Door-Opening Motion Generation Using DAE Modeling for Mobile Manipulators

Yusuke Kuribayashi*, Kimitoshi Yamazaki*, *Member, IEEE*

Abstract—**This paper describes a motion generation method for a robot performing door-opening tasks. Unlike prior studies that divided the door-opening task into subproblems, we propose a single computation approach to obtain robot motions. To achieve this, we construct a robot model utilizing differential algebraic equations (DAE) derived from the differential kinematics of mobile manipulators. Additionally, we present the formulation of a nonlinear optimization problem, including the definition of constraints for robot's via points, collision avoidance, and others. We then present our experimental findings conducted in both simulated and real environments, involving various types of hinged doors. We confirmed that the proposed method can generate suitable motions for door opening, both for the robot itself and for others.**

I. INTRODUCTION

Recently, robots have been widely used throughout society, spanning from factory to daily life. A key societal expectation for robots is to replace the object manipulation tasks that are manually performed. However, these tasks exhibit varying levels of complexity. Some necessitate a combination of multiple actions rather than a singular, straightforward task. Moreover, there are tasks that meet specific requirements, such as execution time and via postures.

We focus on door-opening tasks as a representative object manipulation task that can be substituted by a robot. Door opening is an intriguing research problem at the control level due to the closed-loop structure formed between the robot and the manipulated object, even when considering solely moving the door. Conversely, addressing what actions the robot should undertake post-door opening (e.g., move through the door and go out into the hallway) requires simultaneous consideration of the robot's hand trajectory, collision avoidance with the door, and the robot's standing position while manipulating the door, thereby presenting a formidable motion planning challenge. Therefore, door opening has been studied from the past to the present [1-7].

The purpose of this study is to establish a method for the latter of the above. Specifically, we consider the door-opening task as a problem of generating a sequence of movements involving the robot's entire body while satisfying some required constraints. We propose a method to achieve this problem through mathematical optimization. In this study, we assume the robot performed tasks is a mobile manipulator, hereafter simply termed as "robots".

Various methods have been proposed for acquiring motions to execute object manipulation tasks. Reinforcement learning methods [8][9] enable the acquisition of suitable motions, even for complex tasks involving multiple actions, through learning based on action data from task execution. Additionally, mathematical optimization methods [10][11] can compute a motion that aims to minimize the amount of motion and make it smooth within predefined constraints. However, when applying these methods to door-opening tasks such as the one in this study, segmentation of the robot's motion into distinct simple actions is a prerequisite, requiring separate planning for each. For instance, if a robot's standing position at an intermediate point needs to be constrained, the motion must be separated there, and then the actions before and after that point must be generated separately.

However, mathematical optimization methods have continued to develop, and recently methods have been proposed that are stable and computable under complex and specialized conditions [12][13]. In this study, we formulate robot models utilizing differential algebraic equations (DAE) and propose a method to achieve a unified motion for door opening. This approach considers both the geometric constraints of the robot and the constraints imposed at a certain timing during the motion.

The contributions of this study are as follows:

- We formulate motion generation for a door-opening task as a mathematical optimization problem and propose a method that computes the required robot motion in a single optimization calculation.
- We define novel constraints to prevent collisions with the door and to regulate the robot's motion parameters. Additionally, we demonstrate how our method can generate suitable motions, even if the door size or handle position changes.
- To validate the efficacy of our proposed method, we conducted door opening experiments. In the experiments using an actual robot, we also discuss how to address robot positioning errors.

The structure of this paper is as follows. Section II discusses related work. In Section III, we organize the problem framework and our approach. Section IV describes the formulation of the optimization problem. In Section V, we apply the problem formulated in the previous section to the door-opening task. Section VI describes the experimental method, Section VII presents the results, and Section VIII concludes this study.

II. RELATED WORK

A. *Door Operation by Autonomous Robots*

As noted in the preceding section, door opening has been the subject of many studies. Nagatani et al. [1] achieved

---

*Faculty of Engineering, Shinshu University, 4-17-1 Wakasato, Nagano, 380-8553, Japan. kyamazaki@shinshu-u.ac.jp

successful door opening and passage through a door with an autonomous robot based on a task-oriented approach. Sasaki et al. [2] achieved door opening through a simple control law using passive joints with minimal viscous resistance. Nagahama et al. [3] proposed a real-time method to determine door type and manipulation trajectory by measuring the force exerted on the robot's wrist. Mittal et al. [4] introduced a method for opening various doors in a kitchen using model predictive control in a linear optimization problem, which enables door opening to be accomplished in a dynamic environment with potential collisions. In these studies, door opening is achieved through flexible online motion modification. On the other hand, it is assumed that the robot's motion rules are either given manually or learned in advance.

Some studies have achieved a higher degree of autonomy in motion generation. Chitta et al. [5] achieved a door-opening task using graph search-based motion planning. There are also some studies that aim to acquire door-opening behavior through machine learning. Gu et al. [6] used asynchronous deep reinforcement learning to obtain door-opening motions with multiple robots of the same type and proposed a method to acquire the motions by integrating the training results into a single robot. Ito et al. [7] presented a method to learn a series of door-opening motions by dividing it into multiple actions (approaching the door, opening the door, and passing through the door) and integrating the generated models. In these studies, complex motions were automatically generated. However, the division into simple actions was determined manually, and the suitable motion when the door opening is considered as one complex action was not discussed.

B. DAE-based Modeling on Robotics Motion Generation

In our study, DAE is used for modeling robots. Similarly, there are studies that have applied DAE to the field of robotics. Costa-Castello et al. [14] employed DAE to model the motion of a constrained planar 2 DoFs robot system. Yang et al. [15] used DAE for the dynamic modeling of a multisegmented continuum robot. Wang et al. [16] adopted DAE to model the kinematics of a biped robot and tackled the dynamic optimization of the control system.

As described above, modeling using DAE has the potential to be used in robotics. We posit that it is better to approach the door-opening task from a higher-order perspective than before and to be able to solve it without dividing it into subproblems as in the literature [10][11]. We consider DAE as a potent tool to realize this approach and elaborate on its appropriate utilization in this paper.

III. PROBLEM SETTING AND APPROACH

A. Problem Setting and Issues

In human living spaces, hinged doors of diverse dimensions are installed in furniture and room entryways. We aim to have robots proficiently open these doors. The requisite robot motion varies depending on its objective; therefore, we categorize this task into two types: (1) door opening for others and (2) door opening for itself. Based on the discussion so far, the proposed method is required to address the following two key challenges.

1. It can be used for hinged doors of various dimensions.
2. The same method can be used to generate motion for both above door-opening types.

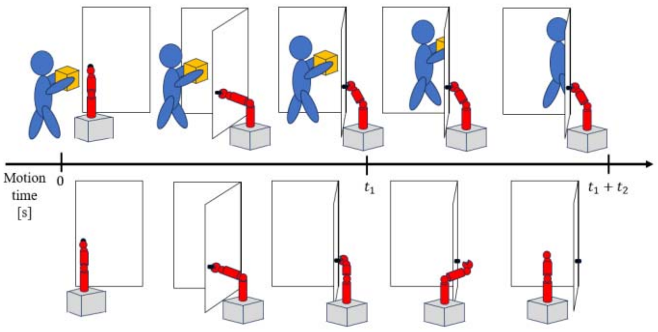


Figure 1. Variation of door opening according to work purpose. Upper row: open the door for others. Lower row: open the door for itself.

We explain item 2 in detail. As shown in the upper row of Fig. 1, type (1) is the case where the door is opened for human or other robots. In this case, the robot tasked with door opening is required to position so as not to impede the movement of others. This task can be realized through relatively simple motion because the robot can execute it if moving with the door's movement. However, as shown in the lower row of Fig. 1, type (2) is necessary when the robot is required to move between rooms or retrieve items from shelves. In this case, it is desirable to come in the area where the door has opened so that the robot can easily perform the next action. If robots have sufficient range of motion same as humans, they can move into the opened area simultaneously with opening the door. In such cases, it is regarded as a single-action task. However, due to limitations in the robot's degree of freedom and range of motion, there may be instances where the above behavior cannot be executed as a single action. In such cases, we should consider this task as composed of multiple actions, such as moving into the opened area after opening the door. Here, the authors argue that it is desirable to generate the motion in a single calculation rather than solving this motion generation problem individually for each action. This is because an action can end in a posture that is conducive to the next action, thereby enabling the generation of more efficient motions.

B. Approach

To address the two issues outlined in the preceding section, we explain our approach. First, for issue 1, we treat the door-opening task as a mathematical optimization problem. By incorporating door dimensions or relevant information as constraints in the optimization problem, motion generation can be executed for various doors.

Regarding issue 2, we construct a robot model using DAE. In the case of type (2) mentioned in Section III-A, multiple actions such as "opening the door" and "moving to the opened area" are required. Generally, it is difficult to generate these multiple actions by conventional optimization-based methods [10-12]. This is because the robot model, represented by algebraic equations, solely considers discrete time at specific points. By employing DAE instead of algebraic equations, we can simultaneously account for previous robot actions and geometric constraints of the robot at discrete times. This approach allows for the generation of robot motion in a single optimization calculation, even for problems requiring complex motions.

IV. FORMULATION TO OPTIMIZATION PROBLEM

A. Objective Function

We define the objective function as Eq. (1):

$$\min \int_0^t \boldsymbol{x}^\top \boldsymbol{Q} \boldsymbol{x} \, \mathrm{d}t. \tag{1}$$

$\boldsymbol{x}$ is a variable related to time. When the number of variables to be optimized is n, $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^\top$. $\boldsymbol{Q}$ is a diagonal matrix of size $n \times n$ with the weights of each variable as the diagonal component. The superscript denotes the matrix transpose. Since the elements of $\boldsymbol{x}$ may use variables with different systems of unit, we divide them by the maximum value of each variable for normalization, such that they can be evaluated equally. When performing optimization calculations, the objective function is discretized based on the discrete points described in the next section using the trapezoid rule.

In this study, to obtain smooth motion, expressed using variables $(x_\mathrm{r}, y_\mathrm{r}, \phi_\mathrm{r})$ denoting the position and angle of a mobile platform and variables $(\theta_1, \theta_2, \ldots, \theta_i)$ denoting joint angles of a manipulator, we represent $\boldsymbol{x}$ to be optimized as follows:

$$\boldsymbol{x} = \left(\ddot{x}_\mathrm{r}, \ddot{y}_\mathrm{r}, \ddot{\phi}_\mathrm{r}, \ddot{\theta}_1, \ddot{\theta}_2, \ldots, \ddot{\theta}_i\right)^\top. \tag{2}$$

B. Formulation with DAE

As detailed in Section III-B, to generate the robot motion in a single optimization calculation, the robot model, which is part of the constraint within the optimization problem, is represented using DAE. In this subsection, we explain the process of formulating the robot model represented by DAE.

The general form of DAE is expressed as follows:

$$F(\dot{\boldsymbol{y}}, \boldsymbol{y}, t) = 0. \tag{3}$$

DAE includes both differential and algebraic variables and can be considered an extension of ordinary differential equations. It is used in various fields to represent dynamic systems.

To realize the form of Eq. (3), we consider the differential kinematics of Eq. (4):

$$\dot{\boldsymbol{r}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{4}$$

where $\boldsymbol{r}$ is the robot hand position based on the mobile platform coordinates, and $\boldsymbol{q}$ is the joint angles. Both variables are variable vectors related to time t. $\boldsymbol{J}$ is the Jacobi matrix. Defining the variable matrix as $\boldsymbol{y} = (\boldsymbol{r}, \boldsymbol{q})^\top$ and the matrix $\boldsymbol{h}(\boldsymbol{y}) = (-\boldsymbol{I}, \boldsymbol{J})$ summarized by the unit matrix $\boldsymbol{I}$, Eq. (4) can be represented as follows:

$$\boldsymbol{h}(\boldsymbol{y})\dot{\boldsymbol{y}} = 0. \tag{5}$$

Since this equation is expressed by one function, i.e., Eq. (3), the robot model can be expressed by DAE.

It is widely known that solving DAE is more challenging compared with ordinary differential equations and algebraic equations. Consequently, in this study, DAE is discretized and solved through the following procedure [17]. Initially, when the task execution time is defined as T, it is divided into N finite elements specified in the range $[0, T]$ and discretized. Next, the specified $k-1$ points within the range of the i-th finite element are collocated by the Gauss–Radau method. This process yields a total of $N \times k$ discrete points. Finally, a Lagrange polynomial is generated through these points and treated as a discrete equation of the DAE. By obtaining discrete equation, it is possible to solve the DAE with constraints on differential variables, such as velocity and acceleration, and boundary conditions at times 0 and T.

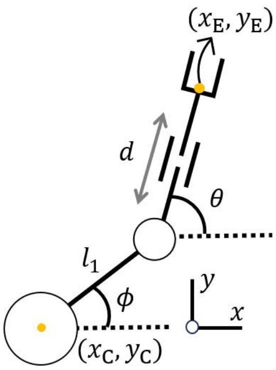


Figure 2. A planar robot example with five DoFs

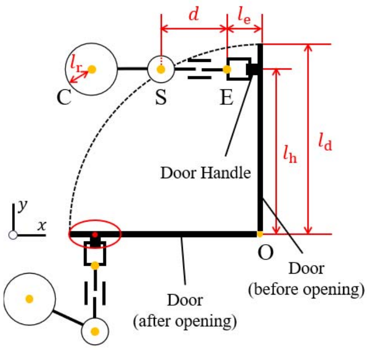


Figure 3. Parameters for door opening

C. Constraints

There are three main types of constraints on the optimization of the proposed method. The first type of constraints arises from the geometric relationships and motion parameters of the robot. Here, motion parameters refer to the variables, such as joint angles, angular velocities, and mobile platform velocity. These constraints include the definition of the robot model represented by DAE and the maximum and minimum values of the motion parameters. The second type of constraints arises from the target object. These include constraints that guide the robot in avoiding obstacles during movement. Additionally, if the object manipulated by the robot follows a predetermined trajectory, the motion of the robot hand is constrained by that trajectory. The third type of constraints arises from the restriction of robot motion at specific discrete times. These constraints come into play when the robot traverses a specified coordinate at a particular discrete time or when the robot stops within a specified discrete time range.

V. OPTIMIZATION PROBLEM FOR DOOR-OPENING TASK

A. Robot Model of Two-Dimensional 5DoFs Manipulator

To briefly describe the DAE model shown in Section IV-B, we show an example of a commonly used planar 5 DoFs mobile manipulator. This robot incorporates a rotation mechanism and a slider mechanism, as shown in Fig. 2. In the door-opening task for a hinged door, the position of the door handle typically remains constant in the vertical direction (z-direction) during the opening process. Consequently, we consider the task in a 2D plane.

The differential kinematics of the manipulator shown in Fig. 2 can be written as follows:

$$\begin{pmatrix} \dot{x}_\mathrm{E} \\ \dot{y}_\mathrm{E} \end{pmatrix} = \begin{pmatrix} -l_1 \sin\phi & \cos\theta & -\mathrm{d}\sin\theta \\ l_1 \cos\phi & \sin\theta & \mathrm{d}\cos\theta \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{d} \\ \dot{\theta} \end{pmatrix} + \begin{pmatrix} \dot{x}_\mathrm{C} \\ \dot{y}_\mathrm{C} \end{pmatrix}, \tag{6}$$

where $(x_\mathrm{E}, y_\mathrm{E})$ is the position of the robot hand. $(x_\mathrm{C}, y_\mathrm{C})$ and ϕ are the position and orientation of the mobile platform, respectively. d is the length of the manipulator extended by the slider mechanism, θ is the angle of the rotation mechanism, and l_1 is the length of the first link of the manipulator. Let $(\dot{x}_{\mathrm{CE}}, \dot{y}_{\mathrm{CE}})$ be the relative velocity of the hand based on the mobile platform. Transforming it as in Section IV-B, we obtain Eq. (7) which corresponds to Eq. (3),

$$\begin{pmatrix} -1 & 0 & -l_1 \sin\phi & \cos\theta & -\mathrm{d}\sin\theta \\ 0 & -1 & l_1\cos\phi & \sin\theta & \mathrm{d}\cos\theta \end{pmatrix} \begin{pmatrix} \dot{x}_{\mathrm{CE}} \\ \dot{y}_{\mathrm{CE}} \\ \dot{\phi} \\ \dot{d} \\ \dot{\theta} \end{pmatrix} = 0. \quad (7)$$

In this section, we provide a concise description using a robot with low DoFs. The door-opening task can also be executed by a robot with a high number of DoFs, as demonstrated in previous studies [7]. In such instances, the DAE model can be obtained by expanding the number of variables in Eq. (6) and applying the same transformation.

B. Motion Parameter Constraints

The constraints expressing the limit values of the robot motion parameters can be written as follows:

$$\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \leq \boldsymbol{0}, \quad (8)$$

where $\boldsymbol{A}$ is represented as $\boldsymbol{A} = (\boldsymbol{I}, -\boldsymbol{I})^{\top}$ using the unit matrix $\boldsymbol{I}$, $\boldsymbol{x}$ is the robot's motion parameter, and $\boldsymbol{b}$ is represented as $\boldsymbol{b} = (\boldsymbol{x}_{\max}, \boldsymbol{x}_{\min})^{\top}$ using the vector $\boldsymbol{x}_{\max}$ of the maximum values and vector $\boldsymbol{x}_{\min}$ of the minimum values.

C. Door Constraints

In Section III-A, we divided the two types of door-opening tasks: "door opening for others" and "door opening for itself." The latter involves both opening the door and moving to the opened area. Here, we introduce the constraints associated with the door's structure. Let t_1 be the execution time for opening the door, and t_2 be the execution time for moving to the opened area. The optimization calculation is considered in a 2D plane; for example, the coordinate of point O, as shown in Fig. 3, is defined as $(x_{\mathrm{O}}, y_{\mathrm{O}})$.

(1) Constraints due to door trajectory

During the execution of the door-opening task, since the robot grasps the door handle to open, the trajectory of the robot hand follows the same circular arc as that of the handle. Therefore, the constraints are as follows:

$$\|\boldsymbol{r}_{\mathrm{E}} - \boldsymbol{r}_{\mathrm{O}}\|^2 = l_{\mathrm{h}}^2 + l_{\mathrm{e}}^2, \quad (9)$$

$$\|\boldsymbol{r}_{\mathrm{S}} - \boldsymbol{r}_{\mathrm{O}}\|^2 = (d + l_{\mathrm{e}})^2 + l_{\mathrm{h}}^2, \quad (10)$$

where $\boldsymbol{r}_{\mathrm{E}}$ is the coordinate of the robot hand, $\boldsymbol{r}_{\mathrm{S}}$ is the coordinate of the robot's wrist, and $\boldsymbol{r}_{\mathrm{O}}$ is the coordinate of the door rotation center. Each variable is represented by a 2D vector. d is the distance from the hand to the wrist, l_{e} is the distance from the hand to the door handle, and l_{h} is the distance from the door rotation center to the door handle. The reason for including the trajectory of not only the robot hand but also the robot's wrist in the constraints serves the purpose of averting the possibility of the robot hand slipping off the door handle. This is achieved by maintaining the robot hand perpendicular to the door, with the wrist securely attached to the hand. These constraints are applicable within the execution time interval $[0, t_1]$.

(2) Constraints due to Movement into the Opened Area

In the task of opening a door for itself, when moving to the opened area, it is critical to avoid colliding with the door. Given that the door handle protrudes from the door, it should also be regarded as an obstacle. Considering these points, we define the following constraints. They are considered within the execution time $[t_1, t_1 + t_2]$.

The first constraints pertain to collision avoidance with the door. Based on the definition shown in Fig. 3, the constraints are as follows:

$$\|\boldsymbol{r}_{\mathrm{C}} - \boldsymbol{r}_{\mathrm{O}}\|^2 \geq (l_{\mathrm{d}} + l_{\mathrm{r}})^2, \quad (11)$$

$$\|\boldsymbol{r}_{\mathrm{E}} - \boldsymbol{r}_{\mathrm{O}}\|^2 \geq l_{\mathrm{h}}^2, \quad (12)$$

where $\boldsymbol{r}_{\mathrm{C}}$ is the coordinate of the mobile platform, l_{d} is the door width, and l_{r} is the radius of the mobile platform as a circle.

Furthermore, to prevent the robot hand from colliding with the door handle, we contemplate an elliptical region around the door handle, as shown in Fig. 3. The following equation is employed for this purpose:

$$\left(\frac{x_{\mathrm{E}} + l_{\mathrm{h}}}{l_{\mathrm{d}} - l_{\mathrm{h}} + l_{\mathrm{o}}}\right)^2 + \left(\frac{y_{\mathrm{E}}}{l_{\mathrm{e}}}\right)^2 \geq 1. \quad (13)$$

This represents an elliptical, with the longest side as $l_{\mathrm{d}} - l_{\mathrm{h}} + l_{\mathrm{o}}$ and the shortest side as l_{e}. l_{o} represents the offset that indicates how much clearance the robot hand has to avoid the door handle.

(3) Constraints due to final standing position

As a constraint on the final standing position of the robot, we initially consider a constraint based on the endpoint of the door-opening task. Assuming an effective angle of rotation for the hinged door as 90°, we define the following equation at discrete time t_1:

$$y_{\mathrm{E}} = y_{\mathrm{O}} - l_{\mathrm{e}}. \quad (14)$$

As shown in Fig. 1, the final destination position of the robot differs between door opening for others and door opening for itself. Therefore, we establish constraints on the final position for each case. In the case of door opening for others, the final position should avoid intruding into the opened area to prevent interference with others' actions. Hence, at discrete time t_1, Eq. (15) should be satisfied:

$$y_{\mathrm{C}} + l_{\mathrm{r}} \leq 0. \quad (15)$$

On the other hand, in the case of door opening for itself, the final destination position should be near the opened area to facilitate the next action. Therefore, at discrete time $t_1 + t_2$, Eq. (16) should be satisfied:

$$y_{\mathrm{C}} - l_{\mathrm{r}} \geq 0. \quad (16)$$

As shown in these equations, constraints can be applied in specific discrete times to restrict the robot's position and motion.

VI. EXPERIMENT

A. Experimental Settings

The proposed method's efficacy was validated through simulations and experiments using a life support robot known as HSR, manufactured by Toyota Motor Corp. The HSR is an 8-DoF robot, comprising an omnidirectional mobile platform (3-DoFs) and a 4-axis manipulator with a lifting function (1-DoF). It is Equipped with an RGBD camera in the head, a 3-axis force sensor in the hand, and an LRF on the mobile platform. In the experiment, the RGBD camera detected a door handle, the force sensor compensated for the door opening trajectory, and the LRF was used by the robot's self-positioning. A detailed description is given later.

The motion generation program was implemented in Python, using Pyomo [18] as the modeling tool for the optimization problem and IPOPT [19] as the solver for optimization calculations. The computer specifications used in the experiments were CPU: Intel (R) Core (TM) i9-9940X 3.30GHz, GPU: NVIDIA GeForce RTX 2080 Ti.

The proposed method addresses the door-opening task as a motion generation problem in a two-dimensional plane. For this setting, the HSR was first represented as a planar model, as shown in Fig. 4. The mobile platform (C_1) involved three motion parameters (x, y, ϕ), and the manipulator featured two DoFs with a rotation axis and a cylinder mechanism. The lifting and lowering of the manipulator were computed based on the original manipulator length and the distance of the extended manipulator, utilizing the Pythagorean theorem.

In simulations, the door width ranged from 0.3 m to 0.8 m in 0.1 m increments, preparing six different door sizes. For the actual robot experiment, doors with widths of 0.3 m and 0.6 m were used. The door handle's position was detected by attaching an AR marker near the handle and detecting it using a camera mounted on the robot's head.

In each motion generation trial, the door width and handle position were predetermined. The robot's initial posture was set in front of the door, a short distance away from the door handle, so that the AR marker was within the camera's field of view. The robot detected the handle on the spot before grasping it and then initiated the optimization calculation. After optimization calculation, the calculated trajectory was transformed from the optimization calculation's coordinate system to the world coordinate system to obtain the ideal trajectory. The execution time of the door-opening motion was manually set to 10 s for door opening for others and 14 s for door opening for itself, considering the HSR's movable speed. In the case of the door opening for itself, the motion releasing the robot hand from the door handle was set when the door-opening phase concluded. Discrete points were set every 1 s, and the point selection method was set to three.

B. Correction of Errors in Real Environment

In the actual robot's self-positioning, a combination of wheel-based odometry and hector-SLAM [20], which estimates the robot's movement solely based on LRF data, was employed. Nevertheless, self-positioning may entail minor posture errors. Furthermore, an error in the handle position detection and depending on the type of hinge on the door, the center of rotation may not consistently be in the same position when the door is opened. These factors can result in a load being placed on the robot's hand even if the robot is moved according to the generated motion. To avoid this problem, offset values $\boldsymbol{x}_{\text{offset}}$ is successively added to the trajectories of the mobile platform calculated by optimization. The offset value is calculated as follows:

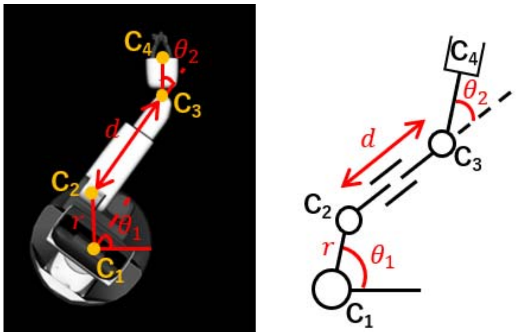


Figure 4. The HSR model in two-dimensional plane

$$\boldsymbol{x}_{\text{offset}} = \boldsymbol{W}\boldsymbol{f}, \tag{17}$$

where $\boldsymbol{W}$ is a third-order real diagonal matrix that stores the weights of each component in the diagonal components. $\boldsymbol{f}$ is a three-dimensional vector and each component f^j $(j = 1, 2, 3)$ is calculated based on each component value of the force sensor f_{now}^j using the following formula:

$$f^j = \begin{cases} 0 & if\ f_{\text{now}}^j \leq \left| f_{\text{t}}^j \right| \\ f_{\text{t}}^j - f_{\text{now}}^j & \text{otherwise}, \end{cases} \tag{18}$$

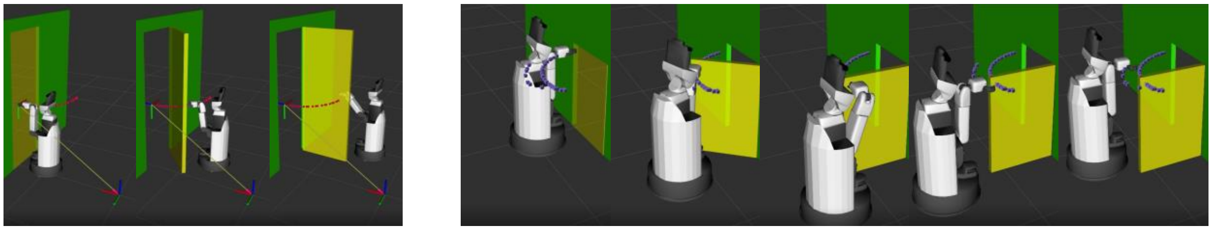
Here, f_{t}^j is a given force threshold for each component. From Eq. (18), $\boldsymbol{f}$ serves to encourage the robot hand to correct the trajectory in the opposite direction when the force exceeds the sensor's threshold set by us is applied.

VII. RESULTS AND DISCUSSION

A. Results Using a Simulator

Fig. 5 shows the simulation of the door-opening motion. In Fig. 6, the graphs present the trajectories of the robot's hand and the mobile platform. The trajectories obtained in the planning are depicted dotted lines, while the trajectories observed during the execution of the door opening are depicted solid lines. The origin in Fig. 6 corresponds to the initial position of the robot. Additionally, the start and goal points in Fig. 6 signify the start and end points, respectively, of the trajectory calculated using optimization calculations. The horizontal axis of each graph represents the x coordinate and the vertical axis represents the y coordinate.

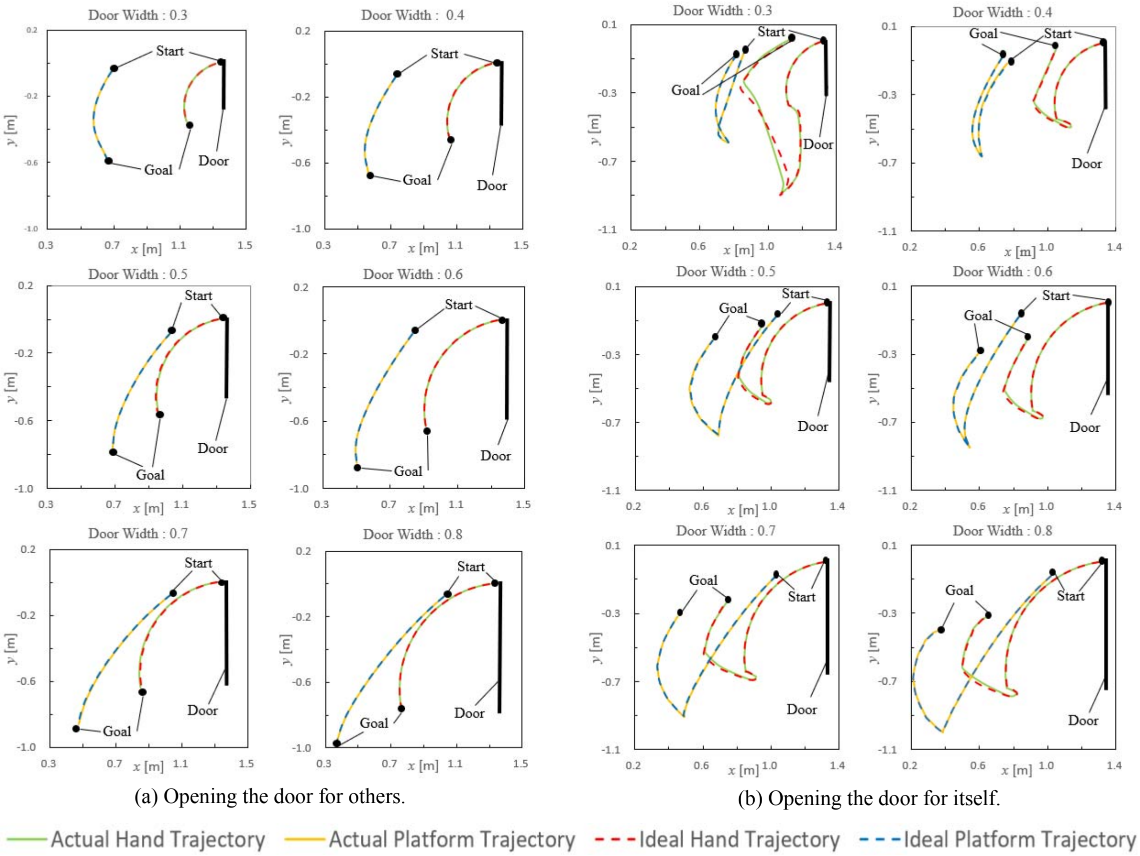
Fig. 6(a) shows that in the case of door opening for others, the planned trajectories almost align with the trajectories during motion execution. Conversely, in the case of opening a door for itself, Fig. 6(b) shows that the trajectories toward the opened area are generally consistent, albeit with a



(a) Opening the door for others.

(b) Opening the door for itself.

Figure 5. Door opening motion executed in simulator.



(a) Opening the door for others.

(b) Opening the door for itself.

Figure 6. Trajectories of hand and mobile platform at simulation.

discrepancy of several tens of millimeters between the planned trajectory and the executed result. This misalignment occurred specifically when the robot moved to the opened area while folding its outstretched arm after releasing the handle. The reason was that the distance between discrete points during motion generation was larger than that between other motion points. Resolving this issue entails potential solutions, such as increasing the number of discrete points and extending the execution time of the motion.

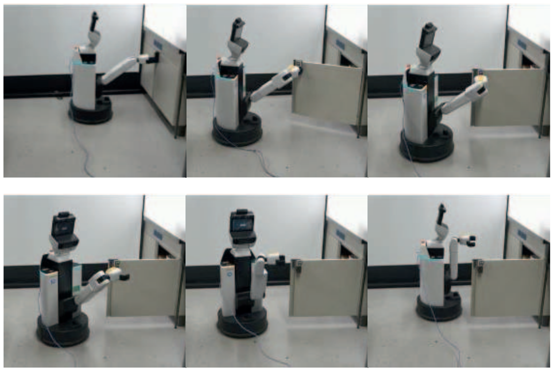
B. Results in Real Environment

Fig. 7 shows an example of actual experiments, while Fig. 8 provides graphs presenting the trajectories of the robot's hand and the mobile platform. The trajectories are delineated for both obtained in the planning (dotted lines) and observed during the execution (solid lines). In Fig. 8, the origin signifies the initial position of the robot, while the start and goal points denote the start and end points of the calculated trajectory, which is similar to Fig. 6. The solid line representing the mobile platform in Fig. 8 reflects results subtracting the offset values using the force sensor described in Section VI-C. The comparison indicates that planned trajectories are the same as the trajectories when opening the door. However, disparities emerge in hand positions when moving to the opened area. Notably, the mobile platform's positions exhibit minimal deviation from the ideal trajectory, suggesting that errors during the arm-folding motion persist and contribute to deviations from the ideal motion during execution, as observed in the simulator.

VIII. CONCLUSIONS

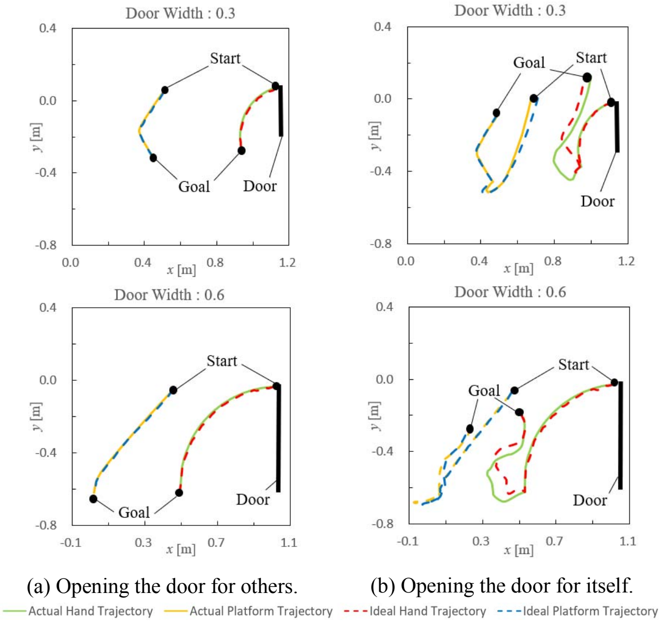
In this paper, we presented a method for generating robot motion specifically tailored for door-opening tasks. We proposed a method to achieve the generation of a robot motion in a single computation without dividing the door-opening task into subproblems. For this purpose, we

(a) Door-opening for others.



(b) Door opening for itself.

Figure 7. Door opening motion executed in real environment.



(a) Opening the door for others.

(b) Opening the door for itself.

Figure 8. Trajectories of hand and mobile platform on actual robot.

proposed a DAE model derived from the robot's differential kinematics and presented the formulation of a nonlinear optimization problem considering both velocity and acceleration. Constraints, including via points of the robot and collision avoidance, are integrated into this formulation. The effectiveness of the proposed method is validated through the execution of door-opening tasks in both simulated and real environments. The experiments involved various types of hinged doors, affirming that the method can generate appropriate motions for both cases: opening doors for others and door opening for itself. Furthermore, we addressed the modification of robot posture errors that might manifest in real-world environments.

Future works will focus on expediting the optimization process. Additionally, we plan to showcase the versatility of the proposed method by applying it to tasks beyond door opening, leveraging its inherent characteristics.

ACKNOWLEDGMENT

This work was partially supported by JST [Moonshot R&D][Grant Number JPMJMS2034]

REFERENCES

[1] K. Nagatani and S. Yuta, "Door Opening Behavior of an Autonomous Mobile Manipulator by Sequence of Action Primitives," Journal of Robotic Systems., Vol. 13(11), pp. 709 – 721, 1996.

[2] H. Sasaki, T. Takahashi and E. Nakano "A Door Opening Method by a Mobile Manipulator with Passive Joints," Journal of the Robotics Society of Japan, vol. 19(2), pp. 277 – 284, 2001.

[3] K. Nagahama et al., "A Learning Method for a Daily Assistive Robot for Opening and Closing Doors Based on Simple Instructions," in Proc. of IEEE Int'l Conf. on Automation Science and Engineering, pp. 599 – 605, 2018.

[4] M. Mittal et al., "Articulated object interaction in unknown scenes with whole-body mobile manipulation," arXiv:2103.10534, 2021.

[5] S. Chitta, B. Cohen and M. Likhachev, "Planning for Autonomous Door Opening with a Mobile Manipulator," in Proc. of IEEE Int'l Conf. on Robotics and Automation, pp. 1799 – 1806, 2010.

[6] S. Gu et al., "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in Proc. of IEEE Int'l Conf. on Robotics and Automation, pp. 3389 – 3396, 2017.

[7] H. Ito et al., "Efficient Multitask Learning with an Embodied Predictive Model for Door Opening and Entry with Whole-Body control," Science Robotics, vol. 7(65), 2022.

[8] Y. Mo et al., "Multi-step motion Learning by combining learning-from-demonstration and policy-search," Advanced Robotics, Vol. 37(9), pp. 560 – 575, 2023.

[9] J. Yamada et al., "Motion Planner Augmented Reinforcement Learning for Robot Manipulation in Obstructed Environments," Proceedings of Machine Learning Research, Vol. 155, pp. 589 – 603, 2021.

[10] J. Schulman et al., "Motion planning with sequential convex optimization and convex collision checking," Int'l Journal of Robotics Research, Vol. 33(9), pp. 1251 – 1270, 2014.

[11] M. Zucker et al, "CHOMP: Covariant Hamiltonian optimization for motion planning," Int'l Journal of Robotics Research, Vol. 32, pp. 1164 – 1193, 2013.

[12] M. Murooka et al., "Robot Motion Generation by Time-series Inverse Kinematics Optimization Considering Time-variance/Time-invariance and Adjacency of Configuration," SICE Journal of Control, Measurement, and System Integration, Vol. 55(11), pp. 664 – 673, 2019.

[13] L. Petrovic, I. Markovic and I. Petrovic, "Mixtures of Gaussian Processes for Robot Motion Planning Using Stochastic Trajectory Optimization", in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 52(12), pp. 7378 – 7390, 2022.

[14] R. C. Castello, R. Grino and L, Basanez, "DAE Methods in Constrained Robotics System Simulation," Computacion y Sistemas, Vol. 1(3), pp. 145 – 160, 1998.

[15] J. Yang et al., "A modular approach for dynamic modeling of multisegment continuum robots," Mechanism and Machine Theory, Vol. 165, 2021, 104429.

[16] S. Wang and S. Lu, "Dynamic Optimization of Robot Automatic Control System Based on Differential Algebraic Equations," Applied Mathematics and Nonlinear Science, Vol. 8(1), pp. 3149 – 3158, 2023.

[17] B. Nicholson et al., "pyomo.dae: a modeling and automatic discretization framework for optimization with differential and algebraic equations," Mathematical Programming Computation, Vol. 10, pp. 187 – 223, 2018.

[18] Pyomo : https://pyomo.readthedocs.io/en/stable/index.html

[19] Ipopt : https://coin-or.github.io/Ipopt/

[20] S. Kohlbrecher et al., "A flexible and scalable SLAM system with full 3D motion estimation," in Proc. of IEEE Int'l Symp. on Safety, Security and Rescue Robotics, pp. 155 – 160, 2011.