

Motion Planning Method with Pushing Posture Selection for Arranging Office Chairs Using Dual-arm Mobile Robot*

Sotaro Suwa, Takuya Iwasaki, Kimitoshi Yamazaki, *Member, IEEE*

Abstract— In this study, a motion generation method is proposed to enable a dual-arm mobile manipulator to move a chair to a designated position. This method enables the selection of a robot pose for grasping a chair from candidates. Furthermore, we propose a method for evaluating the appropriateness for two- and one-handed manipulation. After grasping a chair, the robot should transport it to the desired position. Consequently, we develop a motion generation method using quadratic programming optimization. This method calculates the displacement of the robot's joints for motion while satisfying constraint conditions such as joint limit and obstacle avoidance. Through simulation verification, we confirmed that an upper body humanoid robot can perform a chair arrangement task in three common layouts using our methods.

I. INTRODUCTION

The layouts of meeting rooms in companies and schools are frequently changed according to the number of participants and the meeting purpose. Therefore, most meeting room fixtures are on casters and are easy to move. Nevertheless, considerable manpower and time are required to rearrange tables and chairs in such cases. Using a robot to perform any part of this task would reduce such burden.

Studies based on this consideration have been conducted, where robots are used to automate the carrying of chairs and other objects [1, 2]. In terms of robotic planning, research on tidying tasks is divided into two approaches: task planning and motion planning. The former discusses the order of actions, such as which of several objects to push in what order and direction. The latter deals with issues such as which motion sequence should be used to push a specific object. In both cases, it is typically decided in advance where and how the robot will touch the specific object. Therefore, if there is insufficient open space behind an object for the robot's entire body to enter, manipulation of the object may be postponed or deemed impossible. However, a robot equipped with many manipulators can push the object using one of the manipulators; thus, the transportation task can be performed even in such a narrow space. In other words, a robot with a high degree of freedom, such as a mobile manipulator, is expected to perform chair arrangement tasks efficiently by appropriately utilizing feasible postures.

The objective of this study is to establish a motion generation method for arranging chairs in rooms, as described above. Assuming a dual-arm mobile robot, we present a method of pushing motion generation with posture selection that enables a chair-pushing task from the in situ placement of tables and chairs. Fig. 1 shows a schematic of different ways of performing the task by a person. If the space is wide open, the

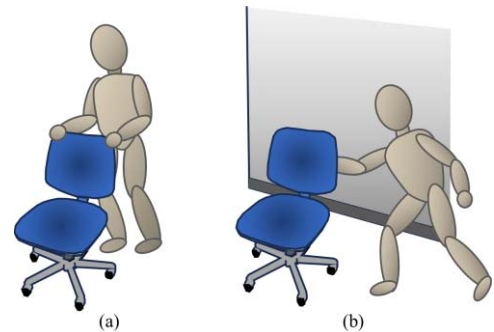


Figure 1. Different ways in which a human can manipulate a chair

person can move to the back of the chair and grasp it with both hands, as shown in (a). If not, the person moves the chair using one hand to push it from a position while avoiding collision with the obstacle, as shown in (b). This is because the person cannot get directly behind the chair. That is, how a human holds and pushes a chair depends on the configuration of the space. If a dual-arm mobile robot can execute such resourceful action choices, the arrangement of chairs will be automated for various situations.

The contributions of this study are as follows:

- A method for selecting grasping and pushing modes depending on the starting posture of the robot and the position of the target object is proposed.
- A method for motion generation with obstacle avoidance is proposed for a robot to deliver a chair.
- The effectiveness of the proposed methods is verified by simulating an orderly arrangement of several chairs based on a desk layout commonly used in conference rooms.

This paper is structured as follows. Section II presents related work. Section III presents the requirements and our approach. Section IV explains the proposed method for arranging office chairs in meeting rooms. Section V introduces the verification simulation and discussion. Finally, Section VI concludes this study.

II. RELATED WORK

Pushing and carrying objects has been a research topic in the field of intelligent robotics for many years. Hugli et al. [3] presented an architecture for integrating various vision devices in an autonomous agent and demonstrated its effectiveness by performing a tidying-up-chairs task as an example. Ben-Shahar et al. [3] proposed a planning method for repositioning multiple movable objects by pushing them

* This work was supported in part by NEDO(JPNP20004) and JST [Moonshot R&D][Grant NumberJPMJMS2034]. Sotaro Suwa and Takuya Iwasaki are with Graduate School of Engineering, Shinshu University, Wakasato 4-17-1, Nagano, Nagano, Japan

K. Yamazaki is with the Faculty of Engineering, Shinshu University, Wakasato 4-17-1, Nagano, Nagano, Japan (corresponding author to provide e-mail: kyamazaki@shinshu-u.ac.jp).

using a mobile robot. Zito et al. [4] extended the rapidly-exploring random tree (RRT) planner to search for a sequence of pushes that move the object. Tekin et al. [2] investigated tidying on the basis of a pushing motion by changing the direction of the caster wheels of the object. There are also studies on cooperative transport using multiple robots. Ross et al. [5] and Chen et al. [6, 7] used multiple robots to transport large objects such as furniture. Rus et al. [8] used two robots to rotate and pass a large sofa through a door. In these studies, the robot did not have multiple degrees of freedom (DoFs); thus, behaviors such as inserting the arm into a narrow space for manipulation were not considered.

It is easy to anticipate that obstacles may be encountered while the robot is carrying a chair. Various methods have been investigated for mobile robots to avoid obstacles in rooms with numerous chairs and desks. In Chen et al. [9], Wang et al. [10], and Stilman et al. [11, 12], moving paths were generated while considering movable chairs and desks. Yoshida et al. [13] developed a method for robots to transport objects without the robot or object colliding with obstacles. With reference to these existing results, we take an approach that separates the transported chair from the robot. We then formulate and solve the optimization problem.

Mathematical optimization is suitable for generating motions for multi-DoF robots. In this regard, many methods have been proposed. Lunenburg et al. [14] presented a method for computing the optimal base pose for manipulation using as few iterations or samples as possible. Harada et al. [15] proposed a method for planning a sequence of base positions for dual-arm mobile manipulators that perform multiple pick-and-place tasks. Wei et al. [16] proposed a method for optimizing the results of inverse kinematics using neural networks. Mittal et al. [17] introduced a method for opening various doors in a kitchen using model predictive control. Kuribayashi et al. [18] formulated the kinematics of a mobile manipulator using differential algebraic modelling and solved the generation of a door-opening motion as an optimization problem. Using these results as a reference, this study also uses mathematical optimization for motion generation. In particular, we refer to the formulation of Agravante et al. [19].

B. Maintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

III. PROBLEM SETTING AND APPROACH

A. Problem Setting

The following conditions are considered in this study.

1. The chair has casters and a swivel seat. The initial posture of the chair is known.

2. The table arrangement is known and does not change during the tidying process.
3. The chairs to be moved are designated in advance.

Under these conditions, the robot is required to move the chair to a predetermined table location. However, for some chair and table postures, the robot's entire body may not be able to pass directly behind the chair.

B. Our Approach

A dual-arm mobile manipulator is used. Depending on the conditions of the open space, the robot selects dual- or single-arm manipulation. The standing position of the mobile platform should also be selected. These can be defined relative to the chair to be pushed and the table where the chair is stored. Therefore, we adopt a policy of defining the relative candidate postures of the robot in advance and selecting an appropriate candidate depending on the specific situation. However, pushing a chair with one hand continuously is less stable than pushing with both hands and places a greater load on the manipulator. Therefore, while transporting the chair close to a target table, the robot chooses a wide space and pushes the chair from the back with both hands. Only when the chair is brought close to the table can the robot manipulate it with one hand.

Performing the arrangement task using the method described above, the robot may be required to move along a curved trajectory or avoid obstacles while pushing the chair. It should be ensured that the chair moves smoothly and efficiently and that there is no collision between the robot, the chair, and its surroundings. To achieve this, we propose a motion generation method based on the visual servo framework.

In the proposed method, the direction of the obstacle (as seen by the robot) is calculated by computing the vector similarity measure. The robot then calculates the direction in which the vector similarity with the obstacle direction is less than a threshold while approaching the target and moves in this direction to avoid the obstacle. However, it is also necessary to ensure that the chair being delivered by the robot does not collide with an obstacle. Therefore, in this study, the obstacle direction is calculated from the perspectives of the robot and the chair, and the robot is moved in the direction closest to the target that has few obstacles to avoid obstacles.

IV. PROPOSED METHOD

A. Overall Structure of the Proposed Methods

The proposed method is mainly divided into three parts. The first part uses an objective function to determine the robot's starting posture for chair grasping based on its positional relationship to an obstacle. The second part is used to avoid obstacles and consists of calculating the vector similarity to determine the direction in which the robot moves away from obstacles while approaching the target posture. The third part uses quadratic programming (QP) optimization to calculate the joint angles of the robot for grasping and pushing the chair.

Figure 2 shows the workflow for arranging one chair. The three aforementioned parts are executed in turn on both

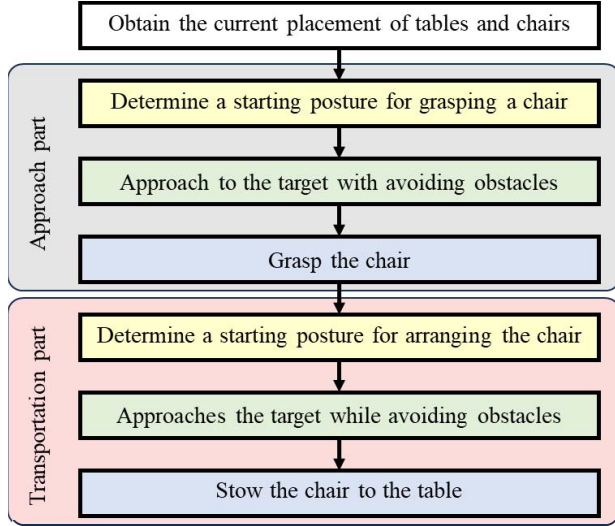


Figure 2. Workflow for arranging one chair.

approaching and manipulating actions. The yellow section represents the determination of the starting posture for chair manipulation, as described in Section IV-B. The green section represents moving the mobile base of the robot while ensuring that obstacles are avoided, as described in Section IV-C. The blue section represents the calculation of the joint angle by QP optimization, as described in Section IV-D.

B. Determination of Starting Posture for Chair Grasping

First, as shown in Fig. 3, we set several oriented points as candidate postures for the mobile platform to begin transporting the chair or stowing the chair to the table. These points are defined relative to the target chair or table. At each oriented point, the working posture of the robot is decided in advance. For instance, in the left panel of Fig. 3, p_1 , p_2 , and p_3 (i.e. $m \in \{1,2,3\}$) correspond to the robot grasping the chair with both arms, the right arm, and the left arm, respectively.

Note that the same method can be used to determine the starting postures for moving the chair. The right panel of Fig. 3 shows a case when stowing the chair to the table, where p_1 indicates a starting point where the chair is held from behind with both arms and moved forward to stow the chair to the table, whereas p_2 and p_3 require two steps of manipulation. After the chair is turned toward the table direction, the back of the chair is pushed toward the table with the right arm in case p_2 and with the left arm in case p_3 .

To determine an oriented point as the pose of a mobile platform, the evaluation value P_m is calculated for each chair or table. Specifically, the following equation is calculated for each of the m candidate points, and the m with the smallest value is selected as the pose for the mobile platform.

$$P_m = \sum_a \frac{1}{\sqrt{(x_m - x_d)^2 + (y_m - y_d)^2} - \frac{w_m}{\sqrt{(x_m - x)^2 + (y_m - y)^2}}}, \quad (1)$$

where (x,y) denotes the current position of the robot, (x_d, y_d) denotes the position of the obstacle, and (x_m, y_m) denotes the candidate position p_m of the m -th candidate point.

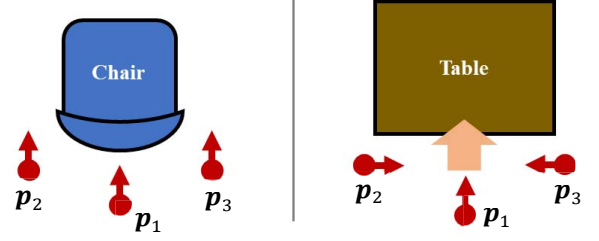


Figure 3. Candidate postures for the mobile platform. Orange arrow indicates the direction which the chair is pushed

The first term in Eq. (1) increases in magnitude with the proximity of the robot to the respective obstacle, and the number of these terms increases with the number of obstacles. The second term decreases in magnitude as the robot approaches the target and is intended to prevent the robot from going too far past the target. w_m denotes the weight coefficient for giving priority to the two-handed manipulation. This is because the chair can be manipulated more stably with two hands than with one hand.

As already mentioned, two-handed pushing is more stable than one-handed pushing. Therefore, if the robot is pushing a chair with one hand and the distance between the robot and surrounding obstacles exceeds a certain threshold, the robot changes its pose to grasp the chair with two hands. This policy necessarily increases the frequency of two-handed pushing and stabilizes the chair-arranging operation.

C. Obstacle Avoidance

1) *Basic algorithm for robot movement*: In this study, the vector similarity measure is computed to determine the direction of obstacles as seen by the robot. Consider that the only obstacles in the room are a table and a chair that is not grasped. These obstacles are approximated by rectangles and are projected onto a horizontal plane, and the robot model is projected onto the same plane. The robot then judges the risk of collision on the plane and generates avoidance motions.

To calculate the risk, equally spaced points are extracted from the contour of all obstacles. We call these points obstacle elements. If the robot approaches an obstacle element, a vector is defined from the robot center to the obstacle element and the moving direction in which the similarity with the vector is smaller than a threshold is calculated. The robot then moves a small distance in the direction. This process is repeated until the mobile platform is sufficiently close to the target pose.

2) *Obstacle avoidance for the robot*: Obstacle avoidance is performed using an extension of the method in [20]. Here, the movement of the robot alone is discussed. As shown in Fig. 4(a), a unit circle around the current robot position is defined, and dozens of observation points \mathbf{g} are placed at appropriate intervals on the circle. Obstacle elements are shown in red.

Let the vector from the center of the mobile platform to the i -th observation point be \mathbf{g}_i . For each obstacle element, \mathbf{v}_{ok} (k is the serial number of the obstacle element) denotes the vector from the center position of the mobile platform to the respective element. The evaluation value G_i at \mathbf{g}_i is calculated as follows:

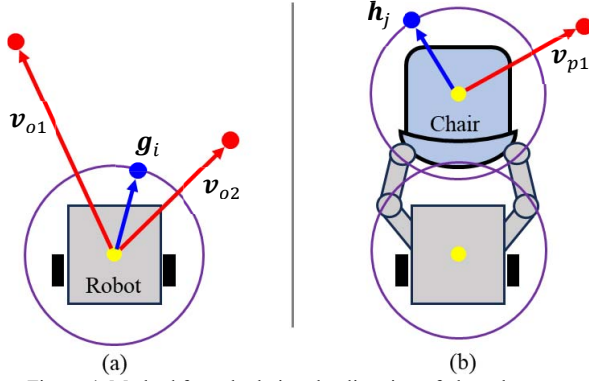


Figure 4. Method for calculating the direction of obstacle

$$G_i = \max \left\{ \mathbf{g}_i \cdot \frac{\mathbf{v}_{o1}}{\|\mathbf{v}_{o1}\|}, \dots, \mathbf{g}_i \cdot \frac{\mathbf{v}_{ok}}{\|\mathbf{v}_{ok}\|}, 0 \right\}. \quad (2)$$

Next, among the observation points for which the evaluated value G_i falls below a given threshold, the point with the largest calculated $\mathbf{g}_i \cdot \mathbf{v}_{g(xyz)}$ is selected. Here, $\mathbf{v}_{g(xyz)}$ denotes the vector from the current center position of the mobile platform to its next target position. The selected point $\mathbf{g}_{max} = (x_{gmax}, y_{gmax})$ is used to construct the pose displacement vector $\hat{\mathbf{v}}_g$ for the mobile platform as follows:

$$\hat{\mathbf{v}}_g = \begin{pmatrix} \alpha \mathbf{g}_{max} \\ \tan^{-1} \frac{y_{gmax}}{x_{gmax}} \end{pmatrix}. \quad (3)$$

where \mathbf{g}_{max} denotes the direction in which the robot approaches the target object most frequently among the directions with no obstacles. Therefore, the mobile platform should be oriented toward the target in the direction of \mathbf{g}_{max} to reduce the possibility of collision. This enables the robot to approach the target pose after prioritizing obstacle avoidance.

If the robot is close to an obstacle and the threshold is too small, the robot may not be able to determine the obstacle-free direction. Therefore, if no solution is obtained, the threshold value is gradually increased and the calculation is repeated.

3) *Obstacle avoidance while transporting a chair*: Next, we consider the problem of avoiding collisions between the robot and chair with obstacles while the robot is transporting the chair. As shown in Fig. 4(b), \mathbf{h}_j denotes the vector from the center position of the chair grasped by the robot to the j th observation point. For each obstacle element, \mathbf{v}_{pk} (k represents the serial number of the obstacle element) denotes the vector from the current center position of the mobile platform to the respective element. The method described in Section IV-B is used to calculate H_j as seen from the chair, as follows:

$$H_j = \max \left\{ \mathbf{h}_j \cdot \frac{\mathbf{v}_{p1}}{\|\mathbf{v}_{p1}\|}, \dots, \mathbf{h}_j \cdot \frac{\mathbf{v}_{pk}}{\|\mathbf{v}_{pk}\|}, 0 \right\}. \quad (4)$$

For an observation point, the cosine similarity between \mathbf{h}_j and the vector in the direction of the obstacle is calculated, and the

largest calculated value (i.e., 0) is chosen as the evaluation value of the point.

There are an equal number of observation points around the chair and the robot. Thus, among the observation points for which the evaluation values G_i and H_j are below the specified threshold, we select the observation point with the largest calculated $\mathbf{h}_j \cdot \mathbf{v}_{h(xyz)}$, where $\mathbf{v}_{h(xyz)}$ denotes the vector from the current center position of the chair to its next target position. Let $\mathbf{h}_{max} = (x_{hmax}, y_{hmax})$ be the selected point. Similar to the previous section (VI-C-2), we construct a pose displacement vector for the mobile platform $\hat{\mathbf{v}}_h$ using a constant value β to reflect the extent of parallel displacement of the mobile base:

$$\hat{\mathbf{v}}_h = \begin{pmatrix} \beta \mathbf{h}_{max} \\ \tan^{-1} \frac{y_{hmax}}{x_{hmax}} \end{pmatrix} \quad (5)$$

The robot moves a fixed distance in accordance with the pose displacement vector. By doing this step by step, the robot and the chair can both approach the target pose and move away from obstacles.

D. Determination of Joint Angles Using QP Optimization

The robot's whole-body movements while moving the chair are generated using QP optimization. This subsection describes the details. First, the basic form of QP is as follows:

$$\min. f_o = \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{c}^T \mathbf{x} \quad (6)$$

$$\text{s.t. } \mathbf{f}_c = \mathbf{A} \mathbf{x} - \mathbf{b} \leq \mathbf{0}, \quad (7)$$

where \mathbf{Q} represents the real object matrix, \mathbf{A} represents a matrix, and \mathbf{b} and \mathbf{c} represent vectors. The superscript T denotes the matrix transpose. We determine the \mathbf{x} that minimizes the objective function (Eq. 6) among the \mathbf{x} that satisfy the constraint function (Eq. 7).

In the proposed method, the objective function is set as follows:

$$\min. f_o = |\mathbf{v}'_{ag} - \mathbf{v}|^2, \quad (8)$$

where \mathbf{v}'_{ag} is the target displacement of the hand pose at each step, and \mathbf{v} is its pose displacement to be obtain. As $\mathbf{v} = \mathbf{J} \mathbf{q}$, where \mathbf{q} is the joint angle displacement and \mathbf{J} is the Jacobi matrix, it's substituted into the objective function to yield

$$\begin{aligned} \min. f_o &= |\mathbf{v}'_{ag} - \mathbf{J} \mathbf{q}|^2 \\ &= \mathbf{q}^T \mathbf{J}^T \mathbf{J} \mathbf{q} - 2 \mathbf{v}'_{ag}{}^T \mathbf{J} \mathbf{q} + \mathbf{v}'_{ag}{}^T \mathbf{v}'_{ag}. \end{aligned} \quad (9)$$

This finds the \mathbf{q} that minimizes $\mathbf{q}^T \mathbf{J}^T \mathbf{J} \mathbf{q} - 2 \mathbf{v}'_{ag}{}^T \mathbf{J} \mathbf{q}$. That is, $\mathbf{Q} = \mathbf{J}^T \mathbf{J}$ and $\mathbf{c}^T = \mathbf{v}'_{ag}{}^T \mathbf{J}$ are used as the objective functions in Eq. (6), and the optimization problem is solved.

As described in the next section, we verified the proposed method using an upper body humanoid robot with DoFs on its waist. In this case, the kinematics of the left and right arms share the waist axis. Therefore, consideration should be given when grasping the back of a chair with both hands or when the robot moves one arm and the waist simultaneously. We can define the Jacobi matrices for the common part and the right



Figure 5. Mobile manipulator

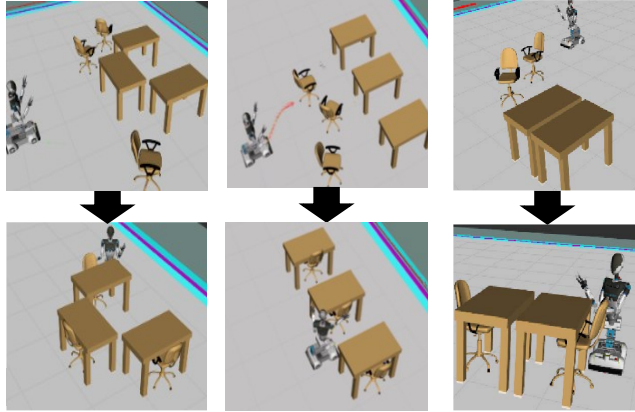


Figure 6. Arrangement of chairs before and after conducting the experiment

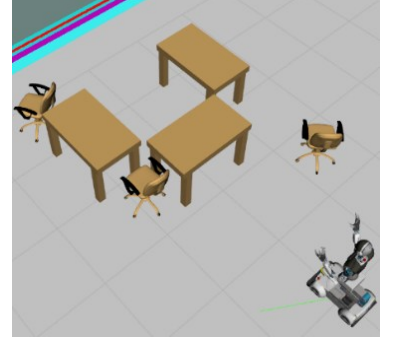


Figure 7. Example of robot failing to arrange the chairs correctly

and left arms as \mathbf{J}_{common} , \mathbf{J}_{left} , and \mathbf{J}_{right} , respectively, and create a single Jacobi matrix from them:

$$\mathbf{J} = \begin{pmatrix} \mathbf{J}_{common} & \mathbf{J}_{left} & \mathbf{0} \\ \mathbf{J}_{common} & \mathbf{0} & \mathbf{J}_{right} \end{pmatrix}. \quad (10)$$

As described in Section III-A, it is necessary to generate a manipulating posture after setting constraints to control the robot's motion. In this study, we set two constraints that are the same as those in [14]. The first constraint is the range of joint angles of the robot. Let $\bar{\mathbf{q}}$ and $\underline{\mathbf{q}}$ be the vectors lining up the upper and lower joint angle limits, respectively. \mathbf{A} and \mathbf{b} in Eq. (7) are calculated as follows.

$$\mathbf{A} = \begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \end{pmatrix}, \mathbf{b} = \begin{pmatrix} \bar{\mathbf{q}} \\ -\underline{\mathbf{q}} \end{pmatrix}, \quad (11)$$

where \mathbf{I} represents the identity matrix.

V. VERIFICATION BY SIMULATION

A. Preferences

A simulation was performed using the Aero [11] robot developed by THK Corporation (Fig. 5) and the Rviz, a visualization tool for the robot operating system (ROS). Aero is a dual-arm mobile manipulator with an omnidirectional mobile platform that measures 700 mm \times 450 mm (length \times width). Aero's arms have three DoFs at the shoulders and wrists and one DoF at the elbows. There are three DoFs for moving the neck and waist, and the height can be increased or decreased over a range of 1200–1700 mm by moving the lift portion between the waist and the mobile platform. In this study, only the posture of the robot arms was calculated for chair grasping, whereas the posture, including the arms and waist, was calculated for stowing a chair to a table. In the simulation, the chair dimensions were 500 \times 500 \times 900 [mm] and the table dimensions were 700 \times 1200 \times 700 [mm].

B. Experimental Conditions

The experimental conditions were configured according to the problem setup and are given below.

1. Chairs were randomly placed in a known posture visible to the robot.
2. All chairs were of the same type, and there were three grasping postures for each chair that the robot could perform (using both arms, left hand only, and right hand only).

3. Tables were arranged in one of three layouts commonly used in meeting rooms: face-to-face, school, or U-shaped. The postures of all tables were known, and the chair to be pushed into each desk was identified manually.
4. The obstacles were tables and chairs except for the chair grasped by the robot.

Two-stage verifications were performed. In the first stage, a simple chair arrangement task was performed without considering obstacle avoidance, which confirmed that the methods described in Sections VI-A and VI-C are effective. In the second stage, the elements described in Section VI-B were introduced to confirm the effectiveness of obstacle avoidance using the proposed method. The success/failure criteria are given below, where Items 1 and 2 were used for the first stage and all items were used for the second stage.

1. The starting pose of the chairs that were manipulated by the robot did not result in collision with an obstacle.
2. The chair was pushed into the specified table.
3. The chair did not collide with an obstacle while moving to the target pose.

Additional confirmation procedures are as follows.

1. Joint angles were determined using QP optimization, and the degree of error incurred in moving the specified joints to the target posture was determined.
2. Changes in the trajectory caused by changing the threshold for obstacle avoidance were determined.

C. Results of the First Stage

Twenty trials were performed in each layout. Figure 6 shows the three layouts before and after successful chair stowing. Figure 7 shows an example in which the arrangement task was deemed impossible. In this case, none of the candidate postures of the mobile platform relative to the chair to be moved could be used because of interference. If such a situation had not occurred, the chair arrangement operations would have been accomplished. However, even when there was sufficient open space enabling the robot to hold the chair with both hands, one-handed pushing was occasionally selected. This problem was caused by the fact that the robot's current position was close to the candidate position for one-handed pushing. This made the second term in Eq. (1) smaller.

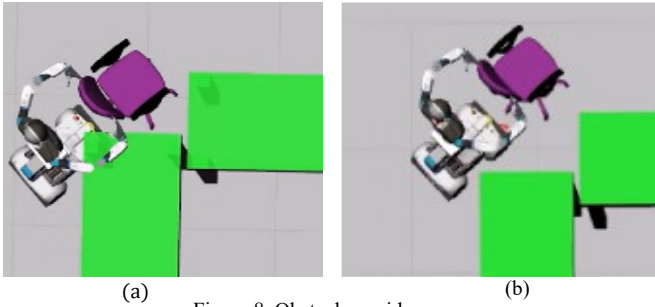


Figure 8. Obstacle avoidance

The manipulator can perform the task more stably using two hands than using one hand. Therefore, except when a candidate pose for pushing with both arms is inside or near an obstacle, the value of w_m should be adjusted to move the robot to that pose as much as possible.

Next, we describe the joint movement obtained using QP optimization. QP optimization was looped 500 times to move the arm of the manipulator to the target joint posture. Approximately 40 s was required to perform the 500 loops. Errors in the robot hand pose were approximately 1–3 mm for (x, y, z) and 2–3° for the three rotational parameters.

D. Results of the Second Stage

The proposed method determines the direction of movement by considering the position of obstacles relative to the center of the robot and the chair. Therefore, there are two ways to adjust the obstacle avoidance performance. One is to adjust the length of movement calculated using Eqs. (3) and (5). The shorter this distance, the easier it is to avoid a collision. The other is to adjust the threshold value for deriving \hat{v}_g and \hat{v}_h from the evaluated values G_n and H_n . The smaller this threshold, the easier it is to avoid obstacles. However, there is a disadvantage that the avoidance path becomes more circuitous.

Fig. 8(a) shows an example when the threshold values of G_n and H_n are set to 0.95. Conversely, (b) shows an example when the threshold is set to 0.85. Thus, we confirmed that the behavior of the robot changes depending on the threshold value. In other cases, a deadlock occasionally occurred when the threshold value was small in areas with dense obstacles. Therefore, we adopted a policy of increasing the threshold value and recalculating it when a deadlock is detected at the set threshold value. This policy increased the achievement rate of the proposed method. However, the proposed method cannot essentially avoid collisions; thus, a combination with global motion planning may be necessary in complicated environments.

VI. CONCLUSION

In this study, we proposed a method for a dual-arm mobile manipulator to move chairs using different pushing methods. We evaluated the method by determining the robot's starting posture for chair manipulation. For transporting a chair, QP optimization was used to calculate the displacement of the robot's joints for appropriate motion while satisfying the constraint conditions such as the joint limit and avoiding obstacles. Through verification by simulation, we confirmed

that an upper body humanoid robot could perform chair arrangement tasks in three common layouts.

Future work includes the combination with global motion planning. We are also considering extending the proposed method to enable actual robots to perform a chair arrangement task while considering dynamics and frictional forces.

REFERENCES

- [1] K. Yamazaki et al., "Home Assistant Robot for an Aging Society," *Proceeding of the IEEE*, Vol. 100, No. 8, pp. 2429 – 2441, 2012.
- [2] Tekin, M. et al., "Push-manipulation of complex passive mobile objects using experimentally acquired motion models," *Autonomous Robots* 38 (2015): 317-329.
- [3] O. Ben-Shahar, et al., "To push or not to push: on the rearrangement of movable objects by a mobile robot," in *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 28, no. 5, pp. 667-679, 1998.
- [4] C. Zito, et al., "Two-level RRT planning for robotic push manipulation," In *IEEE/RSJ IROS*, 2012.
- [5] A. Ross, et al., "Ikeabot: An autonomous multi-robot coordinated furniture assembly system," In: 2013 IEEE International conference on robotics and automation. IEEE, p. 855-862, 2013.
- [6] J. Chen, et al., "Occlusion based cooperative transport with a swarm of miniature mobile robots," *Robotics*, IEEE Transactions on, vol. 31, no. 2, pp. 307–321, 2015.
- [7] J. Chen, et al., "A strategy for transporting tall objects with a swarm of miniature mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 863–869, 2013.
- [8] D. Rus. et al., "Moving furniture with teams of autonomous robots." In: *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems.. IEEE*, p. 235-242, 1995.
- [9] P. C. Chen et al, "Practical path planning among movable obstacles," *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA, USA, 1991, pp. 444-449 vol.1, doi: 10.1109/ROBOT.1991.131618.
- [10] M. Wang, et al., "Affordance-based mobile robot navigation among movable obstacles," In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE (2020) p. 2734-2740.
- [11] M. Stilman, et al., "Navigation among movable obstacles: Real-time reasoning in complex environments," *International Journal of Humanoid Robotics*, pp.479-503, 2004.
- [12] M. Stilman, et al., "Planning and Executing Navigation Among Movable Obstacles," 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 2006, pp. 820-826, doi: 10.1109/IROS.2006.281731.
- [13] E. Yoshida, et al., "Humanoid motion planning for dynamic tasks," in 5th IEEE-RAS Int. Conf. on Humanoid Robots, pp. 1–6, 2005.
- [14] J. Lunenburg, R. van de Molengraft, M. Steinbuch, Minimizing the number of iterations when computing a base pose for manipulation by mobile base inclusion in the inverse kinematics, in: *Proc. of 16th Int'l Conf. on Advanced Robotics*, 2013, pp. 1–7.
- [15] K. Harada et al., Base position planning for dual-arm mobile manipulators performing a sequence of pick-and-place tasks, *Proc. of IEEE-RAS 15th Int'l Conf. on Humanoid Robots*, pp. 194–201, 2015.
- [16] C. Wei, Y. Kong, R. Zhang, Y. Jiang, X. Xia, A repeatable optimization for kinematic energy system with its mobile manipulator application, *Complexity* (2019) <http://dx.doi.org/10.1155/2019/8642027>.
- [17] M. Mittal et al., "Articulated object interaction in unknown scenes with whole-body mobile manipulation," *arXiv:2103.10534*, 2021.
- [18] Y. Kuribayashi, K. Yamazaki, "Door-Opening Motion Generation Using DAE Modeling for Mobile Manipulators," in *Proc. of the 2023 IEEE Int'l Conf. on Robotics and Biomimetics*, pp. 1445 – 1451, 2023.
- [19] D. J. Agravante et al., "Visual Servoing in an Optimization Framework for the Whole-Body Control of Humanoid Robots," in *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 608-615, 2017.
- [20] T. Iwasaki et al., "Visual Servoing in Virtualized Environments Based on Optical Flow Learning and Constrained Optimization," *International Journal of Robotics and Automation*, 2023.
- [21] K. Sasabuchi, et al.: "The Seedroid Robot Platform: Designing a Multipurpose Compact Robot from Continuous Evaluation and Lessons From Competitions," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3983–3990, 2018.