

RESEARCH ARTICLE

Open Access



# Control of mobile robot by switching traveling direction and control gain

Satoko Yamakawa<sup>1\*</sup> and Kenta Ebara<sup>2</sup>

## Abstract

The control method based on the time-state control form has been proposed to stabilize the chained system, which is a canonical-form nonlinear system. In this study, the control method is used for controlling a mobile robot in auto-parking situations. The proposed controller includes a parameter that is allowed to switch at arbitrary times without loss of the stability of the system. The robot employing the proposed controller reaches the target position by switching its traveling direction to avoid collisions with obstacles. However, the shape of the robot gives a problem. We resolve this by using the switchable parameter included in the proposed controller, and show the availability of switching the parameter. Furthermore, the appropriate switching of the traveling direction and the parameter enables the robot to reach the target faster. Thus, we search the appropriate values of the parameter and the switching points of the traveling direction using the genetic algorithm. In the auto-parking experiments that incorporate the search results, the robot can reach the target position faster.

**Keywords:** Time-state control form, Switching control, Mobile robot, Auto parking, Genetic algorithm

## Introduction

Rapid technological advances have led the automotive industry into adopting driver-assistance systems for safety, such as the collision-avoidance systems. In the future, more advanced systems for better driving and assistance are expected to be developed. Automatic parking systems are being researched as advanced support systems. For example, in some studies, the path plans for parking were calculated by using optimization methods when the obstacle maps were given [1–3].

Another approach for controlling the vehicle motion is the use of controller design methods based on kinematic models without pre-calculating the optimal paths for a given map. The kinematic models of four-wheeled cars, two-wheeled cars, truck trailers and so on can be represented in a chained form, which is a canonical form of nonlinear systems. The chained-form systems cannot be stabilized by a time-invariant smooth static controller [4]. Therefore, many controllers for chained systems have been proposed. Such controllers include time-varying

functions or non-smooth functions such as a sign function [5–12]. The control design method based on the time-state control form is also one such controller [13]. By using the time-state control form, the controller can be relatively easily designed based on the linear control theory. However, the controller requires the switching of the direction of the time axis. Because of the switching, the conditions for stabilizing the controlled system had not been exactly proven for a long time. Generally, the stability of the switched system depends on not only the characteristics of individual subsystems but also the timings for switching. It is known that the switched systems may be unstable by switching at inappropriate timings even if they are composed of stable linear subsystems [14, 15]. Therefore, for the switched systems, the stabilization conditions are mostly derived under some assumptions, such as the periodic switching, dwell time, and state-driven switching. We have already proved the stabilization conditions for the time-state control-form system with periodic switching [16]. However, if disturbances exist, the controller cannot switch its parameters periodically. Therefore, we exactly derived the stabilization conditions for the time-state control-form system with arbitrary switching by using the common Lyapunov

\*Correspondence: yamakawa@toyo.jp

<sup>1</sup> Department of Mechanical Engineering, Toyo University, Kujirai 2100, Kawagoe 350-8585, Japan

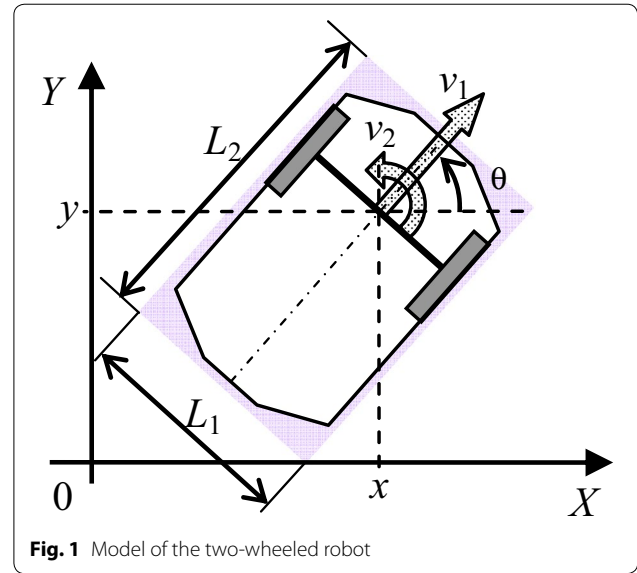
Full list of author information is available at the end of the article

function [17, 18]. To demonstrate the control method, we proposed a tracking controller for a wheeled inverted pendulum robot, which needs to suddenly switch its traveling direction for stabilizing the tilt angle of the robot [19].

Similarly to automobiles, robotic vehicles cannot move laterally. Therefore, the traveling direction of robots needs to be switched backward and forward when parking in a narrow space with obstacles. The controller based on the time-state control form, which switches the direction of the time axis, seems to be adequate to control vehicles that have to switch their traveling direction. Accordingly, we apply the control method to the auto-parking problem of a car-like robot. The controller proposed in this study involves the switching of not only the traveling direction but also a parameter value, which is allowed to switch to an arbitrary value at an arbitrary time while theoretically maintaining the stability of the system. Oyama and Nonaka [20] applied the predictive control method to the time-state control-form system to determine the switching points of the traveling direction under the restriction that the direction is switched only once. In this study, the robot measures the distance to obstacles and switches its traveling direction to avoid collisions as many times as necessary. Although the switches occur aperiodically depending on the shapes of the obstacles and the robot, the robot that employs the proposed controller can reach a given target position without precalculating the optimal paths for a given map. However, in numerical simulations, a problem arises because of the shape of the robot. We use the switchable parameter included in the proposed controller to solve this problem. Switching the parameter not only provides the solution but also enables the robot to reach the target position quickly. The time required to reach the target position also depends on the timing for switching the traveling direction. Consequently, a genetic algorithm is introduced to search the appropriate values of the parameter and the switching points of the traveling direction when the obstacle map is known. In the auto-parking experiments that incorporate the search results, the robot can reach the target position faster.

### Switching control law

The model of a wheeled robot considered in this paper is shown in Fig. 1. The robot has two independent driving wheels, and its translational velocity  $v_1$  and rotational velocity  $v_2$  can be controlled. The ground coordinate system is defined as shown in Fig. 1. The central position of the two wheels is denoted by the  $(x, y)$  coordinates, and the heading angle with respect to the  $x$ -axis is denoted by  $\theta$ . Assuming that the wheels do not slip on the ground, the kinematic model of the robot is represented by



**Fig. 1** Model of the two-wheeled robot

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v_2. \quad (1)$$

Consider applying the controller design based on the time-state control form to the system (1).

**Theorem** Assume that  $|\theta| < \pi/2$  in Eq. (1). Depending on the translational velocity  $v_1$ , the rotational velocity input  $v_2$  is given as

$$\begin{aligned} v_2 &= v_1 \mu \cos^3 \theta, \\ \mu &= -k_1 y - \operatorname{sgn}(v_1) \alpha k_2 \tan \theta, \end{aligned} \quad (2)$$

where  $\operatorname{sgn}(v_1)$  is the sign of  $v_1$ , and a control parameter  $\alpha$  is allowed to switch to an arbitrary positive value at an arbitrary time. Then,  $y$  and  $\theta$  converge to 0 under the arbitrary translational velocity  $v_1 \neq 0$  if the control gains  $k_1$  and  $k_2$  are positive.

*Proof* Please refer [Appendix](#).  $\square$

**Remark 1** The theorem guarantees the convergence of  $y$  and  $\theta$  as long as the robot continues to move backward and forward. In other words,  $y$  and  $\theta$  will not change if the time scale  $x$  does not change. In practical problems, the state  $x$  can be controlled so that  $x$  approaches 0 and stops changing after  $y$  and  $\theta$  become sufficiently small.

**Remark 2** The trajectories in the  $x$ - $y$  plane are independent of the absolute value of the translational velocity  $v_1$  because  $y$  and  $\theta$  change according to the state equation

(A10). Accordingly, the translational velocity  $v_1$  can be changed to accelerate or decelerate while keeping the trajectories of the robot as long as the switching points are not changed.

**Remark 3** When  $\alpha = 1$ ,  $\text{sgn}(v_1)$  does not change, and there are not disturbances, the convergence rate at which the trajectory of the robot approaches the  $x$ -axis is determined by the roots of a characteristic equation:

$$s^2 + k_2 s + k_1 = 0. \quad (3)$$

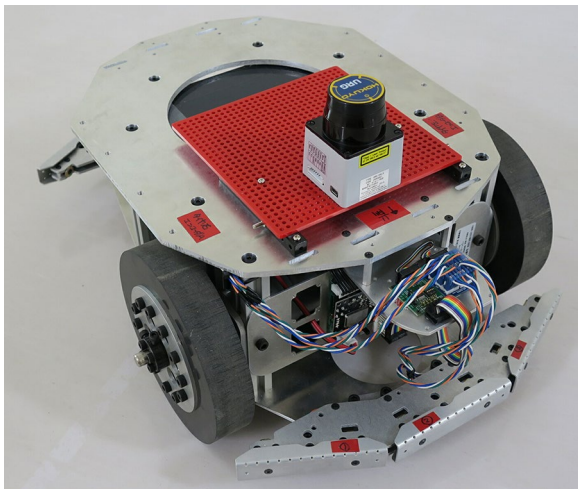
Thus, knowledge of the linear control theory can be used to decide the control gains  $k_1$  and  $k_2$ .

### Experimental setup

We used the robot shown in Fig. 2, which was made by Vstone Co. Ltd. The robot had the dimensions of  $483 \times 314 \times 166$  mm. Two driving wheels were collocated at the front of the robot and a passive caster was attached as a rear wheel. Encoders were attached to both driving wheels to measure the angles of the wheels' rotations. The position  $(x, y)$  and the angle  $\theta$  can be calculated from the measured data.

Equipment may have modeling errors or disturbances that are not included in the kinematic model given by Eq. (1), for example, the friction and the delay of motors. Thus, a proportional-integral controller (PI controller) is used to decrease the difference between the actual yaw rate  $V_2 (= \dot{\theta})$  calculated from the data measured by encoders and the control input  $v_2$  proposed in “Switching control law” section. The following control input is applied to the equipment instead of  $v_2$ .

$$v'_2 = k_p(v_2 - V_2) + k_I \int (v_2 - V_2) dt. \quad (4)$$



**Fig. 2** The wheeled robot used in the experiments

The gains  $k_p$  and  $k_I$  are chosen such that the error  $(v_2 - V_2)$  converges sufficiently fast. In this study, the gains were determined as  $k_p = 0.8$  and  $k_I = 16.0$  from the results of the step-response experiments. A Windows PC connected to the robot by wires calculates the control inputs from the sensor data; the PC also controls the two driving wheels of the robot.

Obstacles with vertical walls were located based on the auto-parking problems. A laser range finder URG-04LX made by Hokuyo Automatic Co. Ltd. was mounted on the robot and used to measure the distance from the robot to the obstacles. The measurable maximum distance of the sensor was 4.0 m, the angular resolution was  $0.36^\circ$ , and the sampling rate of the measurement was approximately 100 ms. As shown in Fig. 1, the rectangular area that includes the robot is defined. The width of the area  $L_1$  is 370 mm, and the length  $L_2$  is 540 mm. When the obstacles enter this area, the robot switches the traveling direction to avoid collisions.

The ground coordinate system is fixed so that the coordinate of a given target position is  $(x, y, \theta) = (0, 0, 0)$ . As mentioned in Remark 1, it is guaranteed that  $y$  and  $\theta$  converge to 0 as long as the robot continues to move using the controller (2). However, they take infinite time to completely converge. Therefore, the robot is stopped by assuming that the state converged, if  $|x| + \sqrt{y^2 + \tan^2 \theta} < 0.02$  is satisfied.

### Auto parking with switching

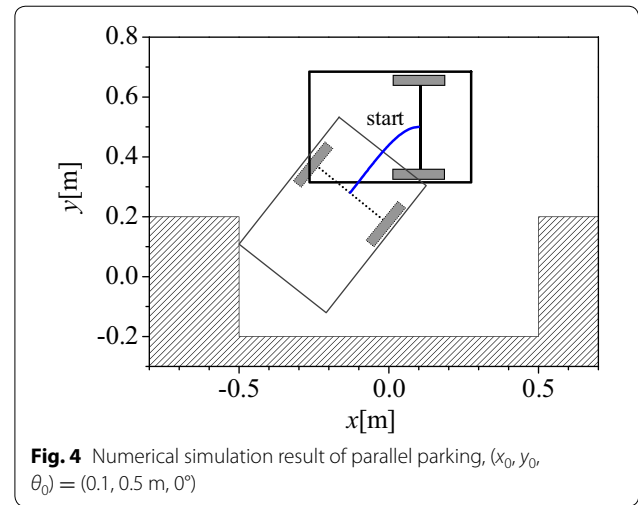
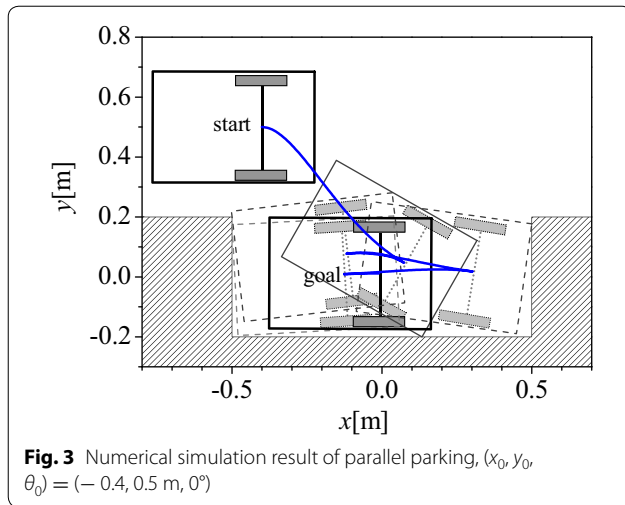
Applying the controller shown in “Switching control law” section to the robot described in “Experimental setup” section, we performed numerical simulations and experiments for auto parking with obstacles. The control gains were chosen as  $k_1 = 32$  and  $k_2 = 8$  so that the roots of Eq. (3) were assigned to the Butterworth pattern  $-4 \pm 4i$ . The traveling velocity was set as  $v_1 = 0.05$  m/s so that the wheels of the robot would not slip on the ground.

### Switching the parameter $\alpha$

Consider a parallel-parking problem. An obstacle is placed on the shaded portion in Fig. 3. The coordinates of the corners of the obstacle are  $(x, y) = (0.5, 0.2)$ ,  $(0.5, -0.2)$ ,  $(-0.5, -0.2)$ , and  $(-0.5, 0.2)$ .

### Results of numerical simulations

The initial state of the robot is  $(x_0, y_0, \theta_0) = (-0.4, 0.5 \text{ m}, 0^\circ)$ . The initial traveling velocity  $v_1$  is positive, that is, the robot moves forward at  $t = 0$ . When the parameter  $\alpha$  is set at 1, the trajectory of the robot in the horizontal plane is shown as a rigid line (see Fig. 3). At first, the robot moved forward. When its right front corner approached the obstacle at  $(x, y) = (0.074, 0.047)$ , the robot switched



its traveling direction to avoid collisions. Next, when the robot moved backward, its rear came close to the corner of the obstacle. Then, the traveling direction was switched again. The thin lines and dotted lines in Fig. 3 illustrate the area of the robot to avoid collisions when its direction was switched. The robot reached the target position after aperiodically switching the direction four times based on the shape of the robot and the obstacle.

Another example is the simulation result in the case that  $(x_0, y_0, \theta_0) = (0.1, 0.5 \text{ m}, 0^\circ)$  and  $v_1 < 0$  at  $t = 0$ , as shown in Fig. 4. In this case, the robot moved backward, and its left rear corner approached the obstacle at a shallow angle, that is, a large  $|\theta|$ . Then, the robot repeatedly switched its direction while staying at the same position; it could not move any further. This happened because the robot turned rapidly the moment the traveling direction was changed. As shown in Fig. 5, the distance between the body of the robot and the obstacles is different based on the heading angle. Thus, the robot repeatedly switched its traveling direction because of rapid turnings before it moved a sufficient distance away from the obstacles. This problem may occur if the vehicle is not circular in shape.

Therefore, we pay attention to the parameter  $\alpha$  included in the controller (2). The parameter  $\alpha$  is allowed to change its value arbitrarily, but its practical applications have hardly been discussed. In this study, the parameter  $\alpha$  was used for solving the problem based on the shape of the robot. For the same initial conditions as in Fig. 4,  $\alpha$  was switched to a small value ( $\alpha = 0.5$ ) when the traveling direction was switched the first time. The small parameter  $\alpha$  made the yaw rate  $\dot{\theta}$  small; therefore, repeated switching did not occur. As a result, the robot reached the target position as shown in Fig. 6. However, the small  $\alpha$  value tended to delay the convergence of  $\theta$  because  $\theta$  is multiplied by  $\alpha$  in Eq. (2). In the simulation result shown

in Fig. 6, the traveling direction was switched 19 times, and it took 115 s for the robot to reach the target position. Therefore, we introduced more switches to  $\alpha$ . After switching  $v_1$  a second time,  $\alpha$  was switched to 8. Furthermore,  $\alpha$  was returned to 1 after switching  $v_1$  a third time. The trajectory in the  $x$ - $y$  plane in this case is shown in Fig. 7. The time transitions of the states  $x$ ,  $y$ , and  $\theta$ , the traveling direction  $\text{sgn}(v_1)$ , the parameter  $\alpha$  and the rotational velocity input  $v_2$  are shown in Fig. 8. The control gain increased by the large  $\alpha$  made the state  $\theta$  approach 0 quickly. As a result, the robot reached the target position in 44 s.

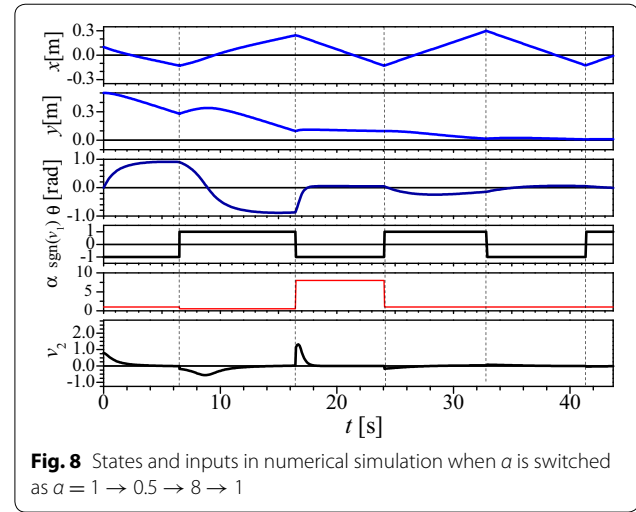
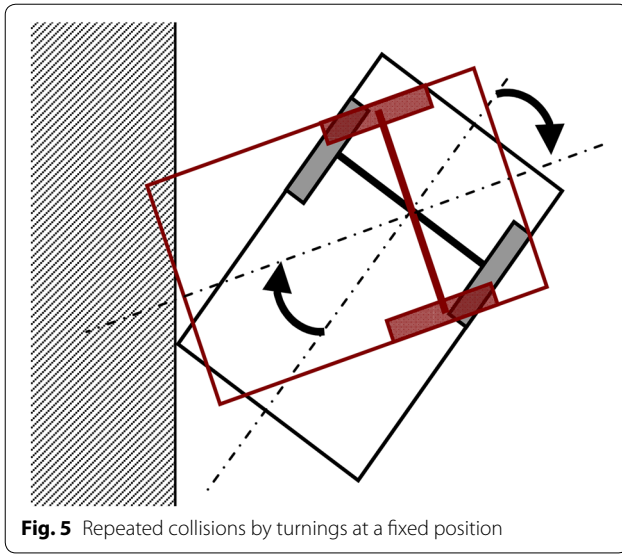
**Remark 4** Generally, it is not easy to guarantee the stability of the system that includes the switching of control gains. However, in the control method in this paper, the absolute value of one control gain can be switched by switching  $\alpha$  arbitrarily at any time while the stability of the system is guaranteed. The appropriate switching of the parameter  $\alpha$  can circumvent the problem associated with robots not being able to move because of their shapes, and enables the robot to reach the target position faster.

**Remark 5** The poles of both subsystems of the switched system (A10), that is, the roots of

$$s^2 + \alpha k_2 s + k_1 = 0, \quad (5)$$

are changed by  $\alpha$  as shown in Fig. 9. Both poles approach the imaginary axis when  $\alpha$  decreases, and the states will converge slowly. On the other hand, when  $\alpha$  increases, one root approaches negative infinity and the other root approaches 0. If  $\alpha$  is too large, not only will the input value  $v_2$  become large but also the convergence may be



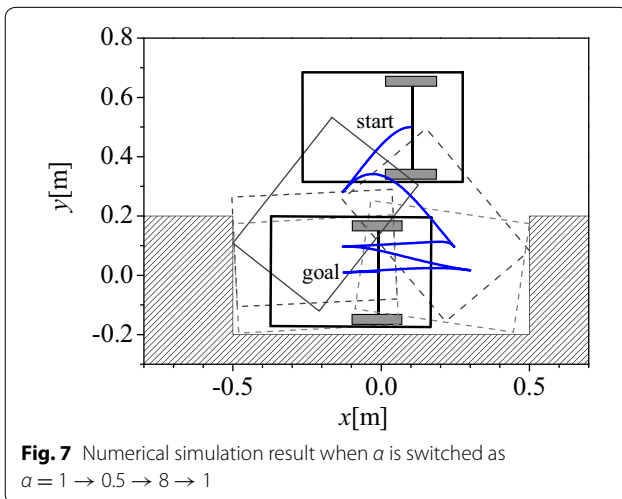
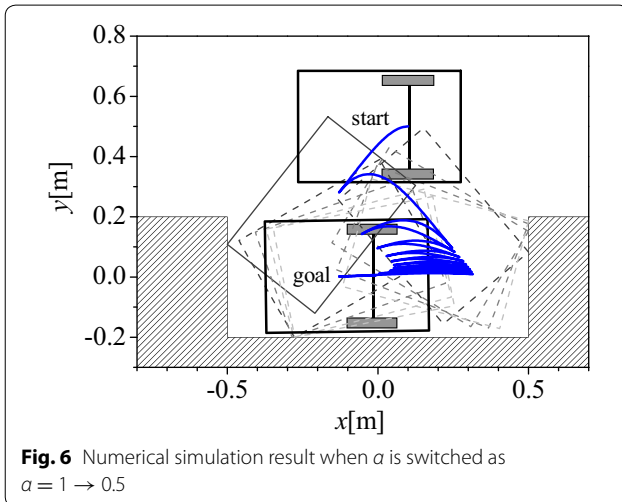


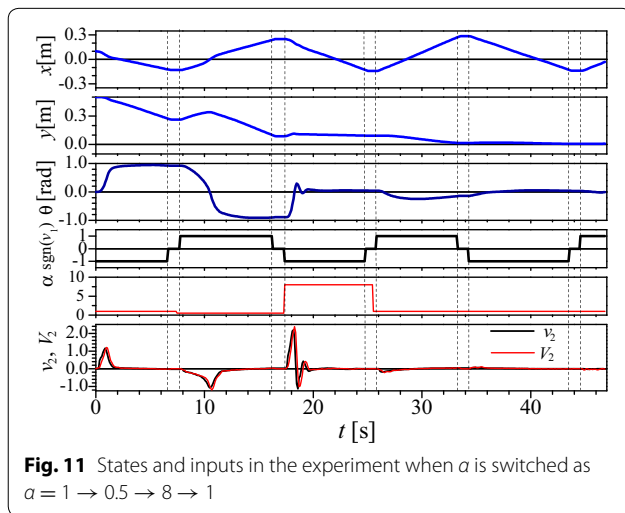
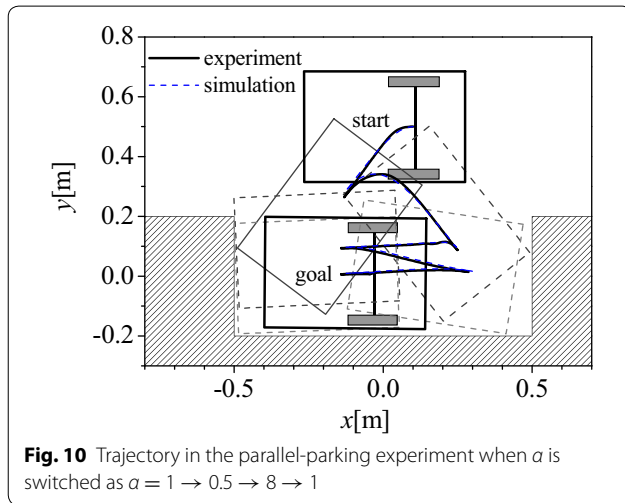
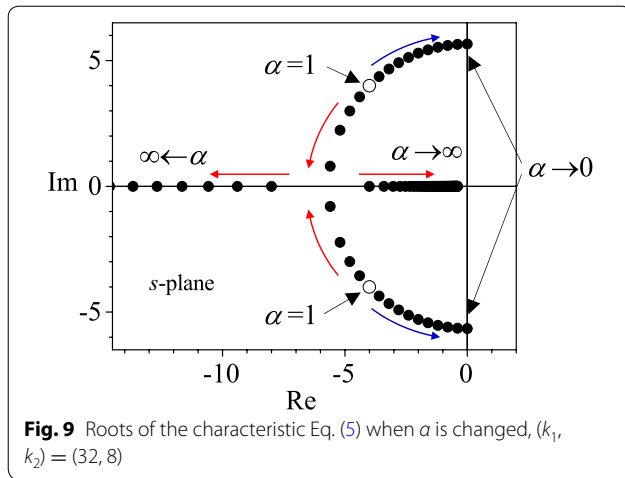
slow. Therefore, it is better to temporarily change  $\alpha$  to an appropriate value.

#### Experimental results

The black rigid line in Fig. 10 shows the trajectory in the experiments in which the switching conditions of  $\alpha$  and the initial conditions are the same (as in Fig. 7). The trajectory is almost the same as the simulation result in “Results of numerical simulations” section, which is shown again as a blue line in Fig. 10. The transitions of the state and the inputs are shown in Fig. 11. In the experiments, to cancel out the influence of inertia, the robot was stopped for a second before it started moving in the opposite direction after approaching the obstacle. In this case, it took a longer time, i.e., 47 s, than the simulation to satisfy the convergence conditions.

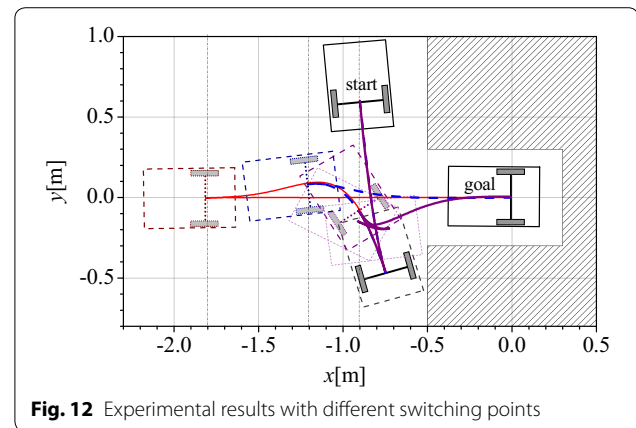
The trajectories of simulations and experiments are slightly different when the angular velocity is large. In the lowest graph of Fig. 11, the black and the red lines indicate the control input  $v_2$  and the actual yaw rate  $V_2$ . After the control input  $v_2$  increases rapidly,  $V_2$  delays, and the experimental trajectory deviates from the result of the simulation. Although the PI controller (4) was used in the experimental equipment to approximate  $V_2$  to  $v_2$ , it could not completely control the angular velocity so as to change rapidly, and some differences in the trajectory were generated. Furthermore, immediately after the second switching of the traveling direction, the angle  $\theta$  overshoot because of large  $v_2$ . However, the oscillation that resulted from the overshooting was restrained soon by the feedback controller (2). Naturally, the fourth switching point was not exactly the same as the simulation result because of these slight differences in trajectory. Nevertheless, the robot reached its target position.

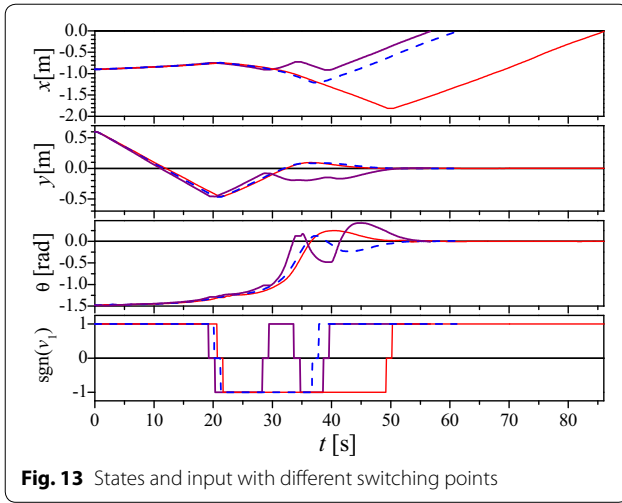




### Influences of switching points

Consider a situation in which the robot turns almost  $90^\circ$  before parking. An obstacle is set up in the shaded portion in Fig. 12. The coordinates of the corners of the obstacle are  $(-0.5, 0.3)$ ,  $(0.3, 0.3)$ ,  $(0.3, -0.3)$ , and  $(-0.5, -0.3)$ . The initial state of the robot is given as  $(x_0, y_0, \theta_0) = (-0.9, 0.6 \text{ m}, -85^\circ)$ , and the initial traveling velocity  $v_1$  is positive. In this example, after switching the traveling direction to avoid collisions, the robot has to switch its direction to forward again somewhere to reach the target position  $x = 0$ . Therefore, the traveling direction is switched at  $x = X_s$ . In the experiments, the position of  $X_s$  was set at  $-0.9$ ,  $-1.2$ , and  $-1.8$  m, and the trajectories were compared. In Figs. 12 and 13, the trajectories and the states are shown as a rigid purple line, a dotted blue line, and a narrow red line. In all the cases, the robot reached the target position after switching the traveling direction several times. However, the time taken to satisfy the convergence conditions was different in each case. In the case of  $X_s = -1.2$ , the required time was 61 s, but it was 86 s in the case of  $X_s = -1.8$ . An additional run and extra time were required when  $X_s$  was very far away from the obstacle. Conversely, if  $X_s$  had been too close to the obstacle, the number of times of switching the traveling direction would have increased. In the experiments, the traveling direction was switched four times in the case of  $X_s = -0.9$ . As shown in the simulation results (see Fig. 14), the narrow space causes frequent switches and long traveling distance before the convergence conditions are satisfied. Furthermore, in practice, switching frequently may require a longer time than the time required for traveling the distance because it is time-consuming to accelerate and decelerate to switch the traveling direction. From the aforementioned results, like the parameter  $\alpha$ , the switching point is one of the factors that affect the time required to reach the target position.





**Fig. 13** States and input with different switching points

### Search of the control parameter and switching points

As mentioned in “Auto parking with switching” section, there is a possibility of fast convergence by appropriately choosing the control parameter  $\alpha$  and the switching points  $X_s$ . Thus, we consider searching for  $\alpha$  and  $X_s$  so that the robot can reach the target position in a shorter time when the obstacle map is known in advance. The time required to reach the target varies discontinuously according to the number of times of switching the direction as shown in Fig. 14; therefore, a genetic algorithm, which is one of heuristic approaches, is used to search for better solutions.

### Application of the genetic algorithm

In this study, a traditional genetic algorithm is used. At first,  $n$  candidate solutions are prepared in the first generation population. Each candidate solution is represented as a string of 1s and 0s which length is  $m$ . The fitness function  $J$  is evaluated for each solution. Next, based on the evaluation results, a new generation population of solutions is generated using three genetic operations: selection, crossover, and mutation. To search for better solutions, this process is repeated until the number of generations reaches  $N$  [21].

We now reconsider the auto-parking problem described in “Influences of switching points” section. The parameter  $\alpha$  is given as 1 at  $t = 0$ . The parameter  $\alpha$  is switched to  $\alpha_1$  and  $\alpha_2$  after the first and second switches in the traveling direction, respectively. The search range of  $\alpha_i$ ,  $i = 1, 2$ , is set as  $0 < \alpha_i \leq \alpha_{\max}$ . The switching point  $X_s$  is searched in the range of  $X_{s\min} \leq X_s \leq X_{s\max}$ . Let the number of elements of the candidate solutions  $m$  be 24; we divide the candidate solution into three 8-bit numbers  $\xi_1$ ,  $\xi_2$ , and  $\xi_3$ . Then,  $X_s$ ,  $\alpha_1$ , and  $\alpha_2$  are defined as

$$X_s = X_{s\min} + \frac{\xi_1}{255}(X_{s\max} - X_{s\min}), \quad (6)$$

$$\alpha_1 = \frac{(\xi_2 + 1)}{256}\alpha_{\max}, \text{ and } \alpha_2 = \frac{(\xi_3 + 1)}{256}\alpha_{\max}. \quad (7)$$

For each solution, the trajectory of the robot with a given initial condition is simulated numerically. The simulation is terminated when the convergence conditions mentioned in “Experimental setup” section are satisfied or when the robot does not reach the target position within 200 s. By using the states  $x_e$ ,  $y_e$ , and  $\theta_e$  and the time  $t_e$  when the simulation is terminated, the fitness function is defined as

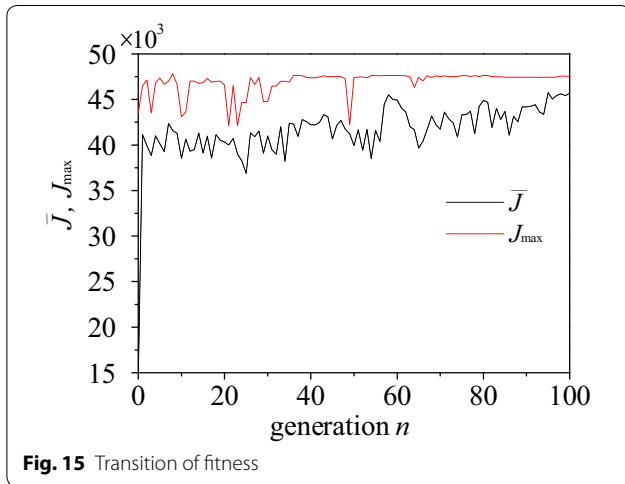
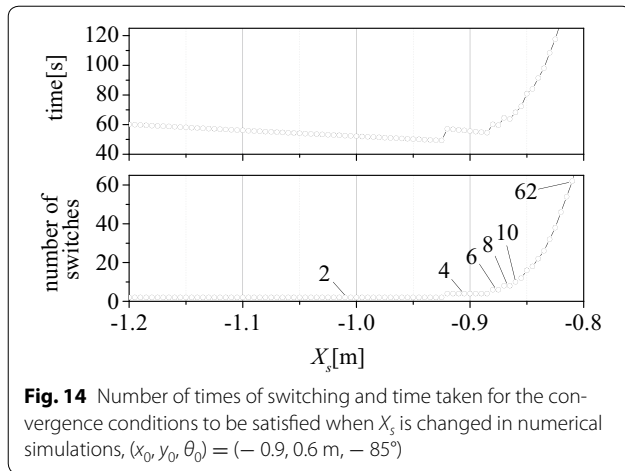
$$J = 50,000 - (x_e^2 + y_e^2 + \tan^2 \theta_e + t_e^2). \quad (8)$$

A large value of  $J$  indicates fast convergence, and the genetic algorithm searches the solutions that indicate large fitness values. To eliminate the situation that the robot stops before reaching the target position (as shown in Fig. 4), the searching program assumes that the robot cannot move any further and terminates the simulation if the number of switching operations exceeds 10. In this case, the fitness function for that solution is assigned the value 0.

### Search results

Let the number of candidate solutions  $n$  be 20, and the terminal generation number  $N$  be 100. We set the search ranges at  $X_{s\min} = -1.2$ ,  $X_{s\max} = -0.6$ , and  $\alpha_{\max} = 10$ . The parameters were searched for the case in which the initial conditions and obstacles were the same as in “Influences of switching points” section. As a result, the average fitness of 20 solutions  $\bar{J}$  and the maximum fitness  $J_{\max}$  changed with the generation, as shown in Fig. 15. The average fitness increased while oscillating. The genetic algorithm does not guarantee an optimal solution, but it can provide several better solutions. In this case, several solutions having high-level fitness were found, for example  $(X_s, \alpha_1, \alpha_2) = (-1.0, 4.57, 1.25)$  and  $(X_s, \alpha_1, \alpha_2) = (-1.0, 0.82, 3.28)$ .

We also searched the parameters for other initial conditions in the range of  $-85^\circ < \theta_0 < +85^\circ$  and  $-1.50 \text{ m} < x_0 < -0.70 \text{ m}$ . When there was sufficiently wide space for the robot to move freely,  $X_s$  became large enough to minimize the number of switching operations and  $\alpha$  tended to be large in order to fast converge. The characteristics of trajectory varied depending on which corners of the robot and the obstacles approached and how the states of the robot were when the collision occurred. Several characteristic trajectories are shown in Fig. 16.



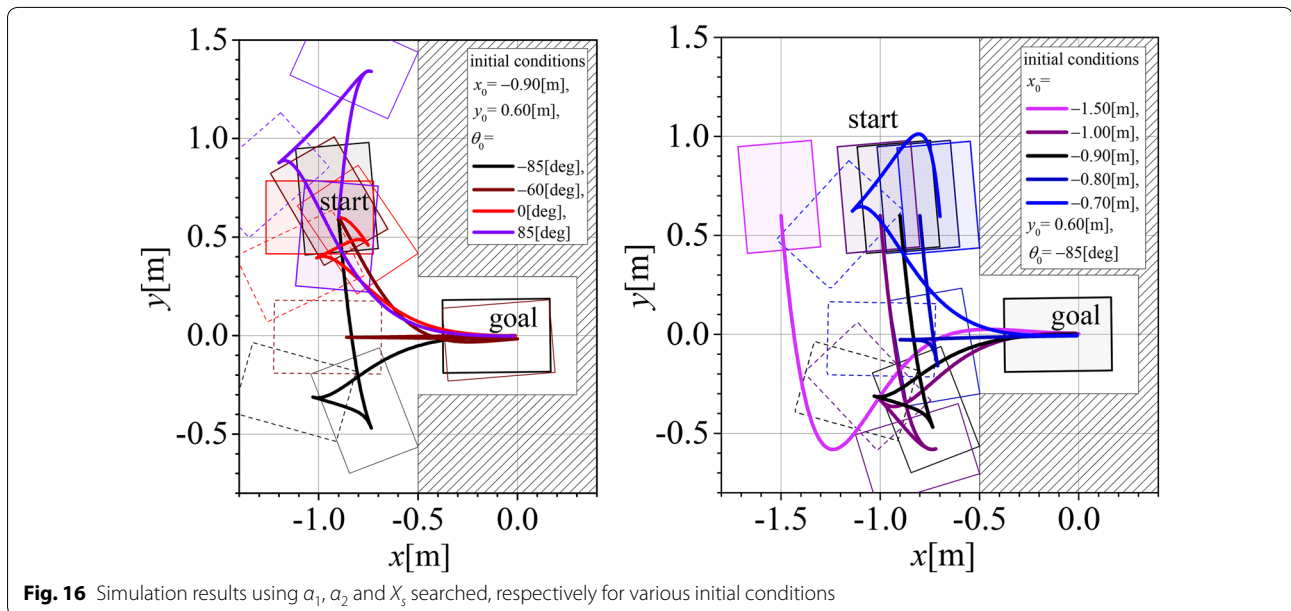
### Experimental results with searched parameters

The results of the auto-parking experiment that incorporates the searched parameters  $(X_s, \alpha_1, \alpha_2) = (-1.0, 4.57, 1.25)$  are shown as a rigid line in Fig. 17. The large value of  $\alpha$  after the first switch rapidly made the heading angle  $\theta$  close to 0. Then, after the traveling direction was switched again at  $X_s = -1.0$ , the parameter  $\alpha$  was switched to the small value of 1.25, and the robot reached the target position by switching only twice. By using the parameter searched by the genetic algorithm, the robot reached the target position in 48 s, which is less than the time taken in the experiments in “Influences of switching points” section (see Fig. 13) in which  $\alpha$  is fixed as 1.

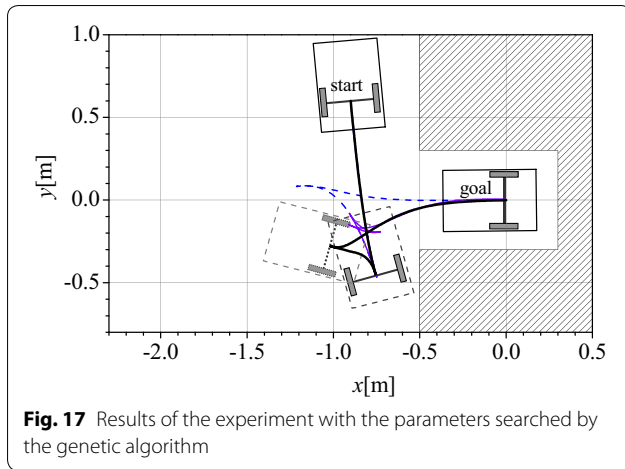
**Remark 6** In this paper, the value of  $\alpha$  was switched at the same switching points as the traveling direction for simplicity. When the parameter  $\alpha$  is allowed to switch at arbitrary timings, better solutions may be found by searching the switching points of  $\alpha$  simultaneously although the time required for searching may be long.

### Conclusions

We applied the control method based on the time-state control form to control a mobile robot in auto-parking situations. The proposed controller includes the parameter  $\alpha$ , which is allowed to switch at arbitrary timings without loss of the stability of the system. The robot that employed the proposed nonlinear controller switched its traveling direction automatically to avoid collisions with obstacles and reached the target position. However, the shape of the robot gave problems. We used the







switchable parameter  $\alpha$  to solve the problem and showed the availability of switching the parameter. Furthermore, by appropriately switching the parameter  $\alpha$  and the traveling direction, the robot could reach the target faster in the simulations and experiments. In this method, only one parameter is temporarily changed and complicated recalculations such as path optimizations are not required. When the obstacle map was known in advance, the appropriate values of the parameter and the switching points of the traveling direction were searched using the genetic algorithm. In the auto-parking experiments that used the search results, the robot reached the target position faster.

#### Authors' contributions

SY proposed the controller, proved the stability of the controlled system theoretically, conducted the numerical simulations, decided the experimental setting, and drafted the manuscript. KE proposed the use of the genetic algorithm and conducted all experiments using the equipment. Both authors read and approved the final manuscript.

#### Author details

<sup>1</sup> Department of Mechanical Engineering, Toyo University, Kujirai 2100, Kawagoe 350-8585, Japan. <sup>2</sup> Graduate School of Science and Engineering, Toyo University, Kujirai 2100, Kawagoe 350-8585, Japan.

#### Acknowledgements

This work was partially supported by JSPS KAKENHI Grant Number JP17K01093.

#### Competing interests

The authors declare that they have no competing interests.

#### Ethics approval and consent to participate

Not applicable.

#### Publisher's Note

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

## Appendix

In the method based on the time-state control form, a chained system is transformed into two linear systems: the state-control part, which is an  $n$ th order linear system, and the time-control part, which is a first-order linear system. We generally provided the conditions for asymptotically converging the state of the  $n$ th order linear system under arbitrary switching of the time axis [18]. The two-wheeled car version of the proof is given below.

Under the assumption that  $|\theta| < \pi/2$ , we use the non-linear coordinate transformation

$$\mathbf{z} = (y, \tan \theta)^T \quad (\text{A1})$$

and the input transformation

$$\mu = \frac{v_2}{v_1 \cos^3 \theta} \quad (\text{A2})$$

for Eq. (1). The state  $x$  increases with respect to time  $t$  when  $v_1 > 0$ ; therefore, the state  $x$  can be regarded as an alternative time scale. Since the derivative of  $y$  with respect to  $x$  is

$$\begin{aligned} \frac{dy}{dx} &= \frac{dy}{dt} \frac{dt}{dx} \\ &= \tan \theta, \end{aligned} \quad (\text{A3})$$

the state equation of  $\mathbf{z}$  with the time scale  $x$  is obtained as

$$\frac{d}{dx} \mathbf{z} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mu. \quad (\text{A4})$$

The state  $x$  is controlled by the input  $v_1$ .

$$\frac{dx}{dt} = v_1 \cos \theta \quad (\text{A5})$$

Eq. (A4) is the controllable canonical form. A state feedback controller for stabilizing the system given by Eq. (A4) can be easily designed using the linear control theory. When  $\mathbf{z} \rightarrow \mathbf{0}$  as  $x \rightarrow \infty$  by using the stabilizing feedback controller,  $\theta$  also converges to 0 because of the assumption.

When  $v_1 < 0$ ,  $x$  decreases with respect to the actual time  $t$ . Therefore, we have to introduce another time scale  $x' = -x$  which increases with respect to  $t$  for analyzing the stability of the system. The state equation of  $\mathbf{z}$  with  $x'$  is obtained as

$$\frac{d}{dx'} \mathbf{z} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \mu. \quad (\text{A6})$$

Now, a new time scale  $\tau$  is defined as

$$\tau = \int_0^t \left| \frac{dx}{dt} \right| dt. \quad (\text{A7})$$

The time scale  $\tau$  is the distance traveled along the  $x$ -axis. The two systems (A4) and (A6) are described as a switched system by using the time scale  $\tau$  [17].

$$\frac{d}{d\tau} \mathbf{z} = \begin{cases} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mu & (v_1 > 0) \\ \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \mu & (v_1 < 0) \end{cases} \quad (\text{A8})$$

The input function  $\mu$  in the controller (2) can be rewritten as

$$\mu = \begin{cases} - \begin{bmatrix} k_1 & \alpha k_2 \end{bmatrix} \mathbf{z} & (v_1 > 0) \\ - \begin{bmatrix} k_1 & -\alpha k_2 \end{bmatrix} \mathbf{z} & (v_1 < 0) \end{cases}. \quad (\text{A9})$$

By substituting (A9) into (A8), the closed-loop system becomes

$$\frac{d}{d\tau} \mathbf{z} = \begin{cases} \tilde{\mathbf{A}} \mathbf{z} & (v_1 > 0) \\ \mathbf{E}_2 \tilde{\mathbf{A}} \mathbf{E}_2 \mathbf{z} & (v_1 < 0) \end{cases}, \quad (\text{A10})$$

where

$$\tilde{\mathbf{A}} = \begin{bmatrix} 0 & 1 \\ -k_1 & -\alpha k_2 \end{bmatrix}, \quad \mathbf{E}_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (\text{A11})$$

Now, consider a Lyapunov candidate function as

$$V = \mathbf{z}^T \mathbf{P} \mathbf{z}, \quad \mathbf{P} = \begin{bmatrix} k_1 k_2 & 0 \\ 0 & k_2 \end{bmatrix}. \quad (\text{A12})$$

If  $k_1$  and  $k_2 > 0$ ,  $V > 0$  ( $\mathbf{z} \neq \mathbf{0}$ ). The derivative of  $V$  with respect to  $\tau$  for both  $v_1 > 0$  and  $v_1 < 0$  is calculated as follows:

$$\begin{aligned} \frac{d}{d\tau} V &= \begin{cases} \mathbf{z}^T (\tilde{\mathbf{A}}^T \mathbf{P} + \mathbf{P} \tilde{\mathbf{A}}) \mathbf{z} & (v_1 > 0) \\ \mathbf{z}^T (\mathbf{E}_2 \tilde{\mathbf{A}}^T \mathbf{E}_2 \mathbf{P} + \mathbf{P} \mathbf{E}_2 \tilde{\mathbf{A}} \mathbf{E}_2) \mathbf{z} & (v_1 < 0) \end{cases} \\ &= \mathbf{z}^T \begin{bmatrix} 0 & 0 \\ 0 & -2\alpha k_2^2 \end{bmatrix} \mathbf{z} \\ &= -\alpha \mathbf{z}^T \mathbf{c}^T \mathbf{c} \mathbf{z}, \end{aligned} \quad (\text{A13})$$

where

$$\mathbf{c} = [0 \quad \sqrt{2} k_2]. \quad (\text{A14})$$

From the condition  $\alpha > 0$ , we have  $dV/d\tau \leq 0$ . Furthermore, both  $(\mathbf{c}, \tilde{\mathbf{A}})$  and  $(\mathbf{c}, \mathbf{E}_2 \tilde{\mathbf{A}} \mathbf{E}_2)$  are observable pairs; therefore, it is guaranteed that  $\mathbf{z} \rightarrow \mathbf{0}$  as  $\tau \rightarrow \infty$  under the arbitrary switching of  $\alpha$  and the sign of  $v_1$  by Lyapunov's stability theory.

Received: 20 March 2017 Accepted: 6 December 2017

Published online: 19 December 2017

## References

- Okawa K, Mogaki A (2011) Path planning and consecutive operation based on discrete acquired motion data for auto parking system. *J Robot Soc Japan* 29(4):376–383 **(in Japanese)**
- Nakanishi H, Fujiwara N (2000) Automatic parking system based on weighted graphs. *Trans JSME* 66(648):2638–2645 **(in Japanese)**
- Li B, Wang K, Shao Z (2016) Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach. *IEEE Trans Intell Transp Syst* 17(11):3263–3274
- Brockett RW (1983) Asymptotic stability and feedback stabilization, in: *differential geometric control theory*. Birkhäuser, Basel
- Murray RM, Sastry SS (1991) Steering nonholonomic systems in chained form. In: *Proceedings of the 30th IEEE conference on decision and control*, vol 2, pp 1121–1126
- Pomet JB (1992) Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift. *Syst Control Lett* 18:147–158
- Sordalen OJ, Wichlund KY (1993) Exponential stabilization of a car with  $n$  trailers. In: *Proceedings of the 32nd IEEE conference on decision and control*, vol 2, pp 978–983
- Samson C (1995) Control of chained systems application to path following and time-varying point-stabilization of mobile robots. *IEEE Trans Autom Control* 40(1):64–77
- Sordalen OJ, Egeland O (1995) Exponential stabilization of nonholonomic chained systems. *IEEE Trans Autom Control* 40(1):35–49
- Marchand N, Alami M (2003) Discontinuous exponential stabilization of chained form systems. *Automatica* 39:343–348
- Qu Z, Wang J, Plaisted C, Hull R (2006) Global stabilizing near-optimal control design for nonholonomic chained systems. *IEEE Trans Autom Control* 51(9):1440–1456
- Liu KZ, Dao MQ, Inoue T (2006) Exponentially e-convergent control algorithm for chained systems and its application to automatic parking systems. *IEEE Trans Control Syst Technol* 14(6):1113–1125
- Sampei M (1994) A control strategy for a class of non-holonomic systems: time-state control form and its application. In: *Proceedings of the 33rd conference on decision and control*, pp 1120–1121
- Liberzon D (2003) *Switching in systems and control*. Birkhäuser, Basel
- Hespanha JP (2004) Uniform stability of switched linear systems: extensions of LaSalle's Invariance Principle. *IEEE Trans Autom Control* 49(4):470–482
- Fujimoto F, Yamakawa S, Funahashi Y (2000) Switching control based on time-state control form. *Trans Soc Instrum Control Eng* 36(6):512–518 **(in Japanese)**
- Yamakawa S (2013) Stabilization conditions for chained systems based on time-state control form. *Trans Soc Instrum Control Eng* 49(5):499–504 **(in Japanese)**
- Yamakawa S (2014) Conditions of gain switching for stabilization control based on time-state control form. *Trans Soc Instrum Control Eng* 50(12):845–851 **(in Japanese)**
- Yamakawa S (2011) Tracking control of wheeled inverted pendulum based on the time-state control form. In: *The 18th IFAC world congress*, ThA20/6
- Oyama K, Nonaka K (2014) Model predictive parking control with obstacle avoidance considering automatic tuning of switching point. *Trans Soc Instrum Control Eng* 50–1:9–17 **(in Japanese)**
- Kodaka T (2011) *Machine learning for the first time*. Ohmsha, Ltd, Tokyo **(in Japanese)**