ROBOMECH Journal

RESEARCH ARTICLE Open Access

A proposal for adaptive cruise control balancing followability and comfortability through reinforcement learning

Nagayasu Maruyama* and Hiroshi Mouri

Abstract

Adaptive cruise control (ACC), which is an extension of conventional cruise control, has been applied in many commercial vehicles. Traditional ACC is controlled by proportional-integral-derivative control or linear quadratic regulation (LQR), which can provide sufficient performance to follow a preceding vehicle. However, they can also cause excessive acceleration and jerk. To avoid these excessive behaviors, we propose reinforcement learning (RL), which can consider various objectives to determine control inputs, as an ACC controller. To balance the performance of following a preceding vehicle and reducing jerk, RL rewards are designed using unique thresholds. Additionally, to balance performance and robustness to the zero-order delay (dead time) of the controlled system, dead time is also considered by scattering it randomly in the learning phase. As a result of this study, an RL agent trained using the proposed RL method and two LQR units specialized for followability and comfortability were simulated using Simulink® (MATLAB®).

Keywords: Vehicle trajectory, Autonomous driving, Adaptive cruise control, Reinforcement learning

Introduction

In recent years, advanced driver-assistance systems (ADAS) have been studied to enhance driving comfort, reduce driving fatigue and stress, reduce accident risks, increase traffic capacity, and reduce fuel consumption [1, 2]. In particular, from the perspectives of driving fatigue and stress reduction, the adaptive cruise control (ACC) system is an important type of ADAS that has been deployed in commercial vehicles since 1995 in Japan [3]. ACC is an extension of the conventional cruise control system, which could only maintain a vehicle's preset velocity, whereas ACC maintains a vehicle's preset distance from a preceding vehicle [4, 5]. ACC performs acceleration and deceleration for a driver based on information observed by various devices such as radar, sensors, and cameras. Even conventional control methods such as proportional-integral-derivative control and linear quadratic regulation (LQR) [6], which have been used commonly for decades, provide sufficient performance to follow a preceding vehicle. However, they can cause excessive acceleration and jerk, which can reduce driving comfort and increase fuel consumption [7]. To avoid these excessive behaviors, ACC should be improved to consider not only followability, but also comfortability (reducing jerk).

In addition, it is also known that these conventional control methods cannot consider the constraint saturated by performance's limits, and it makes the control performance worse. To avoid this deterioration, the model predictive control (MPC) is usually used [8–10]. However, the MPC's control performance depends on the accuracy of modeling the controlled plant (the MPC determines control input by calculating with a modeled plant every sampling time) [11].

To consider the robustness against the modeling accuracy and the disturbance, the fuzzy logic control (FLC) has been also studied for decades. This controller can deal the plant state fuzzily based on the linguistic control rules (LCR). However, this method cannot decide control input

*Correspondence: s229186q@st.go.tuat.ac.jp
Department of Engineering, Tokyo University of Agriculture and Technology, Japan, Koganei

Springer Open

smoothly due to the LCR, so the control input keeps constantly on increasing and decreasing [12].

To solve this problem, in this study, reinforcement learning (RL), which can consider various objectives to determine control inputs, was considered for improving ACC. The RL can also consider the constraints. Additionally, to balance the performance of followability and comfortability, an RL reward function was designed using unique thresholds. These thresholds vary depending on the speed of the host vehicle. When the difference between the target distance and measured distance from the host vehicle to the preceding vehicle becomes less than a certain threshold, the RL agent collects rewards depending on the difference in distance. Another threshold for reducing jerk is set to help passengers feel "comfortable." If jerk becomes less than a certain threshold, the RL agent collects rewards.

Additionally, it is assumed that there is zero-order delay (dead time) in the controlled system. Therefore, one of the goals of the proposed RL agent is to enhance robustness to dead time by learning random variations during the learning phase of RL.

Definition of ACC and the controlled vehicle model

Definition of ACC objectives and conditions

For this research, the ACC objectives were defined as listed below.

O1) To achieve the required performance to follow the preceding vehicle, the error between the reference distance and measured distance to the preceding vehicle should converge to zero.

O2) To achieve the required performance for following a preceding vehicle, the relative velocity between the host vehicle and preceding vehicle should converge to zero (this object is related to O1, so servo balance between O1 and O2 should be tuned experimentally.

O3) To achieve the required performance for driving comfort and fuel consumption, the acceleration and jerk of the host vehicle should converge to zero.

Additionally, to consider an ACC system implemented into an actual vehicle, the following conditions are defined.

C1) The host vehicle has dead time in the controlled system. Therefore, the host vehicle's acceleration responds to a control command from the ACC system with a first-order delay and dead time.

C2) The reference distance which is the ACC system's target is defined based on the policy of time headway (THW). The THW is the time the host vehicle takes to reach a point on the road passed by the preceding vehicle. This parameter is one of factor used by the driver to perceive the risk to close to the preceding vehicle, and its magnitude influences the stress level of the driver when following a preceding vehicle [12].

C3) Acceleration and deceleration are saturated by performance's limits of powertrain, brake and road friction.

C4) States (vector) of the controlled plant, which is defined in "Definition of a vehicle model controlled by ACC" section, are observable.

A scenario in which a host vehicle follows preceding vehicle using ACC is illustrated in Fig. 1

Definition of a vehicle model controlled by ACC

According to C1, the host vehicle's acceleration is defined under the first-order delay as follow:

$$\dot{a}_h = \frac{1}{\tau}(u - a_h).$$

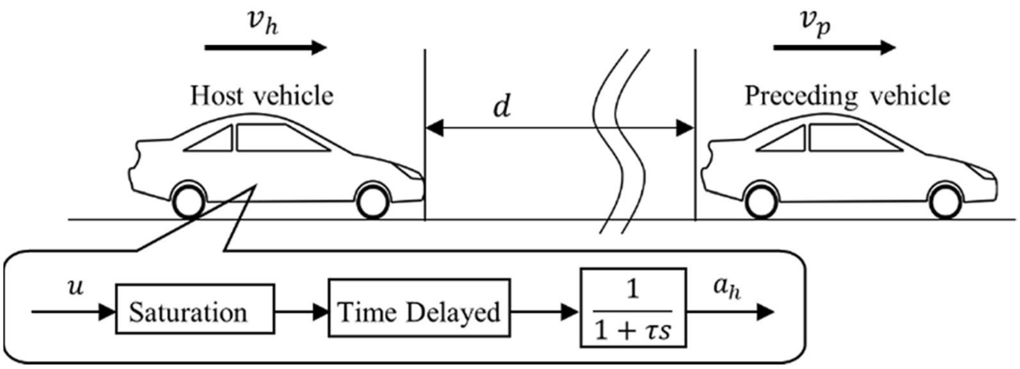


Fig. 1 Scenario for ACC control

Here a_h is the host vehicle's acceleration, u is the acceleration control command from the ACC system, and τ is the time constant of the first-order delay system. By the Laplace transform, this equation expressed as a time function is transformed to (1) with the Laplace operator s:

$$sa_h(s) = \frac{1}{\tau}u(s) - \frac{1}{\tau}a_h(s)$$

$$\frac{a_h(s)}{u(s)} = \frac{1}{1+\tau s}. \tag{1}$$

According to C2, the reference distance to preceding vehicle of the ACC system based on the THW policy is defined as (2)

$$d_r = d_s + t_{hw}v_h, \tag{2}$$

where d_r is the reference distance to preceding vehicle, d_s is the safety distance based on the THW policy, t_{hw} is the THW, and v_h is the velocity of the host vehicle.

Based on the calculated d_r value, the error between the reference distance and measured distance to preceding vehicle are defined in (3). Additionally, the error between the host and preceding vehicle velocity is defined in (4).

$$d_e = d - d_r = d - d_s - t_{thw}v_h \tag{3}$$

$$v_e = v_p - v_h \tag{4}$$

Here, d_e is the error between the reference distance and measured distance, d is the measured distance to preceding vehicle, v_e is the error between the host vehicle velocity and preceding vehicle velocity, and v_p is the velocity of the preceding vehicle.

The controlled plant state vector $\boldsymbol{x}$ and output vector $\boldsymbol{y}$ are defined in (5) and (6), respectively.

$$\boldsymbol{x} = \begin{bmatrix} d & v_e & v_h & a_h \end{bmatrix}^T \tag{5}$$

$$\boldsymbol{y} = \begin{bmatrix} d_e & v_e & a_h & u \end{bmatrix}^T \tag{6}$$

Based on the defined state vector $\boldsymbol{x}$ and output vector $\boldsymbol{y}$, the state-space representation is defined in (7a) and (7b).

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u + \boldsymbol{G}a_p \tag{7a}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}u - \boldsymbol{Z} \tag{7b}$$

Here, a_p is the preceding vehicle's acceleration. The coefficient matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{G}, \boldsymbol{C}, \boldsymbol{D}$, and $\boldsymbol{Z}$ in (7a) and (7b) are expressed as follows:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{\tau} \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{\tau} \end{bmatrix}, \boldsymbol{G} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix},$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & -t_{hw} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \boldsymbol{D} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \boldsymbol{Z} = \begin{bmatrix} d_0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The reinforcement learning applied to an ACC controller

Algorithm for the reinforcement learning

In this study, the deep deterministic policy gradient (DDPG) [13] was used for learning ACC driving behavior to archive the objectives O1 to O3 under the conditions C1 to C4. The DDPG is a common RL method and deep learning is adopted in the actor-critic method. The controller implemented into host vehicle to determine ACC driving behaviors based on learning results is called an "agent." To design the agent and evaluate its performance, the RL toolbox (MATLAB®) was used to implement its RL as a controller.

Design of the actor network and critic network

For the DDPG, there are two networks called the actor network and critic network, each of which have different purposes. The agent observes the state (vector) from the environment surrounding the host vehicle and the actor network determines the best action based on the observed state. The critic network then receives the state from the agent and action from the actor, and returns the expected value of the total reward. Finally, after executing the action from the actor, the actor network is updated based on the calculated reward. A block diagram for an ACC system controlled by this logic is presented in Fig. 2. L in the figure

Table 1 Actor network structure

Network structure of actor	
Name	**State path**
InputLayer	$7 \times 1 \times 1$
FullyConnectedLayer	fc11
ReluLayer	relu11
FullyConnectedLayer	fc12
ReluLayer	relu12
FullyConnectedLayer	fc13
ReluLayer	relu13
FullyConnectedLayer	fc14
TanhLayer	tanh11
ScalingLayer	scale = 2.5, bias = − 0.5

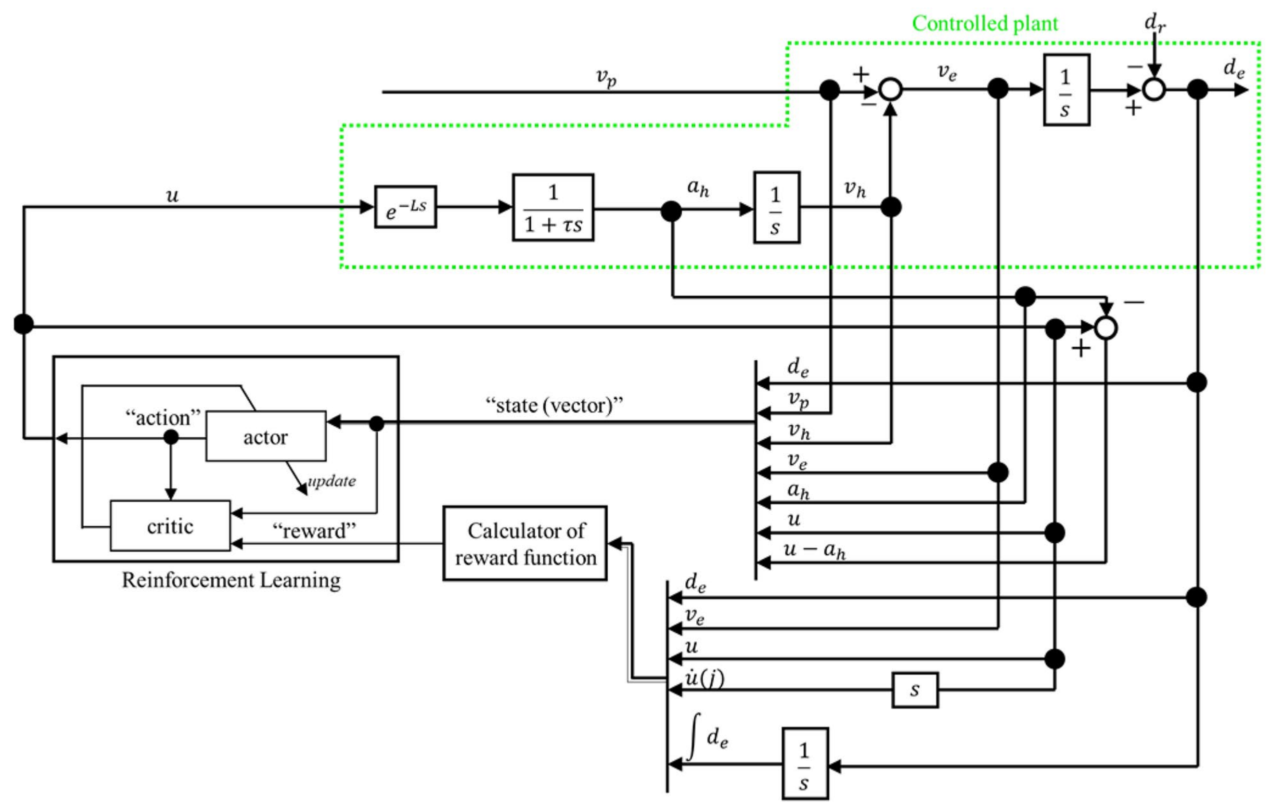


Fig. 2 Block diagram for an ACC system controlled by RL

Table 2 Critic network structure

Network structure of critic		
Name	**State path**	**Action path**
InputLayer	7 × 1 × 1	1 × 1 × 1
FullyConnectedLayer	fc21	fc51
ReluLayer	relu21	–
FullyConnectedLayer	fc22	–
AdditionLayer	add2	
ReluLayer	relu22	
FullyConnectedLayer	fc23	
ReluLayer	relu23	
FullyConnectedLayer	fc24	

Table 3 Initial state for host vehicle and preceding vehicle

State	Value	
	Host vehicle	**Preceding vehicle**
Initial distance	70 m	
Initial vehicle velocity	20 m/s	25 m/s

is the dead time and e^{-Ls} is the dead time factor in the controlled system.

The structures of the actor network and critic network are listed in Tables 1 and 2, respectively.

In the input and output layers, the number of neurons for the state input is four and the number of neurons for the action output is one. Additionally, each fully connected layer contains 48 neurons. There are seven neurons in the input layer on the state path, and four neurons of them are represented as the plant’s output from vector $\boldsymbol{y}$ defined in (6) and three neurons are added to consider the environment precisely. In addition, there is one neuron in the input layer on action path, and it is represented as the acceleration command u defined in (1).

Design of the reward function

To archive O1 to O3, the reward function r_t at time t is defined by (8).

$$r_t = -w_1 d_{et}^2 - w_2 v_{et}^2 - w_3 u_t^2 - w_4 j_t^2 - w_5 \int_0^t d_{eT}^2(T)dT + M_t \tag{8}$$

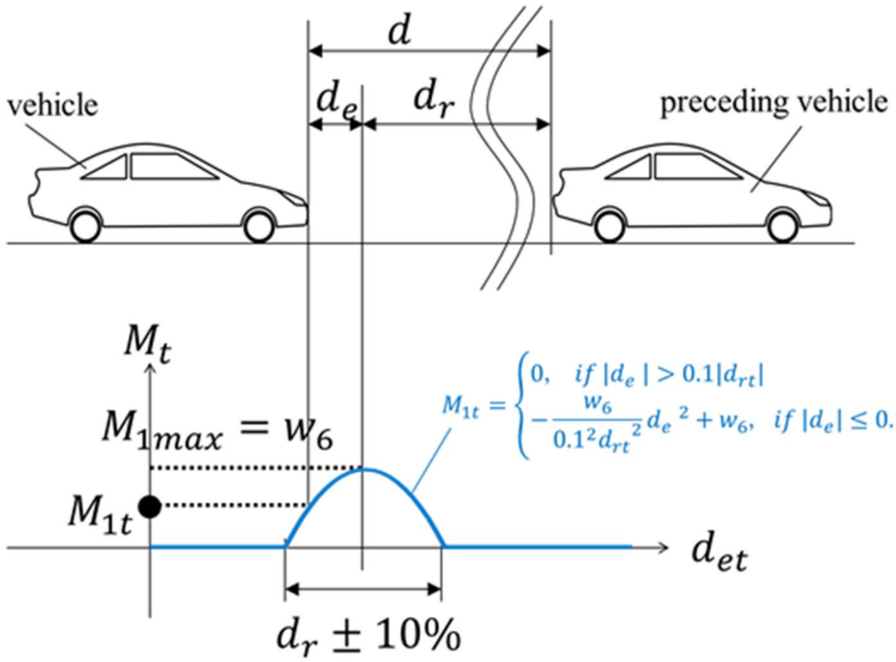

d
d_e
d_r
host vehicle
preceding vehicle
M_t
$M_{1max} = w_6$
M_{1t}
$M_{1t} = \begin{cases} 0, & if\ |d_e| > 0.1|d_{rt}| \\ -\frac{w_6}{0.1^2 {d_{rt}}^2} {d_e}^2 + w_6, & if\ |d_e| \leq 0.1|d_{rt}| \end{cases}$
d_{et}
$d_r \pm 10\%$

Fig. 3 Logic of the reward value M_{1t} ($d > d_r$)

Table 4 Conditions for ACC simulations

State	Value
d_s	10 m
t_{hw}	1.4 s
τ	0.5

Here, w_1 to w_5 are weights, M_t is a reward value, and j_t is the jerk.

The reward value M_t can receive a reward only when the error between the reference distance and measured distance to preceding vehicle or jerk become less than the corresponding thresholds, which are defined as follows. As shown below definition, the value of M_t depends on how much the measurement values exceed the referential values. The reward value of M_t is defined in (9).

$$M_t = M_{1t} + M_{2t} \tag{9}$$

$$M_{1t} = \begin{cases} 0, & if\ \left|d_{et}\right| > 0.1\left|d_{rt}\right| \\ -\frac{w_6}{0.1^2 d_{rt}^2} d_{et}^2 + w_6, & if\ \left|d_{et}\right| \leq 0.1\left|d_{rt}\right| \end{cases}$$

Table 5 Tuned weights for the reward function for RL

State	Value
w_1	0.0003
w_2	0.00001
w_3	0.005
w_4	0.05
w_5	0.000002
w_6	1.5
w_7	1.3

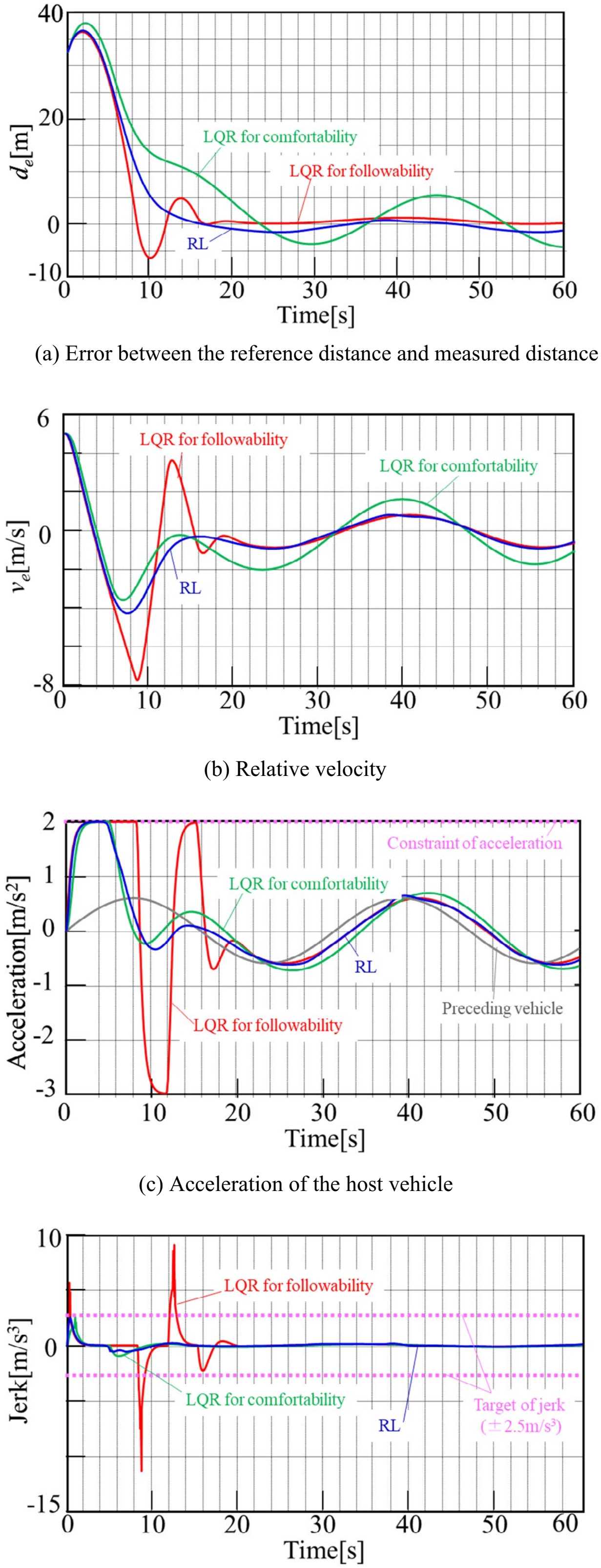
$$M_{2t} = \begin{cases} 0, & if\ j_t > 2.5 \\ w_7, & if\ j_t \leq 2.5 \end{cases}$$

Here, M_{1t} is a reward value that reduces the error between the reference distance and measured distance to preceding vehicle, and M_{2t} is a reward value for reducing jerk. For these reward values, the weights are defined as w_6 and w_7, respectively.

The purpose of M_{1t} is to determine the control target depending on the host vehicle velocity. When the host vehicle velocity is high, the ACC controller wants to avoid approaching the distance to the preceding vehicle to reduce collision risk. In contrast, when the host vehicle velocity is low, the ACC controller's target is to close the distance to the preceding vehicle to increase traffic

(See figure on next page.)

Fig. 4 Simulation results for RL vs. LQR ($L = 0.02$ s)



(a) Error between the reference distance and measured distance

(b) Relative velocity

(c) Acceleration of the host vehicle

(d) Host vehicle jerk

Fig. 4 (See legend on previous page.)

capacity. To achieve the purpose of M_{1t}, M_{1t} should be calculated depends on host vehicle velocity. Thus, in this case, d_{rt} is used to calculate M_{1t} because d_{rt} includes host vehicle velocity as one of factors. The logic of M_{1t} is illustrated in Fig. 3. M_{1t} is calculated only when the absolute value of d_{et} becomes less than 10% of the absolute value of d_{rt}; In other case when the absolute value of d_{rt} becomes more than 10% of the absolute value d_{rt}, M_{1t} becomes zero. The criteria value of 10% is tunable and it is defined experimentally in advance. The maximum value of M_{1t} is w_6.

The purpose of M_{2t} is to control the ACC comfortably. To achieve the purpose of M_{2t}, M_{2t} becomes w_7 only when the absolute value of the jerk becomes less than the threshold of 2.5 m/s^3. This value was selected because passengers feel uncomfortable if the jerk exceeds 2.5 m/s^3 [7].

If the jerk is calculated from the measured acceleration, then there is a huge delay between the actuator and sensor due to sensing delay and the zero-order delay of data communication. Therefore, in this study, jerk was calculated as the rate of change of the acceleration input during one sampling period, as shown in (10).

$$j_t = \frac{u_t - u_{t-1}}{T_s} \tag{10}$$

Here, T_s is the sampling period.

Definition of the linear quadratic regulation to be compared to the reinforcement learning

To evaluate the advantageous effect of applying an RL agent to an ACC controller, the results obtained using LQR is considered for comparison. The system in (7a) is expanded by defining jerk as the control input u' [14] because the LQR should also consider to reduce jerk to be compared with the RL from the perspective of performance and comfortability fairly. The model definition of (7a) is transformed into (11) as follows:

$$\dot{\boldsymbol{x}}' = \boldsymbol{A}'\boldsymbol{x}' + \boldsymbol{B}'u' + \boldsymbol{G}'a_p, \tag{11}$$

where the state vector $\boldsymbol{x}'$ and coefficient matrices $\boldsymbol{A}'$, $\boldsymbol{B}'$, $\boldsymbol{G}'$ are expressed as shown below.

$$\boldsymbol{x}' = \begin{bmatrix} \boldsymbol{x} \\ u \end{bmatrix}, \boldsymbol{A}' = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ 0 & 0 \end{bmatrix}, \boldsymbol{B}' = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \boldsymbol{G}' = \begin{bmatrix} \boldsymbol{d} \\ 0 \end{bmatrix}$$

The optimal control input for state feedback control is defined below. First, the evaluation function for state feedback control is defined in (12).

$$J = \int_0^\infty \left(\boldsymbol{x}'^T \boldsymbol{Q} \boldsymbol{x}' + r u'^2 \right) dt \tag{12}$$

Here, J is the evaluation function, $\boldsymbol{Q}$ is the weight vector for the state, and r is the weight vector for the input. To minimize this evaluation function, the optimal control input is defined in (13).

$$u' = -f\boldsymbol{x}', \text{ and } f = r^{-1}\boldsymbol{B}'^T\boldsymbol{P} \tag{13}$$

Here, the matrix $\boldsymbol{P}$ is a symmetric matrix based on the algebraic Riccati equation.

$$P\boldsymbol{A}' + \boldsymbol{A}'^T\boldsymbol{P} + r^{-1}\boldsymbol{P}\boldsymbol{B}'\boldsymbol{B}'^T\boldsymbol{P} + \boldsymbol{Q} = 0 \tag{14}$$

For comparison to the RL control method, two weight combinations "$\boldsymbol{Q_d}$ and r_d" and "$\boldsymbol{Q_j}$ and r_j," which are specialized for followability and comfortability, respectively, are defined as shown below.

1. Weight combination specialized in followability:

$$\boldsymbol{Q_d} = \mathrm{diag}[\ 5\ \ 100\ \ 0\ \ 40\ \ 0\ ],\ r_d = 60$$

2. Weight combination specialized in comfortability:

$$\boldsymbol{Q_j} = \mathrm{diag}[\ 5\ \ 100\ \ 0\ \ 50\ \ 50\ ],\ r_j = 3000$$

Note that these weight combinations are chosen by parameter studies. Some of ACC simulation results which are studied to decide weight combinations are shown in "Appendix".

Usually, the zero-order delay on communication can be changed fluently, so it is very difficult to consider it. In addition, the zero-order delay is very small value. Thus, the zero-order delay is often considered as one of a disturbance that can be deal with controller's robustness. That's why LQR doesn't consider the zero-order delay on communication here.

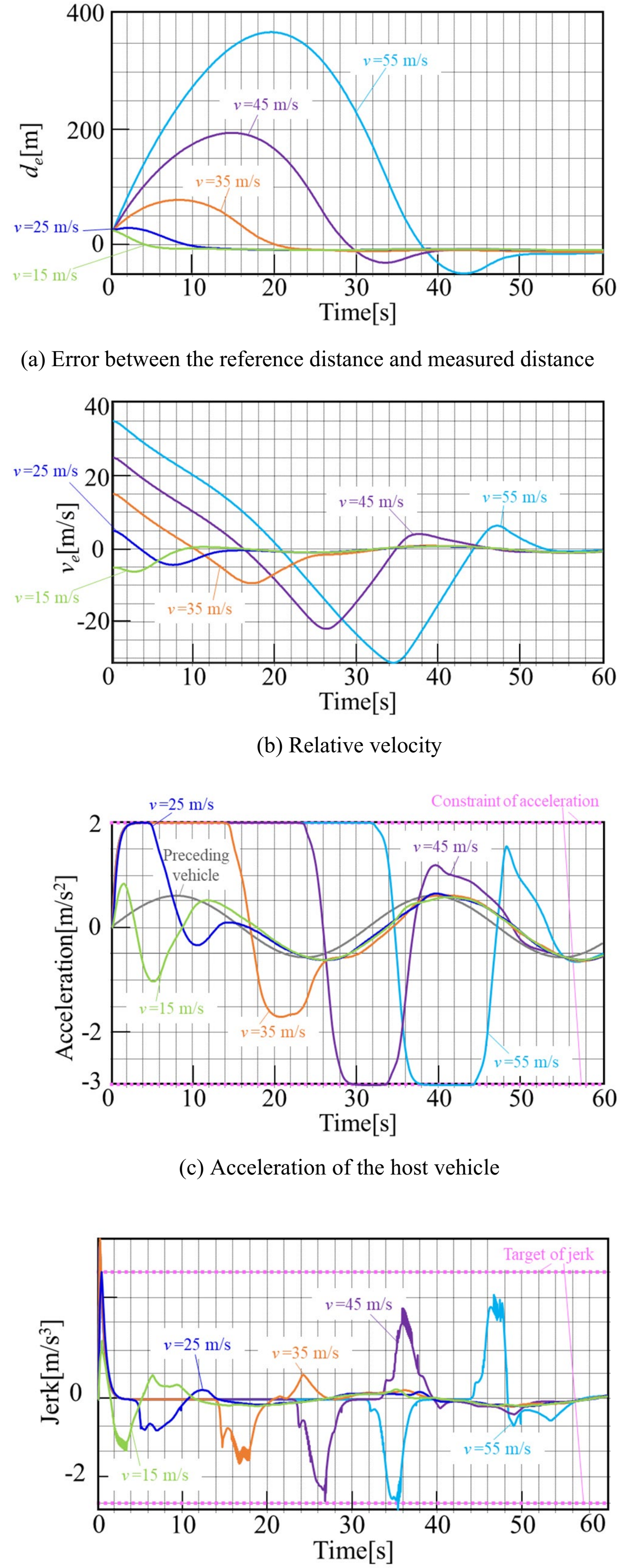
Evaluation of followability and comfortability

Through comparisons to the two controllers for LQR defined in "Definition of LQR to be compared to RL" section, the controller for RL is evaluated based on simulations in this section.

Conditions for simulations

The initial conditions for the host vehicle and preceding vehicle are listed in Table 3. In this simulation, the sampling time was 0.1 s, simulation continued for 60 s, and the preceding vehicle performed repeated accelerations and decelerations during the simulation period.

(See figure on next page.)
Fig. 5 Simulation results for RL (with scattered initial preceding vehicle velocities)



(a) Error between the reference distance and measured distance

(b) Relative velocity

(c) Acceleration of the host vehicle

(d) Host vehicle jerk

Fig. 5 (See legend on previous page.)

Regarding C3, the acceleration and deceleration of the host vehicle were saturated between 2 and -3 m/s^2. The other conditions for conducting simulations are listed in Table 4.

The weights for the reward function for RL were tuned in advance and are listed in Table 5.

To improve the RL controller's followability, the initial velocity of the preceding vehicle was randomly set between 10 and 30 m/s in intervals of 1 m/s during the learning phase. In addition to improving followability, the controller should be robust to the dead time of the controlled system. Therefore, the dead time of the controlled system was also scattered between 0.01 and 0.10 s in intervals of 0.01 s randomly during the learning phase. By this method, the zero-order delay can be considered even if it is changed fluently, so this method requires engineers not to consider the value of the zero-order delay but also the range of the zero-order delay preliminarily. During the learning phase, rewards were returned in every sampling period (0.1 s) and the total rewards were considered as the accumulated results. The threshold for the total reward to stop learning is a tunable parameter because there is a tradeoff between performance and the time required for learning. Typically, after many learning episodes are completed, the total rewards converge and performance decreases with overfitting and overtraining. Additionally, the value of the total reward depends on the definition of the reward function. Hence, the threshold to stop the learning is also tuned in advance depending to the definition of the reward function (for defined reward function, the criteria of the total reward to stop the learning is 1400). For RL, learning was completed when the total reward for an episode with 600 steps reached 1614.3 and the total number of learning steps required to reach the goal was 446,819.

Simulations with one condition of preceding vehicle velocity and dead time

The simulation results when the zero-order delay is 0.02 s and the other conditions are as discussed in "Conditions for simulations" section are presented in Fig. 4. The results for LQR controllers prioritizing followability (to reduce error between the reference distance and measured distance to preceding vehicle) and prioritizing comfortability (to reduce jerk) are plotted as "LQR for followability" and "LQR for comfortability," respectively. The graphs show the (a) error between the reference distance and measured distance to preceding vehicle, (b) relative velocity, (c) host vehicle acceleration, and (d) host vehicle jerk. As mentioned above, jerk is considered as the change rate of the acceleration input during one sampling period in the controller. On the other hand, actual jerk which driver feels should be calculated as the change rate of measured acceleration, so jerk discussed in the graphs is defined (15).

$$j_{tmeasured} = \frac{a_t - a_{t-1}}{T_s} \tag{15}$$

$j_{tmeasured}$ is measured jerk which is calculated by measured acceleration, and it is represented as "Jerk" in below graphs only.

According to the simulation results, our considerations are separated in two perspectives of followability and comfortability.

Followability perspective

The error between the reference distance and measured distance to preceding vehicle controlled by RL converges faster than that controlled by LQR. Additionally, in the steady state [after the host vehicle speed reaches the preceding vehicle speed at approximately 16 s in graph (b)], the relative velocity's overshoot controlled by RL is equal to or better than the overshoot controlled by LQR. The main reason why RL controls with less relative velocity's overshoot than LQR is that RL learns to handle the dead time properly during the learning phase.

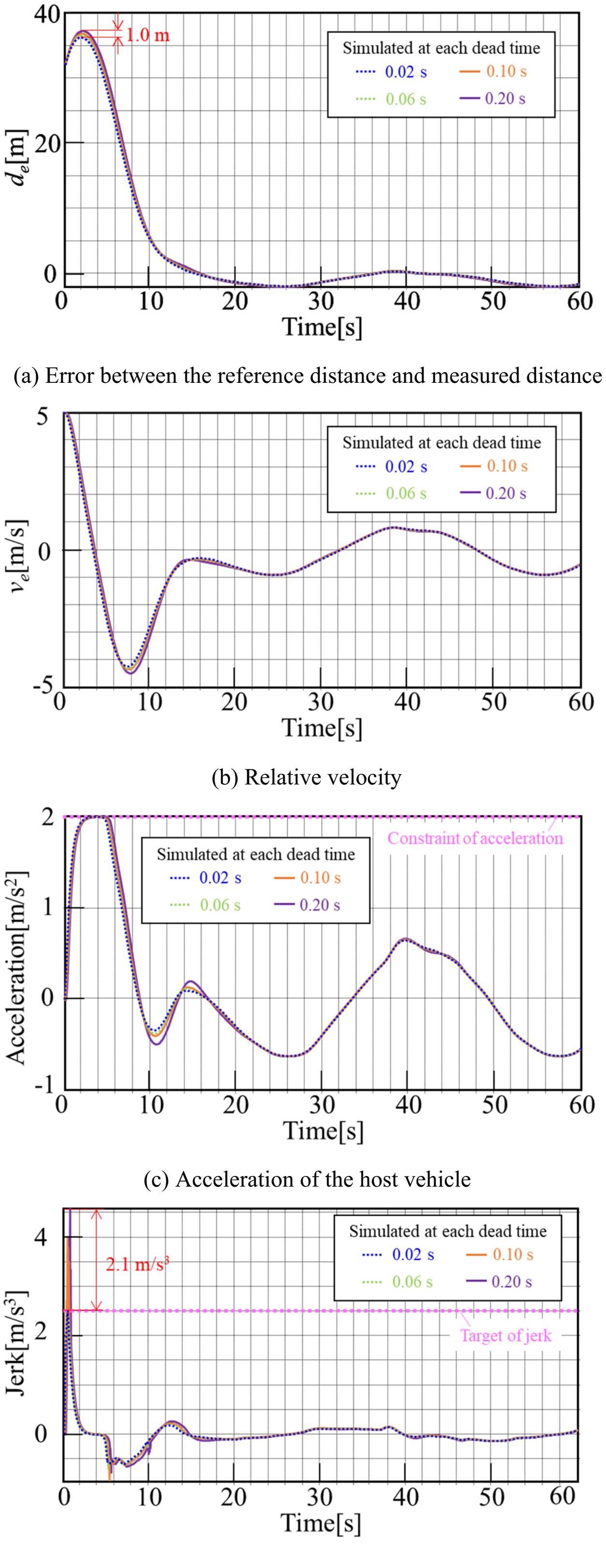
Comfortability perspective

An RL agent with a reward function for suppressing the absolute value of jerk to less than 2.5 m/s^3 can reduce overshoot more effectively than LQR. In fact, the jerk controlled by LQR exceeds 2.5 m/s^3. However, the followability controlled by LQR is clearly worse than RL's followability even though the jerk is almost same. Graph (a) shows that RL can converge d_e more rapidly than LQR for followability and control jerk to be less than 2.5 m/s^3. From these results, it seems that RL can consider the balance between followability and comfortability.

From these two perspectives, it can be concluded that RL can control with higher performance than LQR by balancing followability and controllability. The LQR is well known that it provides the optimal state feedback gain considering the infinite future prediction. However, it can work as the linear controller, so it is also known that it can work without considering the constraint. This cause deterioration of controller's performance because the saturation isn't a scenario which is considered by the LQR controller.

(See figure on next page.)

Fig. 6 Simulation results for RL (with scattered dead time)



(a) Error between the reference distance and measured distance

(b) Relative velocity

(c) Acceleration of the host vehicle

(d) Host vehicle jerk

Fig. 6 (See legend on previous page.)

Simulations with scattered initial preceding vehicle velocities

In this section, the RL agent's behavior is evaluated when the initial preceding vehicle velocity is scattered. The purpose of this confirmation is to check whether unexpected behaviors are caused if the initial preceding vehicle velocity is outside the learned conditions. ACC using RL was simulated when the initial preceding vehicle velocity was scattered from 15 to 55 m/s in 5 m/s intervals and the results are presented in Fig. 5. The other conditions are same as simulation conducted in "Simulations with one condition of preceding vehicle velocity and dead time" section.

In the simulation results, it appears that when the initial preceding vehicle velocity increases, followability performance decreases. The reason for this performance decrease is that the acceleration and deceleration of the host vehicle are saturated between 2 and -3 m/s^2, as defined in "Conditions for simulations" section. Therefore, even if RL controls with high performance by balancing followability and controllability, followability performance still decreases. From a comfortability perspective, even when the error between the reference distance and measured distance to preceding vehicle is large [e.g., when the initial preceding vehicle velocity is 45 m/s or 55 m/s in graph (a)], jerk is controlled under the threshold of 2.5 m/s^3 mostly. As a result, it seems that RL can control for various initial preceding vehicle velocities properly and there is no unexpected behavior (e.g., accelerating unexpectedly or crashing to preceding vehicle), even if the initial preceding vehicle velocity exceeds the learned conditions.

Simulations with scattered dead time variably

In this section, the RL agent's behavior is evaluated when the dead time is scattered. Similar to "Simulations with scattered initial preceding vehicle velocities" section, the purpose of this confirmation is to check whether unexpected behaviors occur if the dead time is not within the learned conditions. ACC using RL was simulated when the dead time in the controlled system was scattered from 0.02 to 0.20 s. The other conditions are same as simulation conducted in "Simulations with one condition of preceding vehicle velocity and dead time" section. The results are presented in Fig. 6.

In the simulation results, it appears that there is a very small impact on followability if the dead time is scattered over the learned range from 0.01 to 0.10 s. When the error between the reference distance and measured distance to preceding vehicle is overshot at approximately 2 s, the maximum gap is 1.0 m (3%). In contrast, from a comfortability perspective, more dead time increases jerk. When the jerk is overshot, the maximum value of jerk is 2.1 m/s^3. These gaps are caused by the balance of the tuned weights in the reward function for RL. It can be concluded that there is no excessive behavior in terms of followability, even if the dead time exceeds the learned conditions. Additionally, when the dead time is doubled from the maximum value in the learned conditions, the comfortability (value of jerk) is almost the same as the result achieved by LQR for comfortability with the dead time of 0.02 s. Additionally, after the second overshoot of the jerk at approximately 6 s, the ACC results controlled by RL are almost the same, even when the dead time is scattered.

Conclusions

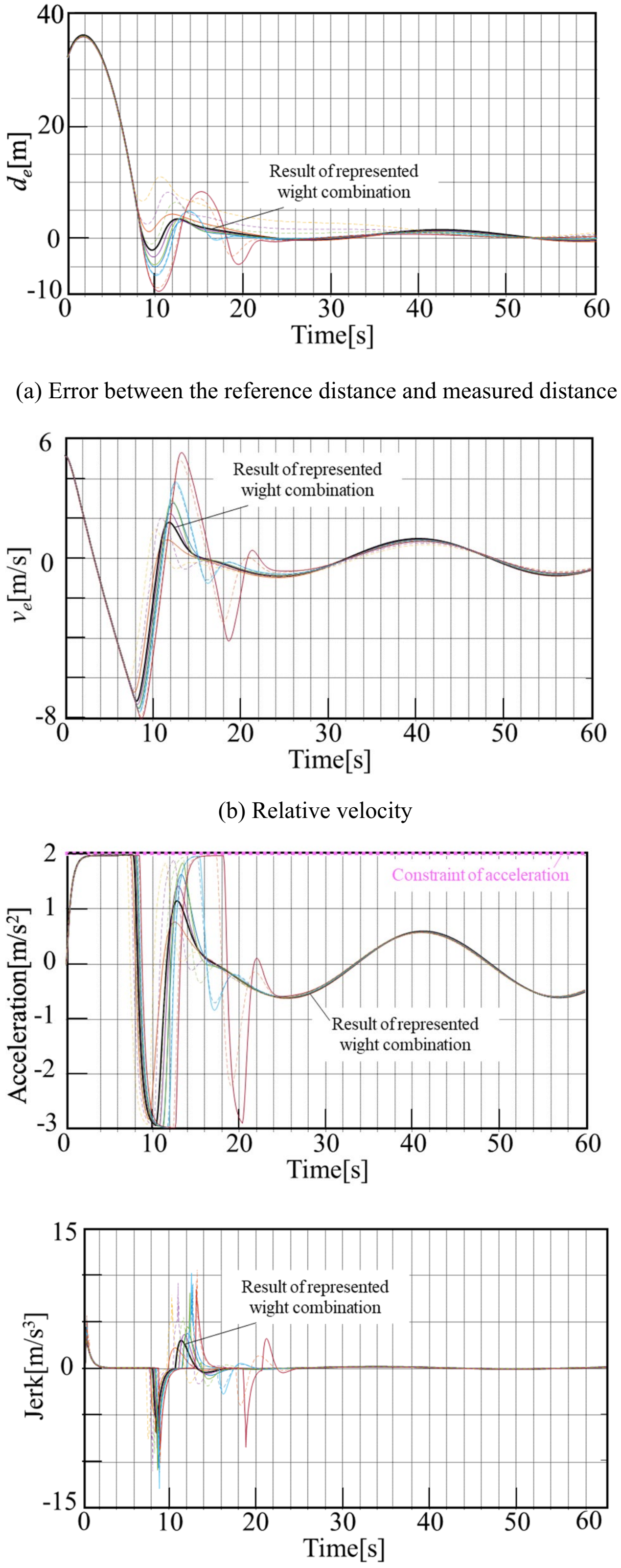
In this study, RL was applied to ACC and a unique reward function with reward value was defined to consider the balance between followability and comfortability.

To evaluate RL performance, the simulation of ACC was conducted using a trained RL agent and two types of LQR controllers (for followability and comfortability). The simulation results revealed that the LQR controllers can control ACC with high performance for either followability or comfortability through proper tuning. However, it is difficult for LQR to balance followability and comfort. In contrast, RL can balance followability and comfortability because RL considers the balance between followability and comfortability, as well as dead time. Additionally, when the initial preceding vehicle velocity and dead time in the controlled system were scattered, RL performance was evaluated. The simulation results confirmed that there is robustness to the initial preceding vehicle velocity and dead time in the controlled system. Based on these results, it can be concluded that the RL method with a unique threshold can ideally control followability and comfortability.

Appendix: Simulation results with the linear quadratic regulation weight combinations

As mentioned above, to decide LQR weight combinations, parameter studies are conducted. Here, some of simulation results with weight combinations are shown. The simulation results when the zero-order delay is 0.02 s and the other conditions are as discussed in "Conditions for simulations" section are presented. The Fig. 7 shows tuning results specialize in followability and Fig. 8 shows tuning results specialize in comfortability. The marked result

(See figure on next page.)
Fig. 7 Simulation results of tuning specialized in followability



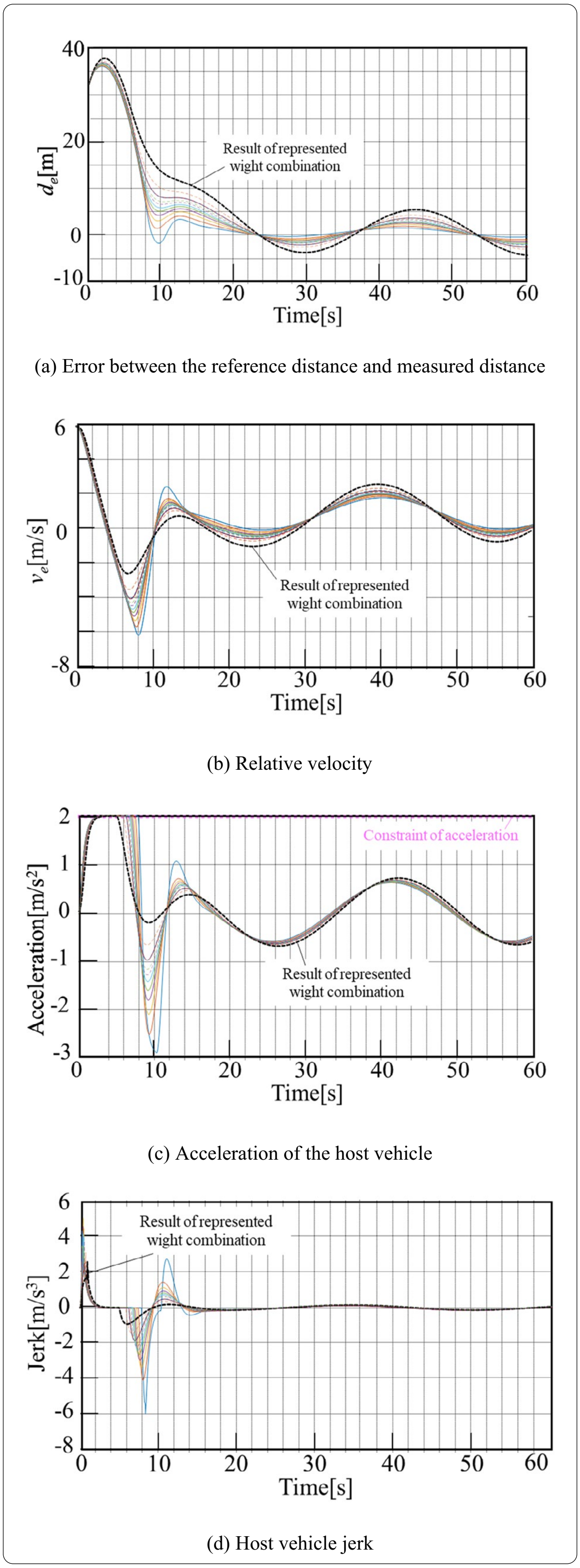
(a) Error between the reference distance and measured distance

(b) Relative velocity

(d) Host vehicle jerk

Fig. 7 (See legend on previous page.)

as “Result of represented weight combination” in each is chosen for comparison with RL above.



(a) Error between the reference distance and measured distance

(b) Relative velocity

(c) Acceleration of the host vehicle

(d) Host vehicle jerk

Fig. 8 Simulation results of tuning specialized in comfortability

Acknowledgements
Not applicable.

Author contributions
MN initiated this research, designed and simulated. MN analyzed the simulation data, interpreted the results and wrote the manuscript with the help and review from MH. Both authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
The datasets analyzed during the current study available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 12 December 2021 Accepted: 19 September 2022
Published online: 08 December 2022

References

1. Yinglong H, Biagio C, Quan Z, Michail M, Konstantinos M, Ji L, Ziyang L, Fuwu Y, Hongming X (2019) Adaptive cruise control strategies implemented on experimental vehicles a review. IFAC 52–55:21–27
2. Kumar R, Pathak R (2012) Adaptive cruise control—towards a safer driving experience. Int J Sci Eng Res 3:676–680
3. Bishop R (2005) Intelligent vehicle technology and trends. Artech House, Boston
4. Rajamani R (2006) Vehicle dynamics and control. Springer Science and Business Media, New York
5. Xiao L, Gao F (2010) A comprehensive review of the development of adaptive cruise control systems. Veh Syst Dyn 48:1167–1192. https://doi.org/10.1080/00423110903365910
6. Park C, Lee H (2017) A study of adaptive cruise control system to improve fuel efficiency. Int J Environ Pollut Rem 5:15–19. https://doi.org/10.11159/ijepr.2017.002
7. Wang F, Sagawa K, Inooka H (2000) A study of the relationship between the longitudinal acceleration/deceleration of automobiles and ride comfort. Ningenkougaku 36:191–200 (**In Japanese**)
8. Takahama T, Akasaka D (2018) Model predictive control approach to design practical adaptive cruise control for traffic jam. Int J Automot Eng 9(3):99–104
9. Zanon M, Frasch Janick V, Vukov M, Sager S, Diehl M (2014) Model predictive control of autonomous vehicles. Optimization and optimal control in automotive systems. Springer, Cham, pp 41–57
10. Luo L, Hong L, Li P, Wang H (2010) Model predictive control for adaptive cruise control with multi-objectives: comfort, fuel-economy, safety and car-following. J Zhejiang Univ Sci A 11:191–201

11. Maciejowski JM, Adachi S, Kanno M (2005) Predictive control with constraints (translated). Tokyo Denki University Press, Tokyo
12. Kondoh T, Yamamura T, Kitazaki S, Kuge N, Boer ER (2008) Identification of visual cues and quantification of drivers' perception of proximity risk to the lead vehicle in car-following situations. J Mech Syst Transp Logist 1:170–180. https://doi.org/10.1299/jmtl.1.170
13. Lillicrap TP, Hunt JJ, Alexander P, Nicolas H, Tom E, Yuval T, David S, Daan W (2015) Continuous control with deep reinforcement learning. arXiv:1509.02971
14. Mikami Y, Takahashi M, Nishimura H, Kubota M (2010) Adaptive cruise control in consideration of trade-off between capability of following a leading vehicle and suppression of fuel consumption. The Jpn Soc Mech Eng (C) 76:9–14 (**In Japanese**)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.