Whole-body motion planning of dual-arm mobile manipulator for compensating for door reaction force

Yujin Kim[1] and Keunwoo Jang[1]

Abstract— **Door traversal is one of the representative benchmark tasks for mobile manipulators. However, it is still challenging since it requires to compute the coupled motion of the whole-body of the robot and door. Furthermore, in the case of heavy doors, not only kinematic constraints should be considered, but also the reaction force from the door should be considered for the balance of the robot. This work proposes a framework that plans the whole-body motion of dual-arm mobile manipulator utilizing dual-arm contact for traversing through the door. The framework is based on our previous work [1] which generates the whole-body motion of single-arm mobile manipulator to approach, open, pass through, and close the door. The proposed framework first computes the path for the pose of the mobile robot, the path for the door angle, and the path for the contact point by using the graph search algorithm. In graph search, an integer value called *contact pair* is introduced as an element of the state, which indicates where each arm contacts with the door or the environment. Then, the framework calculates the configuration of the arm by using inverse kinematics solver. The effectiveness of the proposed framework was validated through experiments in simulation environment.**

I. INTRODUCTION

A mobile manipulator can perform more various and complex tasks than a fixed-base manipulator thanks to the mobility offered by a mobile robot. To maximize the performance, the mobile manipulator should be able to open and pass through doors that act as connectors between spaces. However, it remains challenging to generate the whole-body motion of the robot because the mobile manipulator is a unified system and the door is an articulated object.

To address the problem, many researches in different categories have been developed in recent decades: *sense-and-act*, *plan-and-act*, *sequential sense-plan-and-act*, and *learn-and-act*. In the field of *sense-and-act* approaches [2], [3], [4], they generate an online whole-body motion that opens the door based on estimated geometric information, including the door, from sensory feedback. However, since these approaches produce the motion based on the current state, they may get stuck in local minima and fail to open. The *plan-and-act* approaches [5], [6], [7], [8], [9] plan the motion that opens the door given the information for the environment and door. They are suitable for well-defined and structured environment, but are vulnerable to accumulated errors caused by uncertainty and changes in the environment. To overcome this issue, the *sequential sense-plan-and-act* approaches [10], [11] have been developed. They have cascaded architecture composed of an outer loop with updating the reference motion at a low rate and an inner loop with tracking controller, such as model predictive control (MPC) framework. Recently, the *learn-and-act* approaches [12], [13] utilized learning-based method, such as convolutional neural network and behavior cloning, to generate robust motion in response to environmental change and uncertainty instead of modeling and measuring complex real-world environment.

Our previous work [1], which can belong to *plan-and-act*, *sequential sense-plan-and-act* approach, proposed a framework that computes a path for the whole-body configuration of the mobile manipulator to navigate from the initial position, traverse through the door, and arrive at the target position. Rather than searching for the entire joint space of the mobile manipulator, the proposed framework is designed to sequentially execute two steps denoted as ***S1*** and ***S2*** which generate the motion for the mobile robot and manipulator, respectively. The first step ***S1*** generates the collision-free motion of the mobile robot and door angle. Next, the second step ***S2*** computes the synchronized motion of the manipulator for the generated motion of the mobile robot and door in ***S1***. In ***S2***, the motion of the manipulator is computed by using the inverse kinematics (IK) solver.

In this work, a framework is proposed that extends our previous work [1] to generate the whole-body motion of dual-arm mobile manipulator utilizing the contact of the dual-arm. The proposed framework also has two steps composed of ***S1*** and ***S2***. In ***S1***, the path for contact points of dual-arm is additionally generated by graph search. Especially, an element of the state in graph representation is introduced, called *contact pair*, has an integer value that depends on the combination of positions where each arm is in contact with the environment or the door. Then, the second step ***S2*** computes the configuration of the dual-arm by using IK solver as the results from the first step include the contact position, the pose of mobile robot and the door angle.

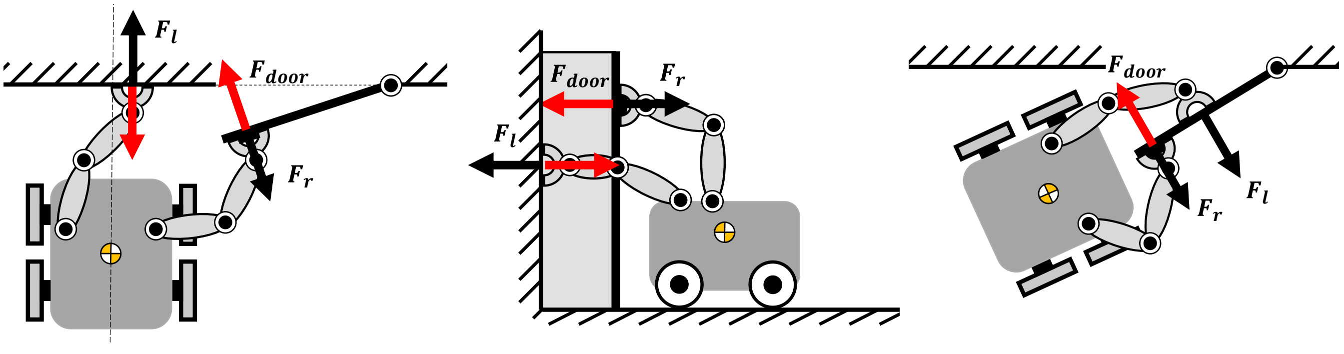
II. PROBLEM STATEMENT

This section demonstrates the motion planning problem of the dual-arm mobile manipulator navigating from the start position, traversing through a door, and reaching the target position. Consider the reaction force of the door on the robot when the robot is opening a heavy or spring-loaded door. In this situation, the reaction force affects the balance of the robot. To address the issue, the dual-arm mobile manipulator can utilize the other arm that is not opening the door by contacting the environment or the door in order

---

*This work was supported by KIST Institutional Program (No. 2E33004, 2E32985).

[1] Yujin Kim and Keunwoo Jang are affiliated with Artificial Intelligence and Robot Institute, Korea Institute of Science and Technology, Seoul, South Korea. `jang90@kist.re.kr`



(a) (b) (c)

Fig. 1. Illustration of advantages for the dual-arm mobile manipulator to utilize both arms when passing through the door. F_r, F_l, and F_{door} denote the force exerted by the right arm, the force by the left arm, and the reaction force from the door, respectively. (a) Top view of the robot and door. The moment generated by the reaction force can be compensated by the contact force of the left arm. (b) Side view of the robot and door. The decrease in ground reaction force caused by the reaction force can be compensated by the contact force of the left arm. (c) Top view of the robot and door. By pushing the door using the dual-arm, the reaction force can be compensated.

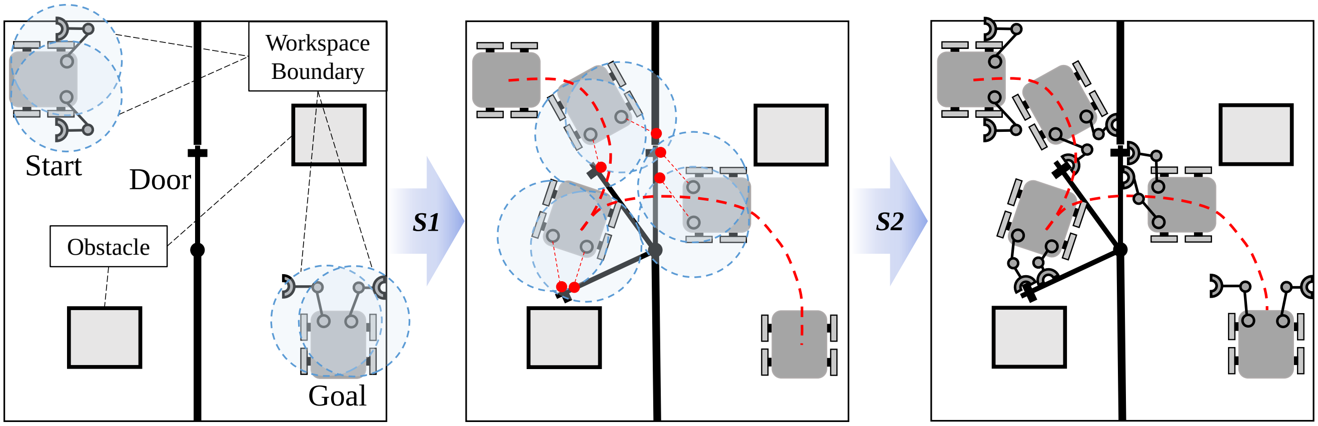


Fig. 2. Overview of the proposed framework. Given the information about the environment, the door, and the start and goal positions, ***S1*** is the step that computes the pose path of the mobile robot, angle path of the door, and the sequences of contact. The paths are computed with the constraint that the position of the door handle or the environment, such as wall, has to be located inside the workspace of the dual-arm as shown in blue dotted circles. Red dotted line shows the computed path of the mobile robot and red circles indicate the contact points for the dual-arm. Then, ***S2*** is the step that computes the path of the joint configuration of the dual-arm by using the inverse kinematics solver. Combining the computed paths, the motion of the dual-arm mobile manipulator is generated that the robot approaches, opens, traverses through, closes the door, and eventually arrives at the goal position.

to compensate for the reaction force, as shown in Fig. 1. This allows the dual-arm mobile manipulator to maintain balance of the robot and overcome the hardware limitation of having only one arm. However, the robot should decide whether the other hand will make contact with the door or the environment, where to make contact, and when to change the contact position. Thus, this work addresses the motion planning problem of the dual-arm mobile manipulator to find a collision-free path of the whole-body configuration that compensates for the reaction force of the door.

III. PROPOSED METHOD

This section describes the proposed framework which computes the collision-free path of the dual-arm mobile manipulator traversing through the door and thereafter reaching the goal position. The proposed framework is designed to sequentially execute two steps denoted as ***S1*** and ***S2*** which generate the motion for the mobile robot and dual-arm, respectively. As shown in Fig. 2, the first step ***S1*** generates a collision-free path for the mobile robot, a path for the door angle, and a sequence of contact points. Next, the second step ***S2*** computes the joint path of the dual-arm for the result from ***S1***. In ***S2***, the motion of the dual-arm is computed by using the IK solver.

*A. Computing paths for mobile robot, door angle, and contact pair - **S1***

The navigation problem of the mobile robot considering the contact pair for the dual-arm is formulated as a graph search. In the following, the elements of the graph are described.

1) State: The search space $\mathcal{S} \in \mathbb{R}^3 \times \mathbb{I}^2$ is five dimensional space. The state is expressed as $s = (x, y, \theta, a, c) \in \mathcal{S}$. (x, y, θ) is the pose of the mobile robot, a is called *area indicator*, and c is called *contact pair*. Compared to our previous work, the contact pair c, which has an integer value from 0 to 2, is added to the search space to define the contact for the dual-arm. As shown in Fig. 3, c is defined as 1 when

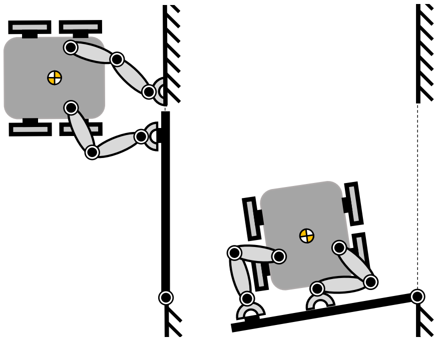

(a) (b)

Fig. 3. Illustration of the contact pair c. (a) when each arm contacts with the door and the environment, respectively, c is defined as 1. (b) when both arms contact with the door together, c is defined as 2.

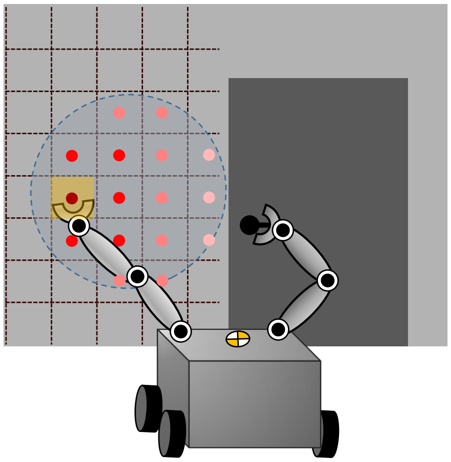


Fig. 4. Illustration of determining the contact point for the left arm when the contact pair is 1.

one arm is in contact with the door and the other arm is in contact with the environment. When both arms are in contact with the door, c is defined as 2. The value of c is 0 when the robot is approaching the door from the start position and when the robot is placed between the door and goal position.

In the contact pair, the details of determining the position of the contact point are described as follows. Assume that the robot is located so that at least one arm is within reach of the door handle. Then, as shown in Fig. 4, the common area is computed between a sphere with the radius of the workspace boundary and the plane of the environment or the door. Since the common area is a two-dimensional plane, the area is divided into a set of cells with a predefined interval. All center points of the cells are selected as possible points in contact with the environment or the door. Each point is scored with a compensating moment for the door reaction force, assuming the reaction force is a fixed value. Among the possible points, the point with the highest compensating moment is selected as the contact point.

If there is no point in the common area that can compensate for the door reaction force, the state is not generated. Also, since the area indicator represents the range of possible door angles in the current pose, the contact pair can be defined differently depending on the value of the area indicator, even for the same pose of the mobile robot. Then, the states with the same pose of the mobile robot, but with different area indicators and contact pairs are generated.

2) Action: To define the successor of any state, the action should be defined. Since the area indicator and the contact pair are variables determined by the pose of the mobile robot, the action is defined only for the pose of the mobile robot. Thus, the pose of the mobile robot is discretized by utilizing the concept of the *lattice* [14] and the feasible actions are defined as the connection between the discretized poses. The graph can be constructed by executing the actions to the states.

3) Cost: The transition cost from the state to its successor state, denoted as $c(s, s_{succ})$, is defined as

$$c(s, s_{succ}) = c_{action} + c_{map} + c_{door} + c_{contact} \tag{1}$$

where c_{action} is the cost to transit to the successor state by executing the predefined action of the mobile robot. This term is proportional to the weighted sum of the translational and rotational displacement. Second, c_{map} is the cost based on the costmap which indicates how close the robot is to the obstacle [5]. Next, c_{door} is the cost to represent how properly the door handle is positioned from the base of the manipulator at the given pose of the mobile robot when the area indicator of the current state includes from 1 to 3. c_{door} is defined as

$$c_{door} = K(d - d_{min})^2, \tag{2}$$

where K is the positive coefficient and d is the distance between the base of the manipulator and door handle at the current pose of the mobile robot, and d_{min} is the distance determined by the reachability data. Lastly, $c_{contact}$ is the cost when the contact pair changes from 1 to 2 or vice versa. This term allows the robot to maintain the current contact pair.

To guide the search for the graph, the heuristic should be defined. The heuristic in this work is defined as the same as in our previous work.

4) Search: Based on the constructed graph, the optimal path can be found by the search algorithm. The search algorithm used in this work is Anytime Repairing A* (ARA*) [15], which quickly finds the initial sub-optimal path, then refines it towards the optimal path for the remaining planning time. Also, the solution path can be obtained by using the other variants of A* algorithm.

*B. Computing path for dual-arm joint configuration - **S2***

In order to compute the path for the joint position of the dual-arm, the paths for the position of the door handle and the position of the contact are computed from the paths for the door angle and contact pair obtained in ***S1***, respectively. Then, the joint position of the dual-arm is computed by the IK solver at the corresponding pose of the mobile robot.

Consequently, the whole body motion of the dual-arm mobile manipulator is generated by combining the path for

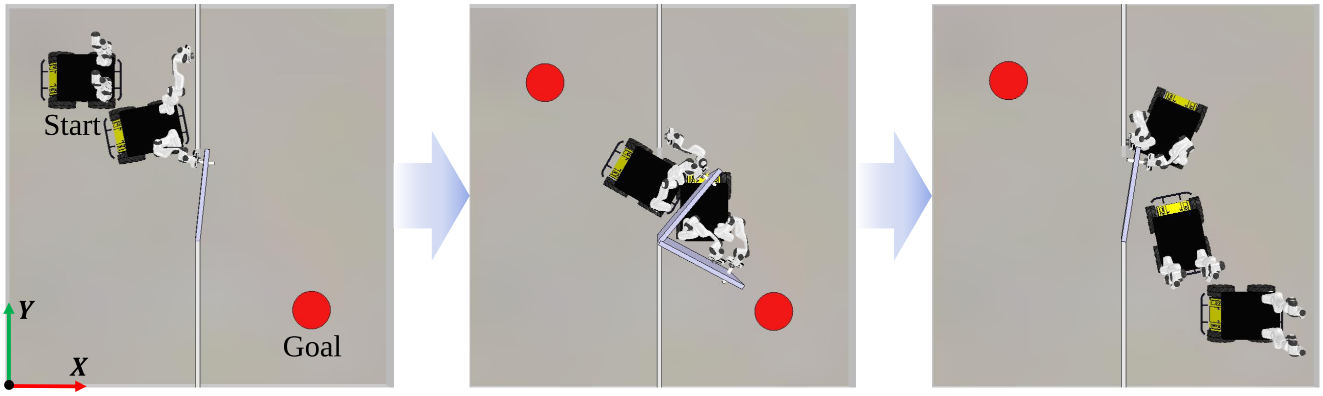


Fig. 5. Snapshots of simulation from the proposed framework for the push-type door. Red circles indicate the start and goal position.

the pose of the mobile robot, obtained in ***S1***, and the path for the joint position of the dual-arm, obtained in ***S2***.

IV. RESULTS

The proposed framework was validated in simulation environment using the differentially-driven mobile manipulator consisting of the four-wheel mobile base *Husky* (*Clearpath Robotics. Co.*) and the 7-DOFs manipulator *Panda* (*Franka Emika. Co.*). The specification of the computer is $i7$ 3.4 GHz with 16 GB RAM. The external libraries were utilized, including SBPL for the graph search algorithm in ***S1*** and TRAC-IK for computing IK solution in ***S2***.

The proposed framework was applied to the push-type door in simulation environment. The snapshots are shown in Fig. 5. The left snapshot in Fig. 5 shows the process of the robot approaching and opening the door. The middle snapshot shows that the robot traverses through the door. In this snapshot, the contact pair is changed from 1 to 2 when passing through the door. The right snapshot shows the robot closing the door and reaching the target position. The inputs to the framework were the start state $s_0 = (1.0, 4.0, 0.0, 0, 0)$ and goal state $s_g = (4.0, 1.0, 0.0, 4, 0)$. The door length is set as 1.0 m. The computation time was recorded as 9.34 s for ***S1*** and 1.71 s for ***S2***.

V. CONCLUSIONS AND FUTURE WORK

This paper presents a framework for solving the navigation problem of a dual-arm mobile manipulator utilizing the dual-arm to compensate for the reaction force of a door. The framework computes the whole-body path in two steps: the first step computes the paths for the mobile robot's pose and contact points, and the second step computes the joint paths of the dual-arm. In particular, the first step utilizes a graph search method to find the optimal path, introducing an integer variable called contact pair to represent the contact pair of the dual-arm in the states of the graph. The effectiveness of the proposed framework was demonstrated through the simulation with the differentially-driven dual-arm mobile manipulator.

Since the reaction force of the door is assumed to be a fixed value, the uncertainty can be handled by a real-time controller that reflects the current state, such as MPC. In this respect, the proposed framework can provide a path for the methods that require nominal or reference paths, such as MPC or learning-based methods. Also, although the framework was applied to wheeled mobile manipulators in this paper, the framework can be applied to legged mobile manipulators, humanoids as well, since the planar paths computed by the framework can be considered as planar paths of the humanoid trunk. On the other hand, since the proposed framework computes the whole-body path considering static equilibrium, in order to apply the framework in the real world, future work includes transforming the computed path into a trajectory over time considering the whole-body dynamics using a method such as trajectory optimization. Future work also includes extending the framework to consider contact transitions in dynamic situations.

REFERENCES

[1] K. Jang, S. Kim, and J. Park, "Motion planning of mobile manipulator for navigation including door traversal," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 4147–4154, 2023.

[2] J. Lee, A. Ajoudani, E. M. Hoffman, A. Rocchi, A. Settimi, M. Ferrati, A. Bicchi, N. G. Tsagarakis, and D. G. Caldwell, "Upper-body impedance control with variable stiffness for a door opening task," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 713–719.

[3] Y. Karayiannidis, C. Smith, F. E. V. Barrientos, P. Ögren, and D. Kragic, "An adaptive control approach for opening doors and drawers under uncertainties," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 161–175, 2016.

[4] M. Stuede, K. Nuelle, S. Tappe, and T. Ortmaier, "Door opening and traversal with an industrial cartesian impedance controlled mobile robot," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 966–972.

[5] S. Chitta, B. Cohen, and M. Likhachev, "Planning for autonomous door opening with a mobile manipulator," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 1799–1806.

[6] S. Gray, S. Chitta, V. Kumar, and M. Likhachev, "A single planner for a composite task of approaching, opening and navigating through non-spring and spring-loaded doors," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 3839–3846.

[7] M. Arduengo, C. Torras, and L. Sentis, "Robust and adaptive door operation with a mobile robot," *Intelligent Service Robotics*, vol. 14, pp. 409–425, 2021.

[8] F. Burget, A. Hornung, and M. Bennewitz, "Whole-body motion planning for manipulation of articulated objects," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 1656–1662.

[9] D. Leidner, A. Dietrich, F. Schmidt, C. Borst, and A. Albu-Schäffer, "Object-centered hybrid reasoning for whole-body mobile manipulation," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1828–1835.

[10] D. Lee, H. Seo, D. Kim, and H. J. Kim, "Aerial manipulation using model predictive control for opening a hinged door," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1237–1242.

[11] G. Rizzi, J. J. Chung, A. Gawel, L. Ott, M. Tognon, and R. Siegwart, "Robust sampling-based control of mobile manipulators for interaction with articulated objects," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1929–1946, 2023.

[12] H. Ito, K. Yamamoto, H. Mori, and T. Ogata, "Efficient multitask learning with an embodied predictive model for door opening and entry with whole-body control," *Science Robotics*, vol. 7, no. 65, 2022.

[13] H. Xiong, R. Mendonca, K. Shaw, and D. Pathak, "Adaptive mobile manipulation for articulated objects in the open world," *arXiv preprint arXiv:2401.14403*, 2024.

[14] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 933–945, 2009.

[15] M. Likhachev, G. J. Gordon, and S. Thrun, "Ara$*$: Anytime a$*$ with provable bounds on sub-optimality," in *Advances in Neural Information Processing Systems*, S. Thrun, L. Saul, and B. Schölkopf, Eds., vol. 16. MIT Press, 2003.