



Hybrid Control of Trajectory Planning for Desired Trajectory and Collision Avoidance Based on Optimization Problem for a Wheeled Mobile Robot

Kobayashi, Masato

Motoi, Naoki

(Citation)

IEEJ Journal of Industry Applications, 9(4):331-340

(Issue Date)

2020

(Resource Type)

journal article

(Version)

Version of Record

(Rights)

© 2020 The Institute of Electrical Engineers of Japan

(URL)

<https://hdl.handle.net/20.500.14094/90007925>



Hybrid Control of Trajectory Planning for Desired Trajectory and Collision Avoidance Based on Optimization Problem for a Wheeled Mobile Robot

Masato Kobayashi* Student Member, Naoki Motoi* Senior Member

(Manuscript received Jan. 22, 2019, revised Oct. 15, 2019)

This paper proposes a hybrid control of trajectory planning for a desired trajectory and collision avoidance based on an optimization problem for a wheeled mobile robot. In the proposed method, the priority of trajectory planning for the desired trajectory and collision avoidance is modified using the priority variable α . The priority variable α is designed by solving the optimization problem with constraints. As a constraint condition, the range of the priority variable α where the robot does not collide with multiple obstacles is derived. Under this constraint condition, the priority variable α with the highest priority of trajectory planning for the desired trajectory is selected as the optimal value. As a result, the velocity commands considering trajectory planning for the desired trajectory and collision avoidance are calculated in real time. The effectiveness of the proposed method is confirmed via experimental results.

Keywords: motion control, robotics, trajectory tracking, collision avoidance, mobile robot

1. Introduction

Recent years, many researches on mobile robots have been reported^{(1)–(3)}. The control methods of mobile robots are roughly classified into two types. One is a remote control, and the other is an autonomous control. In the remote control, several tasks are performed by using the judgment of the operator. On the other hand, in the autonomous control, the movement is performed by using several sensor information. The autonomous control has advantages, since the operator is not necessary. Therefore, this paper focuses on the autonomous control for mobile robots. The autonomous control consists of the localization, map building, trajectory planning, and trajectory tracking. Simultaneous localization and mapping (SLAM) is a useful method that simultaneously generates the localization and the mapping in real time^{(4)–(6)}. By using SLAM, the robot reaches the destination while avoiding obstacles.

This paper focuses on trajectory planning. Many researches of trajectory planning have been reported. One of useful local trajectory planning methods is the potential method⁽⁹⁾. By using this method, the trajectory is generated by using the gradient vector with the goal point as the minimum potential. Even if there are obstacles, the robot can avoid obstacles by using the repulsive potential field. Also, by making the no-entry area for the static obstacle, the collision avoidance for static obstacles is guaranteed. However, in the potential method, the robot motion with nonholonomic constrain is not considered, because the trajectory is determined by the potential field. As a result, this method cannot generate the shortest path from start to goal

because this method does not consider the motion with the nonholonomic. Fox et al. developed the dynamic window approach (DWA)⁽¹⁰⁾. DWA is one of the local trajectory planning methods. By using DWA, the robot avoids obstacles and follows the desired trajectory in the static/dynamic environment. However, there are no guarantees where the robot avoids obstacle and follows the trajectory, because DWA generates the velocity commands by the sum of the function with the weight coefficients. As a result, this method may not navigate robots from start to goal with obstacles. Yamazaki and Inaba developed the collision avoidance using virtual manipulators⁽¹¹⁾⁽¹²⁾. This method generates virtual manipulators between the mobile robot and obstacles. The robot can achieve start/goal navigation with less calculation cost. However, sometimes the robot can not reach the goal position, because this method generates the trajectory in the velocity dimension.

The development of the local trajectory planning, which performs trajectory planning online for the static and dynamic collision avoidance, is very important. In the local trajectory planning, it is required to develop the method that takes the robot motion with nonholonomic constrain into consideration and guarantees collision avoidance. Therefore, this paper proposes the local trajectory planning method that considers the robot motion with nonholonomic constrain and guarantees collision avoidance of static obstacles. The proposed method is the hybrid control and the optimization problem with constrained. This method selects the trajectory that follows the most desired trajectory with no collision. The effectiveness of the proposed method was confirmed from the experimental results.

This paper is organized as follows. Section 2 describes the modeling of the mobile robot. Section 3 describes one of trajectory planning methods for the desired trajectory.

* Graduate School of Maritime Science, Kobe University
5-1-1, Fukaeminami, Higashinada-ku, Kobe 658-0022, Japan

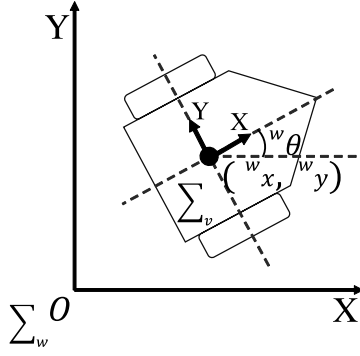


Fig. 1. Mobile Robot and Its Coordinate System

Section 4 describes one of collision avoidance control methods. Section 5 proposes the hybrid control of trajectory planning for the desired trajectory and collision avoidance. Section 6 shows experimental results to confirm the effectiveness of the proposed method. Finally, this paper is concluded in section 7.

2. Modeling of Mobile Robot

This section shows the modeling of the mobile robot. Figure 1 shows the coordinate systems of the mobile robot. The world coordinate system Σ_w and the robot coordinate system Σ_v are defined. The subscript $^w\circ$ means the value in the world coordinate system. There is no superscript for the variable in the robot coordinate system. The center point of the two wheels is set as the origin in the robot coordinate system Σ_v . The X-axis in the robot coordinate system is aligned with the traveling direction of the mobile robot. The Y-axis in the robot coordinate is defined as the vertical left direction of X-axis. As shown in Fig. 1, $(^wx, ^wy)$ means the mobile robot position in the world coordinate system. The angle $^w\theta$ is defined as the angle between the X-axis in the robot coordinate system and the X-axis in the world coordinate system. The translational velocity v and the angular velocity ω of the mobile robot are expressed as follows.

$$v = (v_R + v_L)/2 \quad (1)$$

$$\omega = (v_R - v_L)/2B \quad (2)$$

where v_R and v_L mean the right wheel velocity and left wheel velocity. B means the distance from the center to the wheel. The velocity in the world coordinate system $^w\dot{x}$, $^w\dot{y}$ and $^w\dot{\theta}$ are derived from the translational velocity v and the angular velocity ω as follows.

$$^w\dot{x} = v \cos ^w\theta \quad (3)$$

$$^w\dot{y} = v \sin ^w\theta \quad (4)$$

$$^w\dot{\theta} = \omega \quad (5)$$

The mobile robot position $(^wx, ^wy)$ and angle $^w\theta$ after t seconds are calculated as follows.

$$^wx = \int_0^t v \cdot \cos ^w\theta dt \quad (6)$$

$$^wy = \int_0^t v \cdot \sin ^w\theta dt \quad (7)$$

$$^w\theta = \int_0^t \omega dt \quad (8)$$

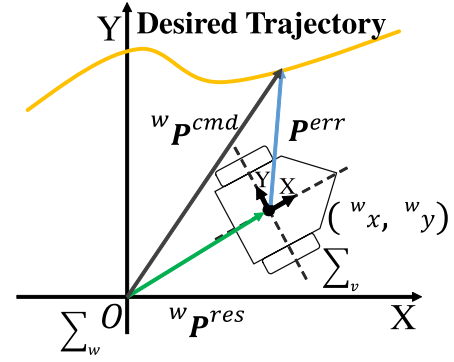


Fig. 2. Kinematic Relation between Desired Trajectory and Mobile Robot

3. Trajectory Planning for Desired Trajectory

As one of trajectory planning for the desired trajectory, this section shows trajectory planning proposed by Kanayama *et al.*⁽⁷⁾⁽⁸⁾. This method uses the time function of the position and angle as the desired trajectory. In addition, the tracking performance is guaranteed by Lyapunov function.

Figure 2 shows the kinematic relation between the desired trajectory and the mobile robot. The subscript \circ^{cmd} and \circ^{res} mean the command value and the response value. \circ^{traj} stands for the desired trajectory value. \circ^{err} represents for the difference between the command value and the response value.

The error value $\mathbf{P}^{err} = [x^{err}, y^{err}, \theta^{err}]^T$ is calculated between the desired trajectory value $^w\mathbf{P}^{traj} = [^wx^{traj}, ^wy^{traj}, ^w\theta^{traj}]^T$ and the response value $^w\mathbf{P}^{res} = [^wx^{res}, ^wy^{res}, ^w\theta^{res}]^T$.

$$\begin{aligned} \mathbf{P}^{err} &= [x^{err} \quad y^{err} \quad \theta^{err}]^T \\ &= \begin{bmatrix} \cos ^w\theta^{res} & \sin ^w\theta^{res} & 0 \\ -\sin ^w\theta^{res} & \cos ^w\theta^{res} & 0 \\ 0 & 0 & 1 \end{bmatrix} (^w\mathbf{P}^{traj} - ^w\mathbf{P}^{res}) \end{aligned} \quad (9)$$

The motion constraint as shown in (10) is imposed, because the mobile robot can not move sideways.

$$^w\dot{x}^{traj} \sin ^w\theta^{traj} - ^w\dot{y}^{traj} \cos ^w\theta^{traj} = 0 \quad (10)$$

By using (9) and (10), $\dot{\mathbf{P}}^{err}$ is derived as (11).

$$\begin{aligned} \dot{\mathbf{P}}^{err} &= [\dot{x}^{err} \quad \dot{y}^{err} \quad \dot{\theta}^{err}]^T \\ &= \begin{bmatrix} \dot{y}^{err} \omega^{res} - v^{res} + v^{traj} \cos \theta^{err} \\ -\dot{x}^{err} \omega^{res} + v^{traj} \sin \theta^{err} \\ \omega^{traj} - \omega^{res} \end{bmatrix} \end{aligned} \quad (11)$$

where the desired trajectory translational velocity v^{traj} and angular velocity ω^{traj} are derived from the differentiation of $^w\mathbf{P}^{traj} = [^wx^{traj}, ^wy^{traj}, ^w\theta^{traj}]^T$. Furthermore, the translational velocity command v_t^{cmd} and the angular velocity ω_t^{cmd} command for the desired trajectory of the mobile robot are defined as (12).

$$\begin{bmatrix} v_t^{cmd} \\ \omega_t^{cmd} \end{bmatrix} = \begin{bmatrix} v^{traj} \cos \theta^{err} + K_x x^{err} \\ \omega^{traj} + v^{traj} (K_y y^{err} + K_\theta \sin \theta^{err}) \end{bmatrix} \quad (12)$$

where K_x , K_y and K_θ represent for positive constants. When

the robot realized the (12), the velocity responses are $v^{res} = v_t^{cmd}$, $\omega^{res} = \omega_t^{cmd}$ in the (11). Therefore, $\dot{\mathbf{P}}_{err}$ is derived from (11) and (12).

$$\begin{aligned} \dot{\mathbf{P}}_{err} &= [\dot{x}^{err} \quad \dot{y}^{err} \quad \dot{\theta}^{err}]^T \\ &= \begin{bmatrix} \{\omega^{traj} + v^{traj}(K_y y^{err} + K_\theta \sin \theta^{err})\} y^{err} - K_x x^{err} \\ -\{\omega^{traj} + v^{traj}(K_y y^{err} + K_\theta \sin \theta^{err})\} x^{err} + v^{traj} \sin \theta^{err} \\ -v^{traj}(K_y y^{err} + K_\theta \sin \theta^{err}) \end{bmatrix} \end{aligned} \quad (13)$$

In order to prove the performance of trajectory planning for the desired trajectory, the Lyapunov function is calculated. In this proof, it is assumed that the velocity for the desired trajectory v^{traj} is more than 0.

$$v^{traj} > 0 \quad (14)$$

One of the candidates for Lyapunov function V_a is expressed as follows.

$$V_a = \frac{1}{2} \{ (x^{err})^2 + (y^{err})^2 \} + \frac{1}{K_y} (1 - \cos \theta^{err}) \quad (15)$$

By using (12)–(15), the time derivative of V_a is derived as follows.

$$\begin{aligned} \dot{V}_a &= \dot{x}^{err} x^{err} + \dot{y}^{err} y^{err} + \dot{\theta}^{err} \sin \theta^{err} / K_y \\ &= [\{\omega^{traj} + v^{traj}(K_y y^{err} + K_\theta \sin \theta^{err})\} y^{err} - K_x x^{err}] x^{err} \\ &\quad + [-\{\omega^{traj} + v^{traj}(K_y y^{err} + K_\theta \sin \theta^{err})\} x^{err} \\ &\quad + v^{traj} \sin \theta^{err}] y^{err} \\ &\quad + [-v^{traj}(K_y y^{err} + K_\theta \sin \theta^{err}) \sin \theta^{err} / K_y] \\ &= -K_x (x^{err})^2 - v^{traj} K_\theta \sin^2 \theta^{err} / K_y \leq 0 \quad (16) \end{aligned}$$

Since the time differentiation of Lyapunov function becomes semi-indeterminate, the difference between the desired trajectory and the mobile robot converges to 0 ($x^{err} \rightarrow 0$, $\theta^{err} \rightarrow 0$, and $y^{err} \rightarrow 0$).

4. Collision Avoidance Control

As one of collision avoidance control methods, this section shows the reflective motion planning using virtual manipulators proposed by Yamazaki and Inaba⁽¹¹⁾⁽¹²⁾. Figure 3 shows the image diagram of collision avoidance. When the number of multiple objects is n , $n + 1$ virtual manipulators are mounted on the mobile robot. One of these manipulators is “leader manipulator” which plays the role to follow the desired trajectory. Others are “assistant manipulators” which play the role for collision avoidance.

Figure 3(a) shows the mobile robot follows the desired trajectory by the leader manipulator. In Fig. 3(b), the mobile robot detects obstacles by using sensors. In Fig. 3(c), the assistant manipulators are generated, when there are obstacles that the distance between the obstacle and the mobile robot is smaller than D_{max} . D_{max} means the distance threshold between the obstacle and the mobile robot. The robot can avoid obstacles by using assistant manipulators. As shown in Fig. 3(d), the assistant manipulators are deleted, the distance between the obstacle and the mobile robot is larger than D_{max} . Therefore, the robot can follow the desired trajectory and avoid the obstacles.

The trajectory planning for the desired trajectory and collision avoidance are calculated as follows. The number of the

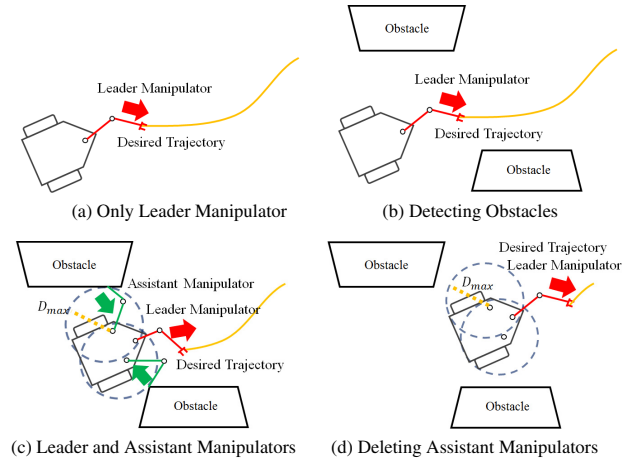


Fig. 3. Image Diagram of Reflective Motion

virtual manipulator is $n + 1$. Each virtual manipulator has 2 joints. Therefore, $[\theta_{01}, \theta_{02}, \theta_{11}, \theta_{12}, \dots, \theta_{n1}, \theta_{n2}]^T$ means the joint angles of $n + 1$ virtual manipulators. This state vector is written as $\dot{\mathbf{q}} = [v_{rf}^{cmd}, \omega_{rf}^{cmd}, \dot{\theta}_{01}, \dot{\theta}_{02}, \dot{\theta}_{11}, \dot{\theta}_{12}, \dots, \dot{\theta}_{n1}, \dot{\theta}_{n2}]^T$. v_{rf}^{cmd} and ω_{rf}^{cmd} mean the translational and the angular velocity of the mobile robot. $\mathbf{x} = [x_{h0}, y_{h0}, \dots, x_{hn}, y_{hn}]^T$ means the position of the end-effector. The relationship between the velocity of the end-effector $\dot{\mathbf{x}}$ and the state vector of the robot $\dot{\mathbf{q}}$ is expressed as (17).

$$\dot{\mathbf{x}} = \mathbf{J} \dot{\mathbf{q}} \quad (17)$$

\mathbf{J} is the Jacobian matrix. The Jacobian matrix from each joint to the end-effector is expressed as follows.

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{b0} & \mathbf{J}_{m0} & \mathbf{0} & \dots & \dots & \dots & \dots \\ \mathbf{J}_{b1} & \mathbf{0} & \mathbf{J}_{m1} & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{J}_{bi} & \mathbf{0} & \dots & \dots & \mathbf{J}_{mi} & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{J}_{bn} & \mathbf{0} & \dots & \dots & \dots & \dots & \mathbf{J}_{mn} \end{bmatrix} \quad (18)$$

The Jacobi matrix \mathbf{J} is classified into two matrices, that is, \mathbf{J}_{bi} and \mathbf{J}_{mi} . \mathbf{J}_{bi} is the Jacobian matrix of the mobile robot movement component for the i -th manipulator. \mathbf{J}_{mi} is the Jacobian matrix of the i -th manipulator. The translational velocity and the angular velocity of the mobile robot and each joint are calculated from the pseudo inverse matrix of (18).

$$\dot{\mathbf{q}} = \mathbf{J}_w^\# \dot{\mathbf{x}} + \lambda (\mathbf{I} - \mathbf{J}_w^\# \mathbf{J}) \mathbf{A} \quad (19)$$

where $\mathbf{J}^\#$ is the SR-inverse of the Jacobian matrix⁽¹³⁾. $\mathbf{J}^\# = \mathbf{J}(\mathbf{k}\mathbf{I} + \mathbf{J}\mathbf{J}^t)^{-1}$. $\mathbf{J}_w^\#$ means $\mathbf{J}^\#$ is multiplied by the weight coefficients matrix \mathbf{W} . w_b , w_l , and w_a are components of the weight coefficients matrix \mathbf{W} . w_b , w_l , and w_a are weight coefficients to determine the relation between the robot movement and the leader/assistant manipulators. λ is the gain of the null-space. \mathbf{A} is the evaluation function in the null space. The detail of this collision avoidance method is described in the references⁽¹¹⁾⁽¹²⁾.

5. Proposed Method

This section proposes the hybrid control of trajectory

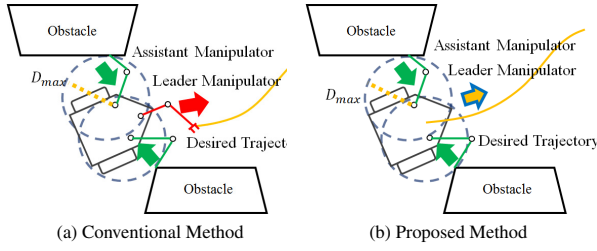


Fig. 4. Concept of Obstacle Avoidance

planning for the desired trajectory and collision avoidance based on the optimization problem. The previous trajectory planning methods for collision avoidance such as, potential method⁽⁹⁾ and DWA⁽¹⁰⁾, do not consider both the robot motion with nonholonomic constrain and guarantees collision avoidance of static obstacles. For this issue, this paper proposes the hybrid control of the tracking planning for the desired trajectory and collision avoidance based on the optimization problem.

5.1 Hybrid Control Figure 4 shows the concept of collision avoidance of the conventional method⁽¹¹⁾⁽¹²⁾ and the proposed method. In the conventional method using virtual manipulator as shown in Fig. 4(a), trajectory planning for the desired trajectory is performed by the velocity control of the leader manipulator. The collision avoidance is achieved by using the virtual manipulator like the reflective motion. The translational velocity and angular velocity of trajectory planning for the desired trajectory and collision avoidance are calculated from (19). However, this method does not consider the position error between the position command and response, but the only velocity command of the leader manipulator. Therefore, the position error occurs after collision avoidance.

Figure 4(b) shows the concept of the proposed method. The trajectory planning for the desired trajectory is performed instead of the leader manipulator by using (12). The collision avoidance is calculated from assistant manipulators by using (19). In the hybrid control, the translational velocity and the angular velocity are described as follows.

$$v_{hy}^{cmd} = \alpha v_t^{cmd} + (1 - \alpha) v_{rf}^{cmd} \quad (20)$$

$$\omega_{hy}^{cmd} = \alpha \omega_t^{cmd} + (1 - \alpha) \omega_{rf}^{cmd} \quad (21)$$

where v_{hy}^{cmd} and ω_{hy}^{cmd} the translational velocity and the angular velocity of the hybrid control. α means the priority variable. The range of α is $0 \leq \alpha \leq 1$. When the priority variable α approaches 1, the robot gives priority to trajectory planning for the desired trajectory. When the priority variable α approaches 0, the robot gives priority to collision avoidance. By modifying α , it is possible to modify the priority of trajectory planning for the desired trajectory and collision avoidance.

5.2 Outline of Proposed Local Trajectory Planning Method Figure 5 shows the flowchart of the proposed local trajectory planning method. As shown in Fig. 5, this flowchart consists of 6 steps.

Step1: The desired trajectory is calculated from the global trajectory planning (offline). The obstacle information is obtained by using URG.

Step2: Based on the desired trajectory, the velocity commands for the desired trajectory v_t^{cmd} , ω_t^{cmd} are calculated by

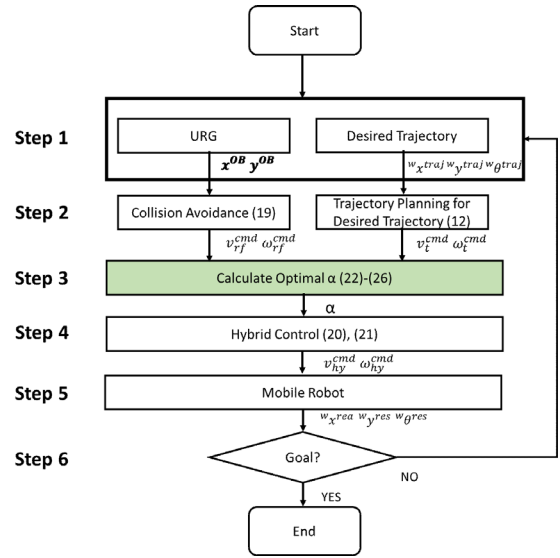


Fig. 5. Flowchart of Proposed Local Trajectory Planning Method

using (12). In addition, the velocity commands for collision avoidance v_{rf}^{cmd} , ω_{rf}^{cmd} are also generated by using (19).

Step3: The priority variable α is calculated by solving the optimization problem with constraints. Details is described in the part of the flowchart of Fig. 6.

Step4: The velocity commands v_{hy}^{cmd} and ω_{hy}^{cmd} are calculated from the optimal α and the hybrid control. The velocity commands v_{hy}^{cmd} and ω_{hy}^{cmd} are used for the robot motion. Details is described in 5.1.

Step5: The translational and angular velocity commands v_{hy}^{cmd} and ω_{hy}^{cmd} are realized by YP-spur Coordinator⁽¹⁴⁾. YP-spur Coordinator means the controller to achieve the velocity commands. Therefore, the mobile robot moves due to the velocity commands.

Step6: When the robot reaches the goal point, the flowchart is finished. Otherwise, go to **Step1**.

5.3 Design Method of Priority Variable α This section shows the design method of the priority variable α . In order to decide α , the optimization problem with the constraints is solved. This optimization is described as follows.

$$\text{maximize} \quad \alpha \quad (22)$$

$$\text{subject to} \quad \bigcap_{i=0}^n \Gamma_i \quad (23)$$

$$\Gamma_0 = \{\alpha \in \mathbb{R} | 0 \leq \alpha \leq 1\} \quad (24)$$

$$\Gamma_i = \{\alpha \in \mathbb{R} | \alpha < \alpha_i^{Small}, \alpha_i^{Large} < \alpha\}, \text{ if } \alpha^v < \alpha_i^{Small}, \alpha_i^{Large} < \alpha^v \quad (25)$$

$$\Gamma_i = \{\alpha \in \mathbb{R} | \alpha_i^{Small} \leq \alpha \leq \alpha_i^{Large}\}, \text{ if } \alpha_i^{Small} \leq \alpha^v \leq \alpha_i^{Large} \quad (26)$$

where n means the number of multiple obstacles. α_i^{Small} ($i = 1, 2, \dots, n$) and α_i^{Large} ($i = 1, 2, \dots, n$) stands for the priority values in the boundary conditions. α_i^{Large} is larger than α_i^{Small} . Γ_0 as shown in (24) means the constraint condition as the range of α . Eqs. (25), (26) mean the constraint condition for no collision between the robot and environment. α^v means the priority variable of the boundary value between

the collision and no collision. Under the constraint condition, the maximum value of α is used as the solution to the optimal problem. In other words, α with the highest priority of trajectory planning for the desired trajectory is derived within the range of collision avoidance.

5.4 Calculation Method of Priority Variable α The turning radius of the mobile robot R^{cmd} is derived from the translational velocity and the angular velocity command as (27).

$$R^{cmd} = \frac{v_{hy}^{cmd}}{\omega_{hy}^{cmd}} = \frac{\alpha v_t^{cmd} + (1 - \alpha)v_{rf}^{cmd}}{\alpha \omega_t^{cmd} + (1 - \alpha)\omega_{rf}^{cmd}} \dots (27)$$

α is derived from (27) as (28).

$$\alpha = \frac{v_{rf}^{cmd} - R^{cmd} \omega_{rf}^{cmd}}{R^{cmd}(\omega_t^{cmd} - \omega_{rf}^{cmd}) - (v_t^{cmd} - v_{rf}^{cmd})} \dots (28)$$

5.5 How to Decide Optimal Priority Variable α

Figure 6 shows the flowchart of the optimization method for priority variable α . The flowchart consists of 7 steps.

Step 3-1: The robot checks the number of obstacles n . If there are some obstacles, the variable i is initialized to 1. If there are no obstacles, go to **Step 3-7**.

Step 3-2: The velocity area that the robot avoids obstacles is calculated. As shown in Fig. 7, there are two trajectories where the left and right wheels of the robot are in contact with the obstacle i . R_i^{right}/R_i^{left} means the turning radius of the trajectory where the right/left wheel is contact with the obstacle (x_i^{OB}, y_i^{OB}) as shown in Fig. 7. Figure 7(a) shows the right wheel is contact with the obstacle. R_i^{RTR} is the distance between the right wheel $(0, B_R)$ and the i -th turning center $(0, R_i^{right})$. R_i^{RTO} is the distance between the i -th obstacle (x_i^{OB}, y_i^{OB}) and the i -th turning center $(0, R_i^{right})$. R_i^{RTR} and R_i^{RTO} are as (29) and (30).

$$R_i^{RTR} = |R_i^{right} - B_R| \dots (29)$$

$$R_i^{RTO} = \sqrt{(x_i^{OB})^2 + (y_i^{OB} - R_i^{right})^2} \dots (30)$$

If the robot moves as the circular orbit, when $R_i^{RTR} = R_i^{RTO}$, the right wheel is contact with the obstacle. At this time, this trajectory is treated as the limitation of collision avoidance. The turning radius R_i^{right} is derived as shown in (31).

$$R_i^{right} = \frac{B_R^2 - (x_i^{OB})^2 - (y_i^{OB})^2}{2(B_R - y_i^{OB})} \dots (31)$$

Figure 7(b) shows the left wheel is contact with the obstacle. R_i^{LTR} is the distance between the left wheel $(0, B_L)$ and the i -th turning center $(0, R_i^{left})$. R_i^{LTO} is the distance between the i -th obstacle (x_i^{OB}, y_i^{OB}) and the i -th turning center $(0, R_i^{left})$. R_i^{LTR} and R_i^{LTO} are as (32) and (33).

$$R_i^{LTR} = |R_i^{left} - B_L| \dots (32)$$

$$R_i^{LTO} = \sqrt{(x_i^{OB})^2 + (y_i^{OB} - R_i^{left})^2} \dots (33)$$

If the robot moves as the circular orbit, when $R_i^{LTR} = R_i^{LTO}$, the left wheel is contact with the obstacle. At this time, the trajectory is treated as the limitation of collision avoidance.

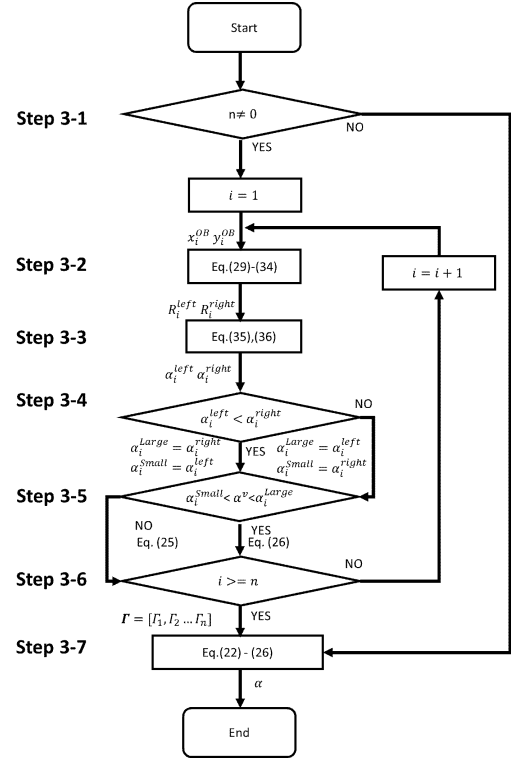


Fig. 6. Flowchart of Optimization Method for α

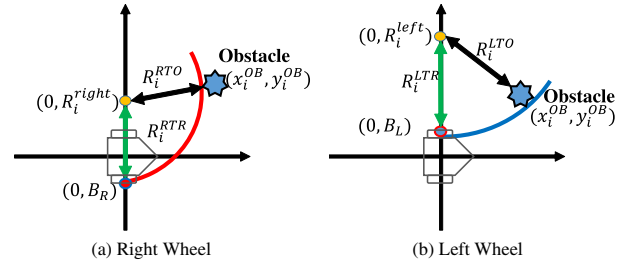


Fig. 7. Image Diagram of Collision Boundary Turning Radius

The turning radius R_i^{left} is derived as shown in (34).

$$R_i^{left} = \frac{B_L^2 - (x_i^{OB})^2 - (y_i^{OB})^2}{2(B_L - y_i^{OB})} \dots (34)$$

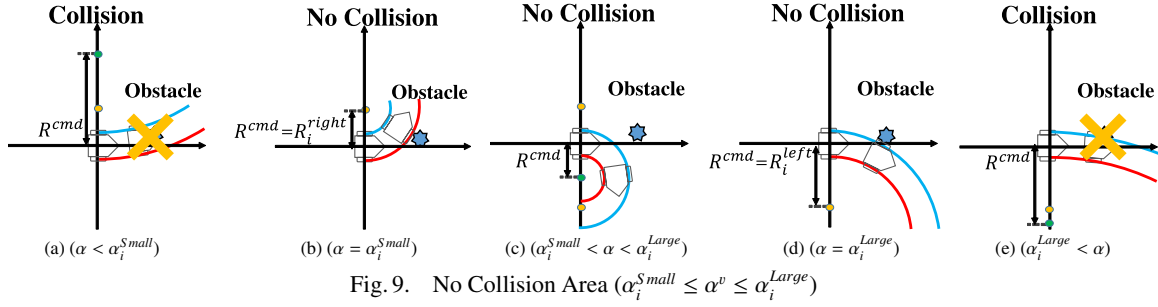
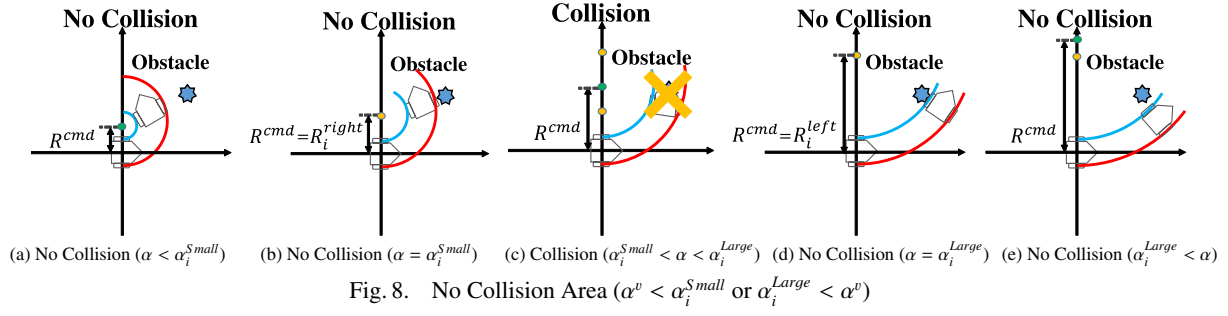
Step 3-3: Each priority variable can be obtained by substituting R_i^{right} and R_i^{left} into (28). The priority variables α_i^{right} and α_i^{left} can be derived as (35) and (36).

$$\alpha_i^{right} = \frac{v_{rf}^{cmd} - R_i^{right} \omega_{rf}^{cmd}}{R_i^{right}(\omega_t^{cmd} - \omega_{rf}^{cmd}) - (v_t^{cmd} - v_{rf}^{cmd})} \dots (35)$$

$$\alpha_i^{left} = \frac{v_{rf}^{cmd} - R_i^{left} \omega_{rf}^{cmd}}{R_i^{left}(\omega_t^{cmd} - \omega_{rf}^{cmd}) - (v_t^{cmd} - v_{rf}^{cmd})} \dots (36)$$

Step3-4: When the situation is $\alpha_i^{right} > \alpha_i^{left}$, $\alpha_i^{Large} = \alpha_i^{right}$ and $\alpha_i^{Small} = \alpha_i^{left}$. When the situation is $\alpha_i^{left} > \alpha_i^{right}$, $\alpha_i^{Large} = \alpha_i^{left}$ and $\alpha_i^{Small} = \alpha_i^{right}$.

Step3-5: The range of priority variables where robot does not collide with the i -th obstacle is derived. As shown in Figs. 8 and 9, there are two types of the relation between the turning



radius and priority variable α due to α^v . α^v is calculated as follows.

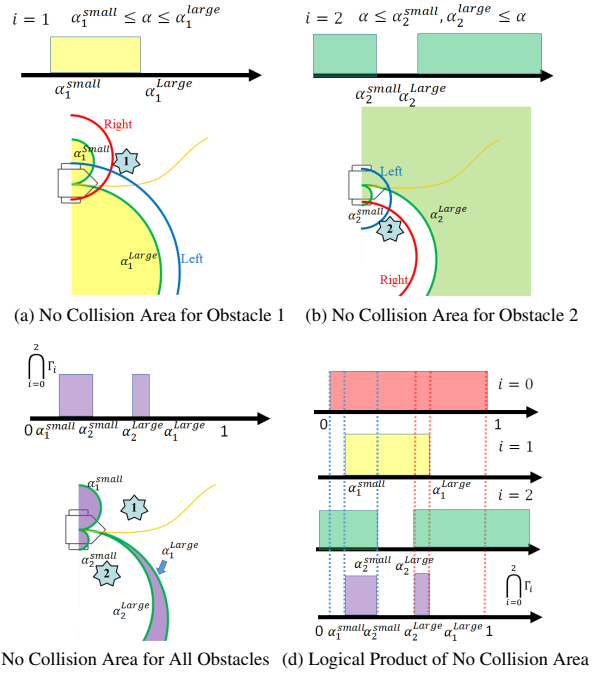
$$\alpha^v = -\frac{v_{rf}}{v_t - v_{rf}} \dots \dots \dots (37)$$

Figures 8–9 shows the image diagram of the no collision area. As shown in Fig. 8, this situation is $\alpha^v < \alpha_i^{small}$ or $\alpha_i^{large} < \alpha^v$. In this case, R^{cmd} moves from 0 to $+\infty$ when α is increased. Therefore, the range of the priority variable where the robot does not collide with the obstacle i is shown as (25). As shown in Fig. 9, this situation is $\alpha_i^{small} \leq \alpha^v \leq \alpha_i^{large}$. In this case, R^{cmd} moves from $+\infty$ to $-\infty$ when α is increased. Therefore, the range of the priority variable where the robot does not collide with the obstacle i is shown as (26).

Step3-6: If the variable i is larger than or equal to the number of obstacles n , go to **Step3-7**. Otherwise, go to **Step 3-2**.

Step3-7: The optimal value of the priority variable α is derived by the optimization problem with constraints. This optimization problem is expressed as (22)–(26). Eq. (23) is the logical product of (24)–(26). Finding the optimal value of α is equivalent to derive the optimal value of R^{cmd} in (27). By using the optimal value of the priority variable α , the trajectory is to be closest to the desired trajectory under the constraint of no collision.

5.6 Image of Proposed Method Figure 10 shows the image diagram of the optimization problem. This situation has two obstacles. Firstly, the robot extracts the range of the priority variable where the robot does not collide with obstacles as shown in Figs. 10(c) and (d). When the situation is $i = 1$ as shown in Fig. 10(a), the range where the robot does not collide with the obstacle is calculated like Fig. 9. When the situation is $i = 2$ as shown in Fig. 10(b), the range where the robot does not collide with the obstacle is calculated like Fig. 8. By using (23)–(26), the range where the robot does not collide with obstacles is extracted as shown in Figs. 10(c) and (d). Secondly, the priority value α which gives the highest priority to the trajectory following within that range is selected as shown in Figs. 10(c) and (d). By using (22), the optimal priority variable α is α_1^{large} . Finding the



optimal value of α is equivalent to derive the optimal value of R^{cmd} as shown in Figs. 10(c), (d) and Eq. (27). By using the optimal value of α , the trajectory is to be closest to the desired trajectory under the constraint of no collision.

6. Experiment

This section shows the experimental results to confirm the validity of the proposed method.

6.1 Experimental Setup Table 1 shows the control parameters of this experiments. These parameters were determined by trial and error in experiments. This research used the evaluation function in the null space $\mathbf{A}(=[0, 0, A_{01}, A_{02}, A_{11}, A_{12}, \dots, A_{ij}, \dots, A_{n1}, A_{n2}]^T)$. A_{ij} was described as follows.

Table 1. Control Parameters

K_x	Position X Gain	10
K_y	Position Y Gain	25
K_{θ}	Angle θ Gain	20
w_b	Weight Coefficient for Vehicle	1
w_l	Weight Coefficient for Leader Manipulator	1
w_a	Weight Coefficient for Assistant Manipulator	0.6
D_{max}	Distance of Assistant Manipulator Generation	0.4 [m]
l_1	Link Length of Assistant Manipulator	0.3 [m]
l_2	Link Length of Assistant Manipulator	0.3 [m]
θ_{01}^{ref}	Reference Value of Joint Angles of Leader Manipulator	$\pi/4$ [rad]
θ_{02}^{ref}	Reference Value of Joint Angles of Leader Manipulator	$-\pi/2$ [rad]



Fig. 11. Mobile Robot

$$A_{ij} = \theta_{ij}^{ref} - \theta_{ij}^{res} \dots \dots \dots (38)$$

where θ_{ij}^{ref} and θ_{ij}^{res} mean the reference and response values of the joint angles of the virtual manipulators. Each virtual manipulator has 2 joints ($j = 1, 2$). The coefficient i means the number of the obstacles. When $i = 0$, these joint angles mean the joint angles of the leader manipulator. When $i \geq 1$, these joint angles mean the joint angles of the assistant manipulator. In this paper, θ_{ij}^{ref} was designed as follows ($i \geq 1$).

$$\theta_{i1}^{ref} = \pi/2 * \text{sgn}(y_i^{OB}) \dots \dots \dots (39)$$

$$\theta_{i2}^{ref} = 0 \dots \dots \dots (40)$$

where $\text{sgn}()$ was the sign function, i.e.

$$\text{sgn}(y_i^{OB}) = \begin{cases} 1 & (y_i^{OB} \geq 0) \\ -1 & (y_i^{OB} < 0) \end{cases} \dots \dots \dots (41)$$

Figure 11 shows the mobile robot. The laser range finder was attached to the mobile robot. In order to measure environmental information, the laser range finder manufactured by HOKUYO AUTOMATIC CO., LTD. was used⁽¹⁵⁾. Figure 12 shows the model of the obstacle. The obstacle was treated as the point information. This point information was extracted from the closest point between the robot and the obstacle as shown in Fig. 12. Also, the sensor area was divided into two areas. One was the right area that was from -90 to 0 degrees in the robot coordinate system. The other was the left area that was from 0 to 90 degrees in the robot coordinate system. Therefore, these experiments do not consider the size of the obstacle, but the position of the closest obstacle. This point information was updated every measurement of URG. Figure 13 shows the experimental environment. This experiment was classified into two experiments. In Case 1, the robot moved along the straight line as shown in Fig. 13. Figure 13(b) shows the experimental environment in Case 2. In Case 1 and Case 2, there were obstacles along the desired

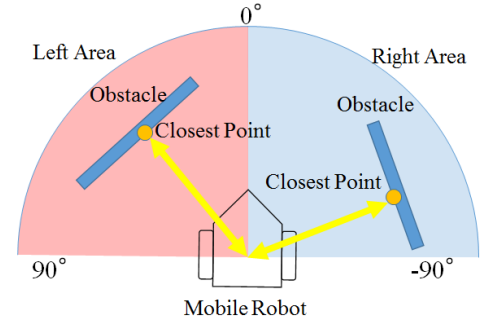


Fig. 12. Image Diagram of Obstacle Model

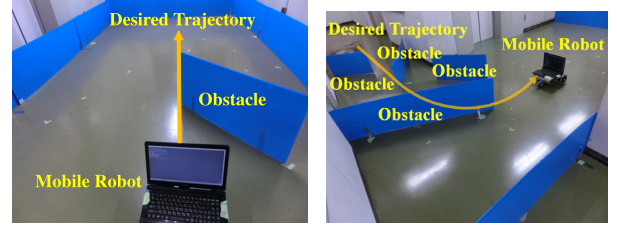


Fig. 13. Experimental Environment

trajectory.

In order to clarify the difference between the conventional methods and the proposed method, the desired trajectory in Case 1 was simple. In Case 1, the experiments were divided into four cases. Case 1A - Case 1C were treated as the conventional methods. In the Case 1D, the proposed method was used.

• Case 1A: Reflective Motion Planning

The reflecting motion planning consists of trajectory planning for the desired trajectory by the guidance of the leader manipulator and collision avoidance by assistant manipulators. The velocity command in (19) was used to the robot.

• Case 1B: Combination Control

The combination control consists of trajectory planning for the desired trajectory and collision avoidance by assistant manipulators. v_c^{cmd} and ω_c^{cmd} mean the translational velocity and the angular velocities considering the combination method. The translational velocity v_c^{cmd} and the angular velocity ω_c^{cmd} are expressed as (42)–(43). The combination velocities were a simple sum of trajectory planning for the desired trajectory and collision avoidance.

$$v_c^{cmd} = v_t^{cmd} + v_{rf}^{cmd} \dots \dots \dots (42)$$

$$\omega_c^{cmd} = \omega_t^{cmd} + \omega_{rf}^{cmd} \dots \dots \dots (43)$$

• Case 1C: Hybrid Control (α was the Constant Value)

The hybrid control consists of trajectory planning for the desired trajectory and collision avoidance. The velocity command (20)–(21) were used to the robot. The priority value α was the constant values ($\alpha=0, 0.25, 0.5, 0.75$).

• Case 1D: Hybrid Control (α was the Optimal Value)

The hybrid control consists of trajectory planning for the desired trajectory and collision avoidance. The velocity command (20)–(26) were used to the robot. The priority value α was the optimal value.

In Case 2, the straight line and the circular orbit was given to the mobile robot as the desired trajectory. This experiment as shown in Fig. 13(b) had several obstacles because of the

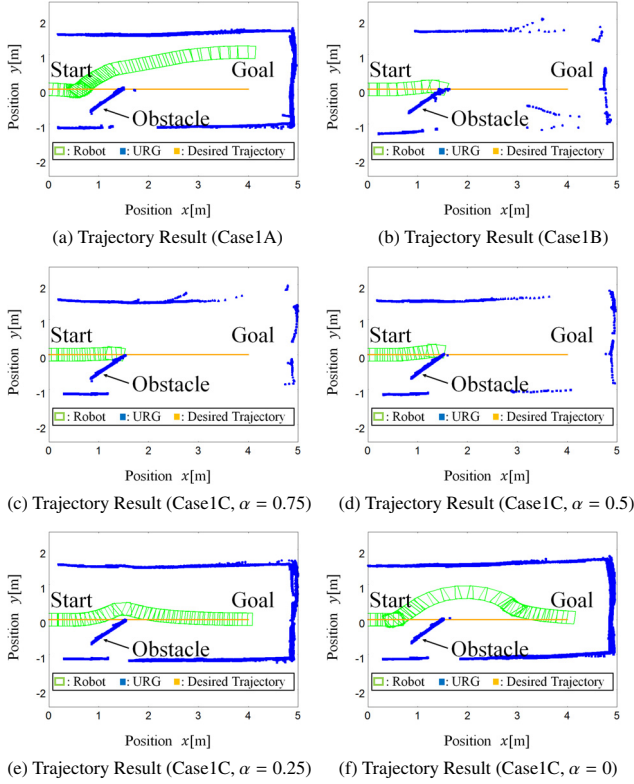


Fig. 14. Experimental Results of Conventional Method

confirmation of the validity of the proposed method in the complex situation.

6.2 Experimental Results In the experiments, environmental information was measured by the laser range finder. Figures 14(a)–(f) show the trajectory results of Case 1A–Case 1C as the conventional methods. Figures 15, 16(a)–(d) show the trajectory results of Case 1D and Case 2, the priority variable α results, the translational velocity command, and the angular velocity command, respectively as the proposed method. In Fig. 14, and Figs. 15, 16(a), the orange line shows the desired trajectory.

In Case 1A, the reflective motion planning was generated by virtual manipulators. In Fig. 14(a), there was the position error after the obstacle avoidance because of the velocity control.

In Case 1B, the combination control was used as the conventional method. Figure 14(b) shows the trajectory results in Case 1B. For collision avoidance, the position error occurred. Due to this position error, the velocity commands for trajectory planning for the desired trajectory was increased. As a result, the mobile robot collided with the obstacle. On the other words, the priority of trajectory planning for the desired trajectory was larger than collision avoidance because of the simple summation between trajectory planning for the desired trajectory and collision avoidance.

In Case 1C, the hybrid control with the constant value α was used as the conventional method. Figures 14(c)–(f) show the trajectory results in Case 1C. As shown in Figs. 14(c) and (d), the robot collided with the obstacle because α was the large value. The priority of the trajectory planning for the desired trajectory was higher than the priority of collision avoidance. From Figs. 14(e) and (f), the robot avoided

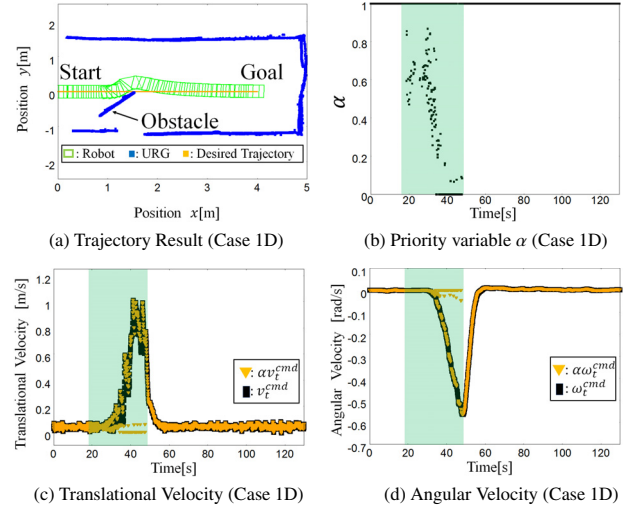


Fig. 15. Experimental Results of Proposed Method

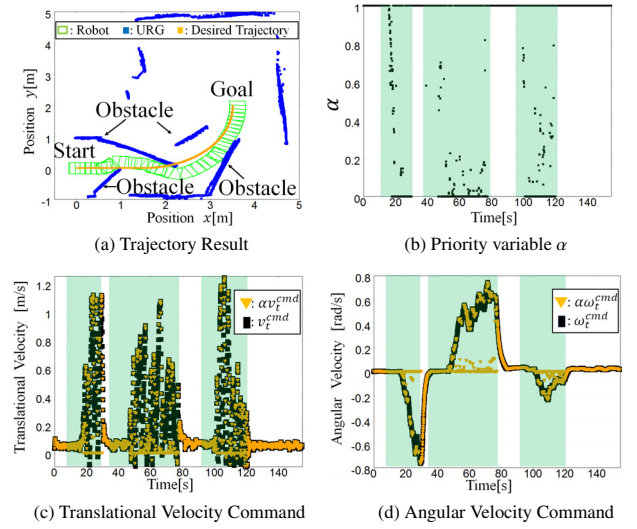


Fig. 16. Experimental Results (Case2)

the obstacle because α was the low value. The priority of collision avoidance was higher than the priority of trajectory planning for the desired trajectory. From Figs. 14(e) and (f), it was possible to achieve trajectory planning for the desired trajectory and collision avoidance. However, the shortest pass was not created, since the priority value α was not the optimal value.

In Case 1D, the hybrid control with the optimal value α was used as the proposed method. When the robot was avoiding obstacles, the mobile robot was deviated from the desired trajectory as shown in Fig. 15(a). The shaded parts in Figs. 15(b)–(d) were the range where collision avoidance was performed. Except for the shaded parts in Figs. 15(c), (d), the translational and angular velocities were about 0.1 [m/s] and 0.0 [rad/s]. On the other hand, in the shaded parts, these velocities were about 1.0 [m/s] and -0.5 [rad/s]. The absolute values of translational and angular velocities were large compared with the values before and after collision avoidance. This fact occurs because of trajectory planning for the desired trajectory. As shown in (12), the velocity generation consists of the position error x^{err} , y^{err} , and θ^{err} . When the robot avoids obstacles, the position error becomes large. Therefore, the

absolute value of velocities were larger than before and after collision avoidance. Also, as shown in the shaded parts in Figs. 15(c), (d), jump values of velocities were seen sometimes because the obstacle was treated as the point information. This point information was extracted from the closest point between the robot and the obstacle. This point information was updated every measurement of URG. Due to collision avoidance, the closest point of the obstacle was successively changed. Therefore, environment information was jumped as shown in Figs. 15(c), (d). Although the robot does not collide with the obstacle, it was the problem that the robot motion was not smooth. From shaded parts in Figs. 15(b)–(d) mean modification of the priority between trajectory planning for the desired trajectory and collision avoidance. From these results, the robot achieved trajectory planning for the desired trajectory and collision avoidance by using the hybrid control and the optimization problem of the priority variable α .

In Case2, there were several obstacles. The mobile robot achieved trajectory planning for the desired trajectory after collision avoidance. The shaded parts in Figs. 16(b)–(d) show the range where collision avoidance was performed. The desired trajectory in Case 2 was the straight line and the circular orbit. When the robot was avoiding obstacles, the mobile robot was deviated from the desired trajectory. As shown in the shaded parts in Figs. 16(c), (d), the absolute values of translational and angular velocities were large compared with the values before and after collision avoidance. From the shaded parts in Fig. 16(b), the priority variable α was modified in real time by solving the optimization problem with the constraints. This means the modifying the priority of trajectory planning for the desired trajectory and collision avoidance. The robot achieved trajectory planning for the desired trajectory and collision avoidance by using the hybrid control and the optimization problem of priority variable α . In these experiments, the environmental information was measured by URG. The cycle time of this measurement was about 100 [ms]. The control cycle of the proposed method was about 43 [ms] on average. In experiments, the mobile robot achieved the start/goal navigation in real time. From this fact, the control cycle was experimentally sufficient to solve in real time.

From these experimental results, the validity the effectiveness of the proposed method was confirmed via experimental results.

7. Conclusion

This paper proposed the hybrid control of trajectory planning for the desired trajectory and collision avoidance based on the optimization problem for the wheeled mobile robot. In this hybrid control, the priority of trajectory planning for the desired trajectory and collision avoidance was modified using the priority variable α . The priority variable α was designed by the optimization method with the constraints. As the constraint condition, the range of the priority variable α where the robot does not collide with the multiple obstacles was derived. Under this constraint condition, the priority variable α with the highest priority of trajectory planning for the desired trajectory was decided as the optimal value. As a result, the velocity commands considering trajectory planning for the desired trajectory and collision avoidance was calculated

in real time. The effectiveness of the proposed method was confirmed via experimental results.

As shown in Figs. 15 and 16(c), (d), jump values of velocities are seen sometimes because the obstacle is treated as the point information. Moreover, the robot motion was not smooth. In other words, this method may generate the large velocity value during collision avoidance. Therefore, these future works are listed.

- Changing of the obstacle model

The obstacle model considering the size should be utilized.

- The design method of parameters

The design method of parameters should be utilized.

- The velocity suppression filter

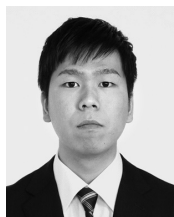
The velocity suppression filter considering the robot's specification should be developed.

Acknowledgment

This research was partially supported by JSPS KAKENHI Grant Number 19K04454, and JGC-S Scholarship Foundation.

References

- (1) B. Rouzier, M. Hazaz, T. Murakami, and W. Xu: "Application of Active Driving Assist to Remotely Controlled Car in Collision Avoidance", *IEEE J. Industry Applications*, Vol.7, No.4, pp.289–297 (2018)
- (2) T. Ohhira and A. Shimada: "Movement Control Based on Model Predictive Control with Disturbance Suppression using Kalman Filter including Disturbance Estimation", *IEEE J. Industry Applications*, Vol.7, No.5, pp.387–395 (2018)
- (3) G. Bresson, Z. Alsayed, L. Yu, and S. Glaser: "Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving", *IEEE Transactions on Intelligent Vehicles*, pp.194–220 (2017)
- (4) C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J.J. Leonard: "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age", *IEEE Transactions on Robotics*, Vol.32, No.6, pp.1309–1332 (2016)
- (5) A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios: "Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications", *IEEE Communications Surveys & Tutorials*, Vol.19, No.2, pp.1327–1346 (2017)
- (6) Y. Hasegawa and Y. Fujimoto: "Experimental Verification of Path Planning with SLAM", *IEEE J. Industry Applications*, Vol.5, No.3, pp.253–260 (2016)
- (7) Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi: "A Stable Tracking Control Method for an Autonomous Mobile Robot", *Proceedings of IEEE International Conference on Intelligent Robots and Automation*, pp.384–389 (1990)
- (8) Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi: "A stable tracking control method for a non-holonomic mobile robot", *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pp.1236–1241 (1991)
- (9) Y. Rasekhipour, A. Khajepour, S. Chen, and B. Litkouhi: "A Potential Field-Based Model Predictive Path-Planning Controller for Autonomous Road Vehicles", *IEEE Transactions on Intelligent Transportation Systems*, Vol.18, No.5, pp.1255–1267 (2017)
- (10) D. Fox, W. Burgard, and S. Thrun: "The Dynamic Window Approach to Collision Avoidance", *Proceedings of IEEE International Conference on Robotics & Automation Magazine*, Vol.4, pp.23–33 (2005)
- (11) K. Yamazaki and M. Inaba: "Trajectory Control of Wheeled Mobile Robots Based on Virtual Manipulators", *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pp.2974–2978 (2009)
- (12) K. Yamazaki and M. Inaba: "Reactive Motion Planning of a Wheeled Mobile Robot by Using Virtual Manipulators", *The Robotics Society of Japan*, Vol.29, No.2, pp.163–171 (2011) (in Japanese)
- (13) Z.Q. Hu, Z.G. Fu, and H.R. Fang: "Study of singularity robust inverse of Jacobian matrix for manipulator", *Proceedings. International Conference on Machine Learning and Cybernetics*, pp.406–410 (2002)
- (14) T-frog Project, <http://t-frog.com/>
- (15) HOKUYO AUTOMATICCO.LTD, <http://www.hokuyo-aut.co.jp/>



Masato Kobayashi (Student Member) received the B.E. degree in marine engineering from Kobe University, Japan, in 2017 and the M.E. degree in the graduate school of maritime sciences, Kobe University, Japan in 2019. He is currently working towards the Ph.D. degree in the graduate school of maritime sciences, Kobe University, Japan. His research interests include robotics, and motion control.



Naoki Motoi (Senior Member) received the B.E. degree in system design engineering and the M.E. and Ph.D. degrees in integrated design engineering from Keio University, Japan, in 2005, 2007 and 2010, respectively. In 2007, he joined the Partner Robot Div., Toyota Motor Corporation, Japan. From 2011 to 2013, he was a research associate at Yokohama National University. Since 2014, he has been with Kobe University, Japan, where he is a currently Associate Professor. His research interests include robotics, and motion control.