

Model Predictive Control Approach to Design Practical Adaptive Cruise Control for Traffic Jam

Taku TAKAHAMA ¹⁾ Daisuke AKASAKA ²⁾

1) Hitachi Automotive Systems, Ltd.

4-7-1 Onna, Atsugi, Kanagawa, 243-8510, Japan (E-mail: taku.takahama.tz@hitachi-automotive.co.jp)

2) The MathWorks GK

4-15-1 Akasaka, Minato-ku, Tokyo, 107-0052, Japan

ABSTRACT: This paper presents a design method of a Model Predictive Control (MPC) with low computational cost for a practical Adaptive Cruise Control (ACC) running on an embedded microprocessor. Generally, a problem with previous ACC is slow following response in traffic jams, in which stop-and-go driving is required. To improve the control performance, it is important to design a controller considering vehicle characteristics which significantly changes depending on driving conditions. In this paper, we attempt to solve the problem by using MPC that can explicitly handle constraints imposed on, e.g., actuator or acceleration response. Furthermore, we focus on decreasing the computational load for the practical use of MPC by using low-order prediction model. From these results, we developed ACC with high responsiveness and less discomfort even for traffic jam scene.

KEY WORDS: Adaptive cruise control. Model predictive control.

1. Introduction

ACC with a stop-and-go function, has been widely commercialized. It enables drivers to be free from driving stress in traffic jams and has the potential to enhance safety driving. However, during the traffic jam, conventional ACCs have a slower response time than normal drivers' traffic flow. The response delay is caused by using general control methods designed based on a linear time invariant model, and not making full use of output characteristics of an engine and brake actuators. Using such ACC, other cars cutting in front of own car from adjacent lane tend to be increased due to the response delay. As a result, traffic flow will be disturbed, and the risk of traffic accident will increase.

Up to now, as for the longitudinal control using the MPC^{(1),(2)}, researches related to stop-and-go⁽³⁾, fuel consumption, vehicle platooning⁽⁴⁾, and multi-objectives⁽⁵⁾ are thriving, however many of them are not able to implement with an embedded microprocessor and are not practical. Since the MPC needs to solve the optimization problem considering future prediction in every sampling period, there is a disadvantage that the calculation load becomes large. Therefore, in this paper, in order to suppress the calculation load of MPC, we propose a designing method for prediction time and prediction model. Furthermore, we devised MPC's weights setting method to relieve discomfort during ACC traffic jam driving, and worked to solve these problems. From these results, we derived a MPC controller with a lighter computation load that can be executed in real time with an embedded microprocessor, and developed an ACC that can be used for traffic jam with high responsiveness and less discomfort.

In the numerical simulation of the ACC, we show the advantages of MPC with high responsiveness compared with an existing control method. We also show that the calculation load problem of MPC can be solved from the measurement result of the processing time of an embedded microprocessor. Moreover, we

show the ACC that has good response even during traffic jam from experimental results in a real traffic.

It is expected that the computing power of embedded microprocessors will continue to increase in the future, however, since functions implemented to an autonomous driving system will also increase, it is desirable that it is a smaller control system, and for popularization it is inexpensive microprocessor is required. This research is also useful in that sense.

The paper is organized as follows. Section 2 describes the ACC problem with a plant model. Section 3 proposes the MPC design method including prediction modeling, optimization problem, and controller tuning strategy. In Section 4 and 5, the result of computer simulation and experimental validation will be shown, respectively. Finally, Section 6 concludes the paper with some future work.

2. Problem Formulation

2.1. Control Problem of ACC

The purpose of ACC is to follow a desired inter-vehicle distance or follow a desired velocity between a preceding vehicle and a host vehicle. In this paper, we consider the control problem C1-C3 as below:

(C1) Inter-vehicle distance following error Δd , defined as a following error between an inter-vehicle distance d and a desired distance d_r , i.e., $\Delta d = d - d_r$ is controlled to converge to zero.

(C2) Velocity following error Δv , defined as a following error between the preceding vehicle's velocity v_p and the host vehicle's velocity v_h , i.e., $\Delta v = v_p - v_h$ is controlled to converge to zero.

(C3) Acceleration of the host vehicle \dot{v}_h is controlled to converge to zero.

In addition, there are some performance requirements that are necessary to actually use ACC on public roads as follows:

- (R1) Relative velocity must be not too large, except for the situation a preceding vehicle is initially detected outside the range.
- (R2) Acceleration of the host vehicle must be greater than or equal to -0.25 [G].
- (R3) Acceleration change must be slow except for traffic jam scene.

2.2. Vehicle Dynamics

A vehicle dynamics model will be built to design MPC and analyze the control performance through computer simulation. The longitudinal dynamics of a host vehicle is given by

$$m\dot{v}_h = ma_f - r_{travel} \quad (1)$$

where m is the vehicle mass, v_h is the vehicle speed, a_f is the traction force converted to acceleration and r_{travel} is the travel resistance. The dynamics of a vehicle actuation including engine, transmission or brake has nonlinear characteristics. In general, the input/output relationship of the actuation dynamics is described as an ordinary differential equation below:

$$\begin{cases} \dot{x}_f = f_{act}(x_f, u) \\ a_f = h_{act}(x_f) \end{cases} \quad (2)$$

where $x_f \in \mathbb{R}^{n_f}$, $u \in \mathbb{R}$ are respectively the state, the input of the actuation system. u is an acceleration command, i.e., a control input calculated by adaptive cruise controller. The output of the system (2) is a_f . The travel resistance r_{travel} contains several factors to resist its motion. A model of the resistive force is expressed as

$$r_{travel} = r_{air} v_h^2 + r_{roll}(v_h) + r_{accel} \dot{v}_h + r_{grad}(\theta) \quad (3)$$

In the right-hand side of the equation (3), from left to right, each term describes the air drag, the rolling resistance, the acceleration resistance and the grading resistance. r_{air} is the air drag coefficient, r_{accel} is the acceleration resistance coefficient and θ denotes the slope angle.

2.2. State-Space Model for ACC System

To build a plant model for the ACC system design, two state variables are defined: inter-vehicle distance following error $\Delta d = d - d_r$ and velocity following error $\Delta v = v_p - v_h$. The d_r is determined based on the constant time headway policy given by

$$d_r = T_{hw}v_h + d_0 \quad (4)$$

where T_{hw} is the constant time headway and d_0 is the stopping distance for safety margin.

Let us define the state variables of the plant as $x = [x_1 \ x_2 \ x_3^T]^T \in \mathbb{R}^{2+n_f}$ with $x_1 = \Delta d$, $x_2 = \Delta v$ and $x_3 = x_f$. Then, the state-space model is formulated as

$$\begin{cases} \dot{x} = f(x, u) + Gv + Hw \\ y = Cx + Jv \end{cases} \quad (5)$$

$$f(x, u) = \begin{bmatrix} x_2 - T_{hw}x_3 \\ -h_{act}(x_f) \\ f_{act}(x_f, u) \end{bmatrix}, \quad G = \begin{bmatrix} T_{hw}/m \\ 1/m \\ 0 \end{bmatrix}, \quad H = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad J = \begin{bmatrix} 0 \\ 0 \\ -1/m \end{bmatrix}$$

where $u \in \mathbb{R}$ and $y = [\Delta d \ \Delta v \ \dot{v}_h]^T \in \mathbb{R}^3$ are the input and output of the plant and $v = r_{travel}$ and $w = \dot{v}_p$ represents disturbances into the plant.

3. Model Predictive Control Design

3.1. Prediction Model for Controller Design

MPC requires a plant model to predict the future behavior of the plant along a finite time horizon. We can use the plant model (5) removing unmeasurable disturbance term as a prediction model.

However, a simple prediction model with lower dimension is strongly desired in order to reduce the computational cost on microprocessor. Therefore, the actuation system model (2) is approximated to a switched linear time variant system as

$$\begin{cases} \dot{x}_f = A_f(t)x_f + B_f(t)u \\ a_f = C_f x_f \end{cases} \quad (6)$$

where the state $x_f \in \mathbb{R}$, i.e., $n_f = 1$ and

$$A_f(t) = \begin{cases} -\frac{1}{T_{eng}}, & \text{if } u(t) \geq a_{thr_off} \\ -\frac{1}{T_{brk}}, & \text{if } u(t) < a_{thr_off} \end{cases}$$

$$B_f(t) = \begin{cases} \frac{K_{eng}(t)}{T_{eng}}, & \text{if } u(t) \geq a_{thr_off} \\ \frac{K_{brk}(t)}{T_{brk}}, & \text{if } u(t) < a_{thr_off} \end{cases}$$

$C_f = 1$.

T_{eng} is the time constant of acceleration using engine, T_{brk} is the time constant of deceleration using brake, a_{thr_off} is the acceleration generated when the throttle valve is closed, $K_{eng}(t)$ and $K_{brk}(t)$ are the steady-state gains depending on time. In the system (6), the dynamics is separated into acceleration and deceleration side and simply modeled as first-order delay system. However, prediction error due to modeling error sometimes degrades control performance such as vibration phenomenon. To compensate the prediction accuracy, the steady-state gains can be changed. For example, $K_{eng}(t)$ is composed of constant value K_c and compensation signal $\Delta K(t)$ as below.

$$K_{eng}(t) = K_c + \Delta K(t) \quad (7)$$

where $\Delta K(t)$ is computed by the filter $\Delta K(s) = L(s)u(s)$ with $L(0) = 0$. With the model (6), the computation load of the prediction can be reduced while the prediction accuracy is increased by updating the parameter value at each sampling time.

Considering the above, the prediction model (5) can be simplified to the linear time variant system with three states below:

$$\begin{cases} \dot{x} = A(t)x + B(t)u + Gv + Hw \\ y = Cx + Jv \end{cases} \quad (8)$$

where $x \in \mathbb{R}^3$ and

$$A(t) = \begin{bmatrix} 0 & 1 & -T_{hw} \\ 0 & 0 & -1 \\ 0 & 0 & A_f(t) \end{bmatrix}, \quad B(t) = \begin{bmatrix} 0 \\ 0 \\ B_f(t) \end{bmatrix}$$

The plant dynamics is affected by the travel disturbance r_{travel} and it can cause prediction error. Especially, the acceleration resistance has an impact on the transient response, i.e., stop and go action. To prevent from deteriorating control performance due to such disturbance, it is important to explicitly consider the disturbance characteristics in MPC design. The acceleration resistance model in (3) is a linear function of the acceleration of the host vehicle. Using state transformation, the disturbance characterization can be built into the prediction model. This simple method can reject the disturbance effect without increasing computation load.

The travel disturbance model (3) can be written as

$$r_{travel} = r_{accel} \dot{v}_h + \tilde{r}_{travel} \quad (9)$$

where the first term is the acceleration resistance and the second term is other resistances $\tilde{r}_{travel} = r_{air} v_h^2 + r_{roll}(v_h) + r_{grad}(\theta)$. Here, let us define a state transformation for x_3 as below.

$$\tilde{x}_3 = \frac{1}{1 + \frac{r_{accel}}{m}} x_3 \quad (10)$$

From the vehicle dynamics (1), the acceleration of the host vehicle becomes

$$\dot{v}_h = x_3 - \frac{1}{m} r_{travel} = \left(1 + \frac{r_{accel}}{m}\right) \tilde{x}_3 - \frac{1}{m} r_{travel} \quad (11)$$

Substituting (11) to (9) gives

$$r_{travel} = r_{accel} \tilde{x}_3 + \frac{1}{1 + \frac{r_{accel}}{m}} \tilde{r}_{travel} \quad (12)$$

The travel resistance can be described using the new state \tilde{x}_3 instead of \dot{v}_h . Finally, the state transformation (10) and (12) yields the prediction model with the new state coordination below:

$$\begin{cases} \dot{x} = A(t)x + B(t)u + Gv + Hw \\ y = Cx + Jv \end{cases} \quad (13)$$

where

$$x = [\Delta d \quad \Delta v \quad \rho_{accel} x_f]^T \in \mathbb{R}^3$$

$$v = \tilde{r}_{travel}$$

$$w = \dot{v}_p$$

and

$$A(t) = \begin{bmatrix} 0 & 1 & -T_{hw} \\ 0 & 0 & -1 \\ 0 & 0 & A_f(t) \end{bmatrix}, \quad B(t) = \begin{bmatrix} 0 \\ 0 \\ \rho_{accel} B_f(t) \end{bmatrix}$$

$$G = \begin{bmatrix} \rho_{accel} \frac{T_{hw}}{m} \\ \rho_{accel} \frac{1}{m} \\ 0 \end{bmatrix}, \quad H = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad J = \begin{bmatrix} 0 \\ 0 \\ -\rho_{accel} \frac{1}{m} \end{bmatrix}$$

$$\rho_{accel} = \frac{1}{1 + \frac{T_{accel}}{m}}$$

In the above model, ρ_{accel} implies the scaling factor for control input and disturbance input. If ρ_{accel} is increased, i.e., the acceleration resistive force gives a large effect, the magnitude of the control input is decreased.

Hereinafter, we use the prediction model below based on (13) and assume that the disturbances v and w are unmeasurable.

$$\begin{cases} \dot{x} = A(t)x + B(t)u \\ y = Cx \end{cases} \quad (14)$$

In the controller, the prediction model (14) must be converted to a discrete-time system.

$$\begin{cases} x_{t+1} = A_d(t)x_t + B_d(t)u_t \\ y_t = C_d x_t \end{cases} \quad (15)$$

One simple method of the discretization is forward difference approximation $\dot{x} \approx (x_{t+1} - x_t)/T_s$ in which T_s is a sampling time and $A_d(t) = (I + T_s A(t))$, $B_d(t) = T_s B(t)$ and $C_d = C$ are obtained.

3.2. Optimization Problem

The control input u , i.e., the acceleration command, is calculated by solving the constrained optimization problem below during each sampling period:

$$\min_u J = \sum_{k=1}^p \{y_{t+k}^T Q_t y_{t+k} + \Delta u_{t+k}^T R_t^{\Delta u} \Delta u_{t+k} + u_{t+k}^T R_t^u u_{t+k}\} \quad (16)$$

s.t.

$$\begin{cases} y_t^{\min} \leq y_{t+k} \leq y_t^{\max} \\ \Delta u_t^{\min} \leq \Delta u_{t+k} \leq \Delta u_t^{\max} \\ u_t^{\min} \leq u_{t+k} \leq u_t^{\max} \end{cases}$$

where t is the current time, p is the prediction horizon (number of interval), Δu_* is the increment of the control input, and the suffix ‘‘min’’ and ‘‘max’’ in the inequality constraints denotes lower bound and upper bound. Q_t , $R_t^{\Delta u}$ and R_t^u are the positive-semi definite weight matrices for the following error, the change rate and the magnitude of the control input, respectively.

The linear MPC optimization problem above results in Quadratic Programming (QP) problem. Hence, QP solvers are available such as active-set methods and interior-point methods to solve the problem. Once an optimal solution, i.e., a control input sequence $u^* = [u_t^* \ u_{t+1}^* \ \dots \ u_{t+p-1}^*]^T$ along prediction horizon is numerically obtained, we only use the first element u_t^* of the sequence as an actual control input.

In our ACC problem, the inter-vehicle distance error y_1 , and also the velocity error y_2 and the acceleration y_3 must be regulated to converge to the origin under the upper and lower limits of the input and output variables, e.g., the acceleration command input u and output y_3 . In (16), let the weight matrices be of the form below:

$$\begin{cases} Q_t = \text{diag}(w_{y1}(t), w_{y2}(t), w_{y3}(t)) \\ R_t^{\Delta u} = w_{\Delta u}(t) \\ R_t^u = w_u(t) \end{cases} \quad (17)$$

where $\text{diag}(\cdot)$ represents a diagonal matrix. Each weight value can be changed at every sampling time.

3.2. MPC Parameter Design

To achieve satisfactory following performance without feeling uncomfortable, the parameters in MPC should be appropriately designed and tuned. The MPC parameter includes prediction horizon, control horizon and weights in the performance index. In this subsection, we will show the parameter design and tuning strategy obtained through a number of closed-loop simulation and experiment.

Let us define a relative horizon distance as $r_h = p - c$ in which p is the prediction horizon and c is the control horizon. Let $w_y(t) = [w_{y1}(t) w_{y2}(t) w_{y3}(t)]$ be a row vector of the weights for the output variables.

STEP 1. First, determine the sampling time T_s , the prediction horizon p and the control horizon c by tuning MPC with a linear invariant prediction model approximated to either of engine or brake dynamics. Set $w_y(t) = [1 \ 0 \ 0]$ and $w_{\Delta u}(t) = w_u(t) = 0$ to emphasize the inter-vehicle distance error. The plant model in simulation is the same as the simplified prediction model. The value of T_s , p and c are chosen in reference to the knowledge obtained as below:

- The larger r_h is, the slower the following response is.
- The smaller r_h is, the faster the following response is. Especially when p is smaller, the response tends to suffer vibration like windup phenomenon, depending on the control input constraint.
- The smaller the sampling time is, the faster the following response is.

Also, the weights $w_y(t)$, $w_{\Delta u}(t)$ and $w_u(t)$ have the effects on the ACC system behavior as below:

- If $w_{y2}(t)$ is larger, MPC focuses on the velocity following to the previous vehicle.
- If $w_{y3}(t)$ is larger, the magnitude of the acceleration \dot{v}_h is reduced and the overall following responses are slower.
- If $w_{\Delta u}(t)$, $w_u(t)$ is larger, the magnitude of Δu , u is reduced and the overall following responses are slower.

These intuitive relationship between the weight and the resulting behavior helps us tune the controller for various scenes.

STEP 2. Using a phase plane as depicted in Fig.1, divide the operating region and specify a set of the weights $w_y(t)$, $w_{\Delta u}(t)$, $w_u(t)$ in each region. The purpose of control is to make the state keep or converge to the origin in the region #9. The basic method of the weighting based on the behavior on the phase plane is as follows:

- In the region #9, set $w_{y3}(t)$ as a comparatively larger value to moderate accelerating or braking manipulation and avoid uncomfortable feeling in the steady state. The area of the region becomes larger according to the host vehicle velocity.
- In the region #1-3, set $w_{y1}(t)$ and $w_{y2}(t)$ as comparatively smaller values to prevent the host vehicle from accelerating with full throttle when a preceding vehicle is detected faraway
- In the region #5 and #8, set $w_{y3}(t)$ as comparatively smaller values to quickly respond to the sudden deceleration of a preceding vehicle.
- In the region #4 and #6, set $w_{y1}(t)$ as comparatively smaller values to avoid uncomfortable feeling due to deceleration of the host vehicle when the inter-vehicle distance is close and the relative velocity is increasing.

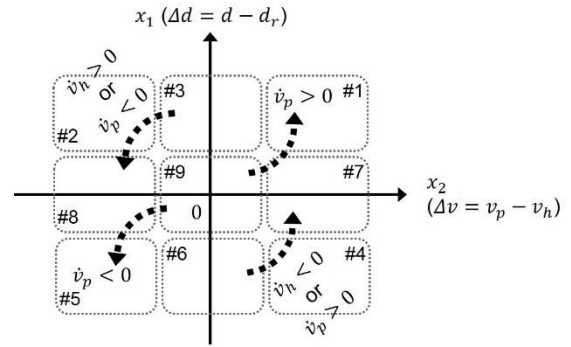


Fig. 1 The operating region division in phase plane for the weight design.

4. Numerical Simulation

Numerical simulation was carried out to evaluate the performance of the MPC applied to the ACC system. The control performance was compared with linear quadratic regulator (LQR).

In the simulation scenario, assuming a stop-and-go traffic jam scene, the both of the host and preceding vehicle are initially stopped and the initial inter-vehicle distance is 6.1 m. Then, the preceding vehicle rapidly accelerates at 2 m/s² from 0 to 10 m/s and finally deaccelerates to stop.

In the simulation, the controller's sampling time T_s is 0.05 sec. The constant time headway T_{hw} is 1.3 sec. The prediction horizon $p = 20$ and control horizon $c = 1$ are chosen to be as small as possible. The control input and its change rate have the lower and upper limit $-2.5 \leq u \leq 1.5$ and $-1.5 \leq \Delta u \leq 1.5$. Table 1 shows the parameters of the actuation system model (6) for prediction. As seen in the table, there is a difference in the time constant and the steady-state gain between engine acceleration and braking deceleration. In the detailed model (2), we assumed an overshoot occurs in the engine acceleration response. To compensate the prediction error, $\Delta K(t)$ is added and computed by the 2-order filter $F(s)$ like a band-pass filter.

Table. 1 The parameters of the actuation system model.

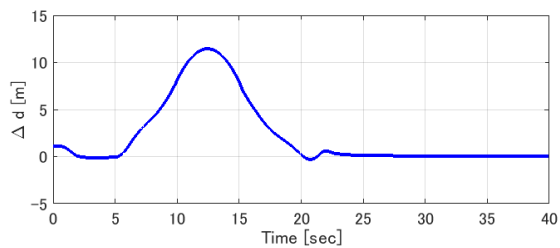
T_{eng}	0.46
$K_{eng}(t)$	$0.732 + \Delta K(t)$ where $\Delta K(t)$ is obtained through the filter $\Delta K(s) = F(s)u(s)$. $F(s) = \frac{1.5s}{s^2 + 3s + 4}$
T_{brk}	0.193
$K_{brk}(t)$	0.979

MATLAB®, Simulink® and Model Predictive Control Toolbox™(6),(7) were used in the simulation. The optimization problem is solved by a QP solver, based on KWIK algorithm(8), in Model Predictive Control Toolbox. We used "Adaptive MPC Controller" block provided by the toolbox to model the controller and run the closed-loop simulation with the plant model in Simulink. C-code can be generated from the MPC block for the implementation to an embedded microprocessor.

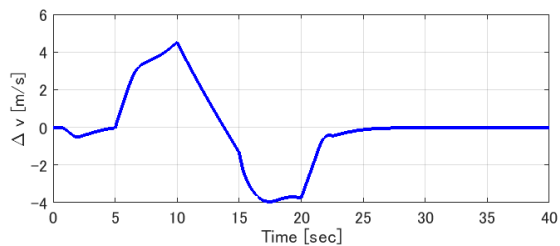
Fig. 2 and 3 illustrates the simulation result of MPC and LQR. Each figure shows, from above, (a) inter-vehicle distance following error y_1 , (b) relative velocity y_2 and (c) preceding vehicle's acceleration \dot{v}_p , host vehicle's acceleration y_3 and control input u . As can be seen in Fig. 2, the MPC achieves that all the plant output y_i ($i = 1, 2, 3$) converges to zero while those input constraints are

satisfied. To follow the preceding vehicle fast, the MPC tries to make full use of the actuator capability. On the other hand, as shown in Fig. 3, the LQR preforms slower response compared with the MPC. The weight of the LQR is tuned on a trial and error basis to meet the input constraints for this specific scenario. The response is conservative due to the small feedback gain to reduce overall magnitude of the control input by using such linear controller.

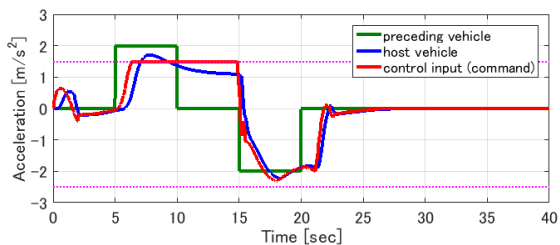
The advantage of MPC for ACC is that we can realize high control performance since such constraints can be explicitly dealt with and the tuned weights can be intuitively and flexibly designed as the function of time or if-then rule, considering various driving situations.



(a) inter-vehicle distance following error y_1

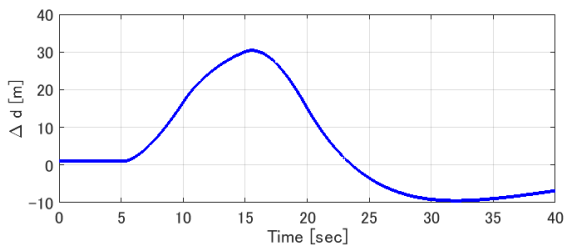


(b) relative velocity y_2

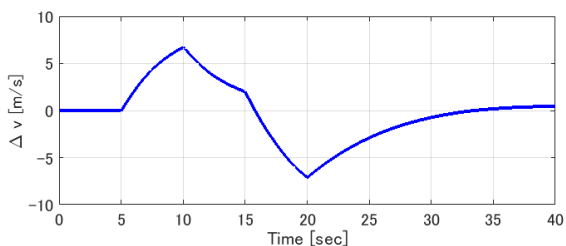


(c) acceleration \dot{v}_p , y_3 , and u

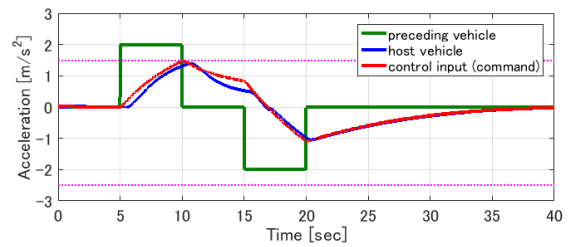
Fig. 2 The simulation result of MPC.



(a) inter-vehicle distance following error y_1



(b) relative velocity y_2



(c) acceleration \dot{v}_p , y_3 , and u

Fig.3 The simulation result of LQR.

5. Experiment

The system configuration of the experimental vehicle which we prepared is shown in Fig. 4. A stereo camera is used for relative distance sensor, and relative velocity is obtained from a digital derivation of the relative distance. The other signals such as vehicle velocity are obtained via CAN communication. The desired engine torque is obtained by converting acceleration command of the MPC with an engine characteristic map and an estimated travel resistance. The desired brake pressure is obtained by converting acceleration command of the MPC with an engine brake characteristic map and an estimated travel resistance.

The MPC controller was implemented in an embedded microprocessor (Renesas SH-4A, 32-bit processor), we confirmed the processing time of the MPC. The measurement result is shown in Fig. 5, the average time of the ACC function was 1.1ms. The C-code is automatically generated from a Simulink model using Embedded Coder[®].

Evaluation of the ACC was conducted on the metropolitan expressway because there are many traffic conditions such as traffic jam and high speed scene. Results of the experimentation are shown in Fig. 6 (MPC) and Fig. 7 (normal driver). From the comparing the both results at time=21–26[s] which is the acceleration scene during traffic jam, we were able to obtain the equivalent response delay (approx. 1.5s) as a normal driver. Also, it can be seen that sudden acceleration/deceleration is suppressed in the high speed driving scene in Fig. 8. This is because the weight for the change in the control input is functioned effectively. From the above, it is understood that the response is good even in the traffic jam scene, and the controller is functioning with less discomfort in the high speed driving scene.

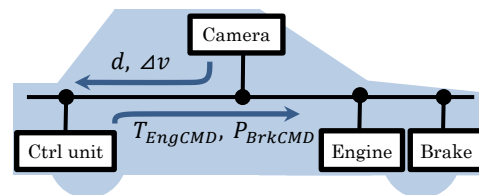


Fig. 4 System configuration of the experimental vehicle

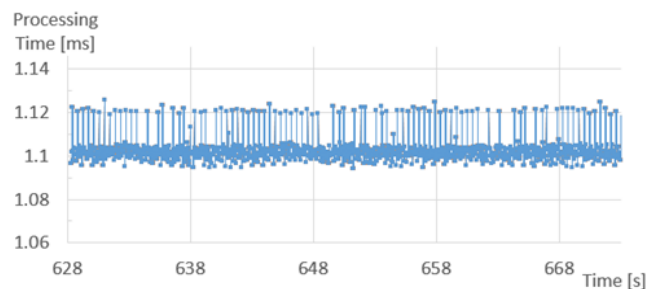


Fig. 5 The processing time of the proposed MPC

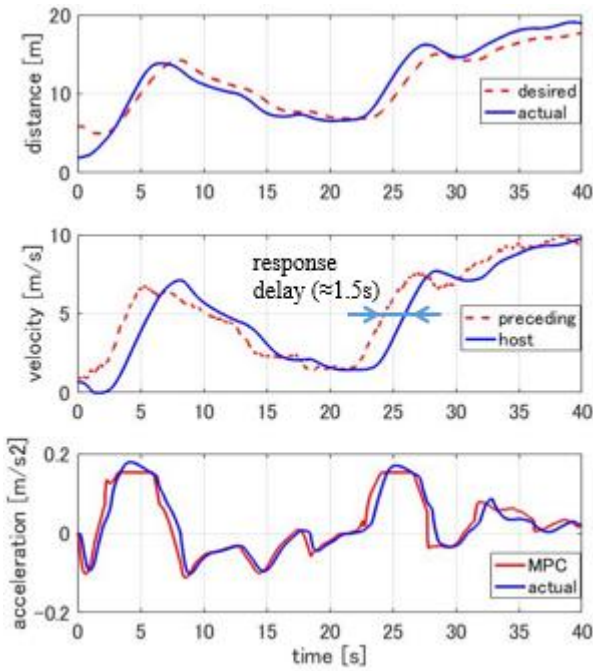


Fig. 6 Experimental result during traffic jam

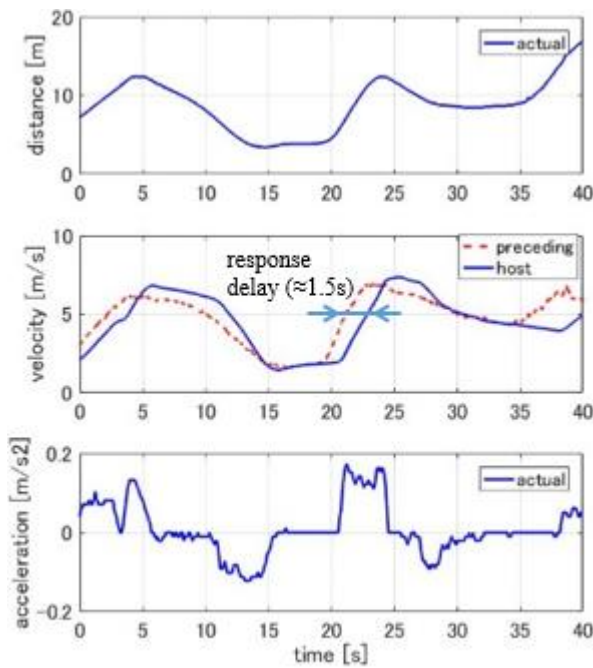


Fig. 7 Vehicle behavior driven by a normal driver

6. Conclusion

In this paper, we proposed a method of designing a practical MPC suitable for embedded processors corresponding to various traffic situations. Computer simulation showed that the MPC is superior to conventional controllers because it treats actuator constraints and has a higher response than ACC. Experimental verification results showed that the proposed MPC controller can be implemented in embedded microprocessors and can achieve high responsiveness and less discomfort.

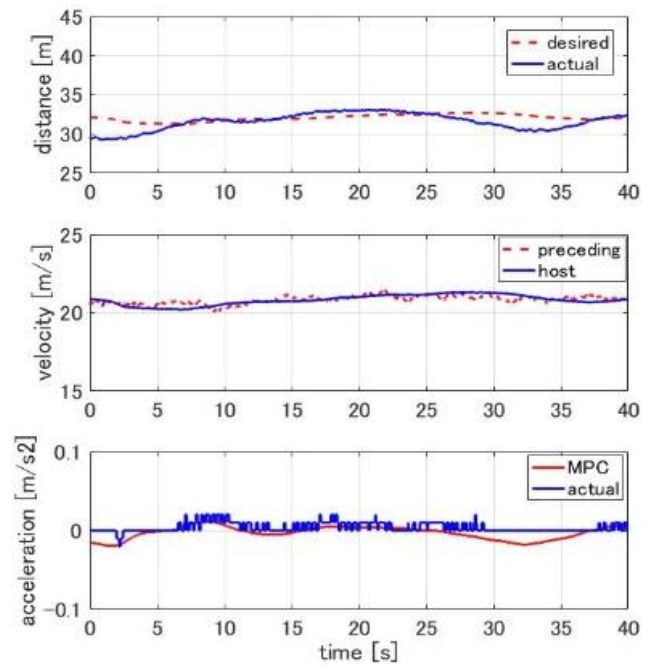


Fig. 8 Experimental result during high speed driving

As future tasks, reduction of discomfort in tight curves by cooperation with a lateral control system, improvement of operability by cooperation with a driver, and so on are considered.

References

- (1) J. Rawlings and D. Mayne : Model Predictive Control: Theory and Design, Nob Hill Publishing (2010).
- (2) J. Maciejowski, Predictive Control with Constraints. Harlow, UK: Prentice Hall (2002).
- (3) G. Naus, R. Bleek, J. Ploeg, B. Scheepers, R. Molegraf, and M. Steinbuch : Explicit MPC Design and Performance Evaluation of an ACC Stop and GO, in Proc. Amer. Contr. Conf., pp. 224-229 (2008).
- (4) H. Aoki, M. Mukai, T. Kawabe : Model Predictive Control for Eco-Driving Based on Prediction of Preceding Vehicle Motion with Traffic Signal Information, SICE Annual Conference (2014).
- (5) S. Li, K. Li, R. Rajamani, and J. Wang : Model Predictive Multi-Objective Vehicular Adaptive Cruise Control, IEEE Trans. Contr. Systems Technology, Vol. 19, No. 3, pp. 556-566 (2011).
- (6) A. Bemporad, M. Morafi, and N. Ricker : Model Predictive Control Toolbox™ User's Guide, The MathWorks, Inc., https://www.mathworks.com/help/pdf_doc/mpc/mpc_ug.pdf
- (7) A. Bemporad : Model Predictive Control Design: New Trends and Tools, in Proc. 45th IEEE Conf. Dec. Control, pp. 6678-6683 (2006).
- (8) C. Schmid and L.T. Biegler : Quadratic programming methods for reduced Hessian SQP, Computers & Chemical Engineering, Vol. 18, No. 9, pp. 817-832 (1994).