goes to the outside from P2 at an early timing compared with other cases. Since the prediction distance is short, the MPC tries to minimize the yaw rate and obtain a bigger turning radius. We can conclude that the prediction horizon dominantly affects the generated trajectory.

For racetrack driving, the velocity should be a design variable to minimize a travel time using a tire friction circle efficiently for a fastest driving trajectory. As mentioned above, we assume the velocity is constant and we cannot easily solve the travel time minimization problem. On the other hand, for automated vehicles for general consumer, its operation range is within low acceleration on normal roads. If the travel time needs to be reduced more, a nonlinear optimization algorithm and a high-spec CPU may be required.

At the moment, as an algorithm for commercialization, the proposed algorithm with constant velocity would provide a good benefit by cost from the viewpoint of system performance, cost and reliability.

On the other hand, we should pursue more sophisticated driving so as a future possibility. The vehicle lateral position can be controlled by the longitudinal velocity in addition to the steering action. In general, in a high velocity range, a turning radius can be adjusted more easily by the velocity than the steering. Therefore, in addition to the steering angle, the manipulation of an accelerator and a brake is effective for a precise trajectory generation. If a fast and reliable nonlinear MPC solver is available in the future, it is desirable to use it to optimize both steering and velocity at the same time.

5.4. Simulation of Multiple Curved Road Scenario

As above, we discussed how to determine the MPC parameters through the simple L-shaped curved road. We will simulate how it works for a multiple curved road scenario. Fig.13 shows the simulation result for a multiple curved road in Case 11 and 12, in which we used the same MPC parameter set as Case 1 and 2 in Table 2, respectively. the MPC generates a trajectory to reduce the yaw rate by previewing upcoming corner in driving. The trajectory of Case 12 reduces the yaw rate more than Case 11 by appropriate parameter in achieving our objectives. As a result, the trajectory of Case 12 is more linearly generated at the corner P4 and P5.

From the above, the MPC parameters are set to $w_1 = 0$, $w_2 = 100$, $r = 100$ and $T_p = 3.5$ (=35steps). This example assumes the operating range is up to 0.3G. It is concluded that the prediction horizon is reasonable between 35 and 50 steps. For example, if you consider a turn with 0.1G on a curved road, a longer prediction horizon is required, however such slow turn is regarded as almost a straight line and there is no problem to track the center line for riding comfort. After all, it is required to set parameter values considering how to use the system.

Fig. 13 Case 11 vs Case 12: Generated trajectories.

6. Conclusion

In vehicle tracking control, the vehicle is usually controlled to track the reference trajectory. The objective of this paper is to automatically drive like a skilled-driver and generate the trajectory within an entire road width. To this end, we proposed a design approach with the MPC which can generate feedforward trajectory using preview information and a dynamical model and derive an optimal solution online by solving an optimization problem with constraints in real-time. This approach has the potential for the system to judge and drive like human.

This paper showed that the MPC is able to generate the human-like trajectory with the constraint of the road width by minimizing the yaw rate. Also, we discussed how to simply design the cost function and the parameters including the weight and the prediction horizon.

By the simulation, we can also analyze the required specifications of the system hardware, the system performance and the robustness against variation and uncertainty. The required hardware can be specified by clarifying a target performance and designing the MPC parameters appropriately.

In this paper, we considered an ideal plant model. However, for example, the actual actuator cannot work as commanded due to a deadzone. For commercialization, the controller must be designed well-balanced according to the target performance, the usecase and the vehicle characteristics, taking a robustness and a software implementation into consideration. It is necessary to utilize the lateral deviation y_e and its time integration z in (5) to deal with external disturbance or variation in sensors and determine a design principle by repeating testing between simulation and actual on-road test.

References

- (1) 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 Transactions on Intelligent Vehicles, vol. 1, no. 1, pp. 33-55 (2016)

- (2) P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng and D. Hrovat : Predictive Active Steering Control for Autonomous Vehicle Systems, IEEE Transactions on Control Systems Technology, vol. 15, no. 3, pp. 566-580 (2007).
- (3) J. Turri, A. Carvalho, H. E. Tseng, K. H. Johansson and F. Borrelli : Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads, 16th International IEEE Conference on Intelligent Transportation Systems, pp. 378-383 (2013).
- (4) A. Liniger, A. Domahidi, and M. Morari : Optimization-based autonomous racing of 1:43 scale RC cars, Optimal Control Applications and Methods, vol. 36, no. 5, pp. 628-647 (2015).
- (5) E. Kim, J. Kim, and M. Sunwoo : Model Predictive Control Strategy for Smooth Path Tracking of Autonomous Vehicles with Steering Actuator Dynamics, International Journal of Automotive Technology, vol. 15, pp.1155-1164 (2014).
- (6) B. Yi, P. Bender, F. Bonarens and C. Stiller : Model Predictive Trajectory Planning for Automated Driving, IEEE Transactions on Intelligent Vehicles, vol. 4, no. 1, pp. 24-38 (2019).
- (7) C. Liu, S. Lee, S. Varnhagen and H. E. Tseng : Path planning for Autonomous Vehicles Using Model Predictive Control, 2017 IEEE Intelligent Vehicles Symposium (IV), pp.174-179 (2017).
- (8) Y. Rasekhipour, A. Khajepour, S. Chen and B. Litkouh : A Potential Field-Based Model Predictive Path-Planning Controller for Autonomous Road Vehicles, IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 5, pp. 1255-1267 (2017).
- (9) M. Nolte, M. Rose, T. Stolte and M. Maurer : Model Predictive Control Based Trajectory Generation for Autonomous Vehicles - An Architectural Approach, 2017 IEEE Intelligent Vehicles Symposium (IV), pp. 798-805 (2017).
- (10) C. E. Garcia, D. M. Prett, and M. Morari : Model Predictive Control: Theory and Practice - a Survey, Automatica, vol. 25, no. 3 pp. 335-348 (1989).
- (11) R. Rajamani : Vehicle Dynamics and Control, Springer (2006).
- (12) MathWorks : Model Predictive Control Toolbox Documentation, <https://www.mathworks.com/help/mpc/index.html>.
- (13) C. Schmid and L.T. Biegler : Quadratic Programming Methods for Reduced Hessian SQP, Computers & Chemical Engineering 18, no. 9, pp.817-832 (1994).

θ : Heading angle [rad]
 $\dot{y}$: Lateral velocity [m/s]
 y : Lateral position [m]
 δ : Actual steering angle [rad]
 l_f : CG-front wheel distance [m]
 l_r : CG-rear wheel distance [m]
 C_f : Front tire cornering stiffness [N/rad]
 C_r : Rear tire cornering stiffness [N/rad]
 I : Yaw inertia [kg/m²]
 m : Total vehicle mass [kg]
 V : Vehicle speed [m/s]
 T : Time constant of steering actuation [sec]
 N : Overall gear ratio [-]

A.2. Vehicle Specification

Parameter	Value
l_f	1.49
l_r	1.48
C_f	1.4×10^5
C_r	2.3×10^5
I	5.0×10^3
m	2.6×10^3
T	0.3
N	20

A.3. Derivation of the prediction model (3)

In this section, we will derive the dynamics of the error state θ_e and $\dot{y}_e$ from (1). The states $\theta_e = \theta - \theta_{ref}$ and $\dot{y}_e = \dot{y} + V\theta_e$ are substituted into the yaw rate equation below in (1).

$$\begin{aligned}\ddot{\theta} &= -\frac{2l_f^2 C_f + 2l_r^2 C_r}{IV} \ddot{\theta} - \frac{2l_f C_f - 2l_r C_r}{IV} \dot{y} + \frac{2l_f C_f}{IN} \delta \\ \Leftrightarrow \ddot{\theta}_e + \ddot{\theta}_{ref} &= -\frac{2l_f^2 C_f + 2l_r^2 C_r}{IV} (\ddot{\theta}_e + \ddot{\theta}_{ref}) - \frac{2l_f C_f - 2l_r C_r}{IV} (\dot{y}_e - V\theta_e) \\ &\quad + \frac{2l_f C_f}{IN} \delta \\ \Leftrightarrow \ddot{\theta}_e &= -\frac{2l_f^2 C_f + 2l_r^2 C_r}{IV} \ddot{\theta}_e + \frac{2l_f C_f - 2l_r C_r}{I} \theta_e - \frac{2l_f C_f - 2l_r C_r}{IV} \dot{y}_e \\ &\quad + \frac{2l_f C_f}{IN} \delta - \frac{2l_f^2 C_f + 2l_r^2 C_r}{IV} \dot{\theta}_{ref} - \ddot{\theta}_{ref}\end{aligned}$$

In this paper, for simplicity, we assume that $\ddot{\theta}_{ref} = 0$ to avoid possible noisy input and reduce the computational cost. As the same way, $\theta_e = \theta - \theta_{ref}$ and $\dot{y}_e = \dot{y} + V\theta_e$ are substituted into the lateral velocity equation below in (1).

$$\begin{aligned}\dot{y} &= \left(-\frac{2l_f C_f - 2l_r C_r}{mV} - V\right) \dot{\theta} - \frac{2C_f + 2C_r}{mV} \dot{y} + \frac{2C_f}{mN} \delta \\ \Leftrightarrow \dot{y}_e - \dot{V}\theta_e - V\dot{\theta}_e &= \left(-\frac{2l_f C_f - 2l_r C_r}{mV} - V\right) (\dot{\theta}_e + \dot{\theta}_{ref}) \\ &\quad - \frac{2C_f + 2C_r}{mV} (\dot{y}_e - V\theta_e) + \frac{2C_f}{mN} \delta \\ \Leftrightarrow \dot{y}_e &= -\frac{2l_f C_f - 2l_r C_r}{mV} \dot{\theta}_e + \frac{2C_f + 2C_r}{m} \theta_e - \frac{2C_f + 2C_r}{mV} \dot{y}_e + \frac{2C_f}{mN} \delta \\ &\quad + \left(-\frac{2l_f C_f - 2l_r C_r}{mV} - V\right) \dot{\theta}_{ref}\end{aligned}$$

In the above, V is assumed to be constant and therefore $\dot{V} = 0$.

Appendix

A.1. Notations

$\dot{\theta}$: Yaw rate [rad/s]