IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 March 2024, accepted 13 April 2024, date of publication 14 May 2024, date of current version 5 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3400854

RESEARCH ARTICLE

Motion Planning of Mobile Manipulator Using Virtual Impedance Energy Field

JUNG HYUN CHOI[1], (Graduate Student Member, IEEE), UI HUN SAGONG[1], (Graduate Student Member, IEEE), JONG HOON PARK[2], MURIM KIM[3], (Senior Member, IEEE), AND MYUN JOONG HWANG[1], (Member, IEEE)

[1]Department of Mechanical and Information Engineering, University of Seoul, Seoul 02504, South Korea
[2]Robot Navigation Project, Robot Business Division, LG Electronics, Seoul 07796, South Korea
[3]Human-Centered Robotics Research and Development Division, Korea Institute of Robotics and Technology Convergence, Pohang 37553, South Korea

Corresponding author: Myun Joong Hwang (mjhwang@uos.ac.kr)

This work was supported by the 2023 Research Fund of the University of Seoul.

ABSTRACT Motion planning for mobile manipulators is challenging because of their high degrees of freedom. The most effective approach for the motion planning of a mobile manipulator is to consider the different characteristics of a mobile robot and a manipulator while planning and controlling each system separately. In a previous study, different characteristics were considered using virtual impedance. This method involves forming a virtual impedance relationship between the two subsystems, enabling the mobile robot to track the movement of the manipulator. However, this study had certain limitations. Firstly, this method is not applicable to non-holonomic mobile robots. Secondly, obstacle avoidance methods for mobile robots are not considered. To address these limitations, our study proposes a novel concept for our motion planner that is called the virtual impedance energy field (VIEF), which refers to the work and change in total energy. We solved the first limitation of the previous study by transforming the virtual impedance force into the VIEF. Moreover, by integrating an obstacle field with the VIEF and using it as a local costmap, the second limitation was addressed. To validate the performance of our motion planner, we conducted simulations in environments comprising obstacles, with straight and curved trajectories of the end-effector. We then analyzed the pose change graph of the mobile robot and end-effector, pose error of the end-effector, and velocity of the mobile robot. We therefore confirmed that the manipulator successfully followed the desired trajectory, while the mobile robot maintained a distance from the end-effector and avoided obstacles.

INDEX TERMS Mobile manipulator, motion planner, virtual impedance, virtual impedance energy field.

I. INTRODUCTION

Mobile robots use their mobility to move objects in expansive environments or perform tasks such as surveillance and cleaning [1], and manipulators use their manipulation capability to grasp specific objects for pick-and-place or assembly tasks [2]. However, the mobile robot lacks manipulation capabilities, and the manipulator lacks the mobility to change its working space. Therefore, a mobile manipulator with both mobility and manipulation capability has significant advantages in the use of robots for various tasks within an infinite working space.

Fig. 1 presents a mobile manipulator that combines a mobile robot and a manipulator. Mobile manipulators have various applications, including tasks in factories [3], indoor and outdoor coffee delivery [4], assisting patients in hospitals [5], and space exploration [6]. However, combining a mobile robot and a manipulator increases the overall system's degrees of freedom. Consequently, the motion planning for mobile manipulators remains a challenging problem and an ongoing research area.

Motion planning methods for mobile manipulators in previous studies can be categorized into three perspectives of

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng Chen.

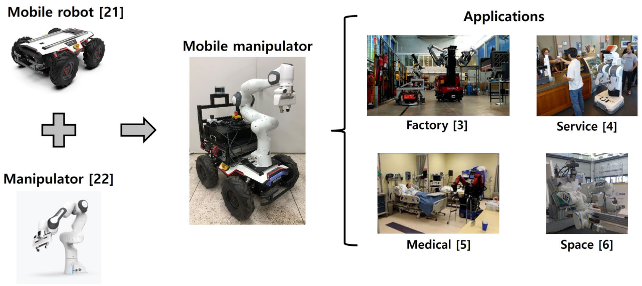

FIGURE 1. Examples and applications of mobile manipulator.

the system [7]. The first approach considers the mobile manipulator as separate systems [8]. However, this method has a limitation in fully utilizing a mobile manipulator. The second approach considers a mobile manipulator as a single high-degree-of-freedom system [9], [10]. However, the high computational cost of this method makes it challenging to perform other tasks such as online obstacle avoidance. In the third method, the mobile robot and manipulator are considered as separate systems, while considering the different characteristics of each system. Mobile robots and manipulators have different characteristics in terms of their purposes, control methods, kinematic features, and other aspects. Therefore, both the systems can be integrated to reflect these characteristics. In previous studies, hierarchical motion planners were used [11], [12] or the motion planner was designed by optimizing the characteristics of both subsystems to reflect their different characteristics [13]. By reflecting on the different characteristics of each system in the motion planning system, it is possible to address the limitations of the first separate system and simultaneously reduce the computational costs, which is a drawback of the second high-degree-of-freedom system.

This study introduces a virtual impedance relationship between the two subsystems to reflect their different characteristics. This relationship is formed between the end-effector and mobile robot, thus allowing the mobile robot to smoothly follow the end-effector while the end-effector tracks a path [14]. Virtual impedance has been widely used in human interaction research, particularly for handling the contact between robots and their surrounding environment. A prominent example is impedance control [15]; in a previous study, virtual impedance was applied to the control of a mobile manipulator [16]. This study proposes a control method that uses virtual impedance for a mobile manipulator performing collaborative human–robot tasks. The proposed control method involves establishing a preferred operating area for the end-effector and creating a virtual impedance wall at its boundaries. This wall is connected to a mobile robot through virtual impedance. Based on the characteristics of the virtual impedance, when the end-effector exceeds the boundaries, the mobile robot moves to maintain the impedance relationship. However, this study has two limitations: it is only focused on holonomic mobile robots, and it does not consider obstacle avoidance in the control process.

To overcome these limitations, this study proposes a novel concept called the virtual impedance energy field (VIEF), which can be used for the local planning of mobile robots. In a previous study [16], a force was directly applied using the virtual impedance relationship to holonomic mobile robots, thus allowing them to move according to the applied force. However, nonholonomic robots cannot move directly in the direction of the applied lateral force. In this study, we solved this challenge by indirectly applying the force from the virtual impedance to the mobile robot. We transformed the virtual impedance force into a VIEF, which represents the work and change in the total energy. We used the VIEF as a local costmap for the local planning of the mobile robot. Furthermore, we integrated an obstacle field with the VIEF and used it as a local costmap of the local planner. In this study, we selected the dynamic window approach (DWA) algorithm as the local planner [17], which effectively considers simultaneous obstacle avoidance and nonholonomic constraints. More detailed information regarding the DWA algorithm can be referred to in Section III-B.

In our motion planning method, we establish a virtual impedance relationship between the end-effector and mobile robot, thus allowing the mobile robot to track the movements of the end-effector. To make nonholonomic mobile robots applicable to our method, we transform the virtual impedance force into a VIEF. Furthermore, to address obstacle avoidance for mobile robots, we integrated an obstacle field with the VIEF and used it as a local costmap. The velocity of the mobile robot was calculated by selecting the DWA algorithm as the local planner of the mobile robot. This study makes the following contributions. Firstly, by transforming the virtual impedance relationship into the VIEF, nonholonomic mobile robots can be considered as applicable robots. Secondly, we provide a framework in which the VIEF is integrated with motion planners of the mobile robot and manipulator.

The remainder of this paper is structured as follows. Section II presents a literature review of previous studies, and Section III explains the theoretical background knowledge used in this paper, as well as the proposed motion planning method. In Section IV, the simulations conducted in a GAZEBO environment are described, and Section V presents the conclusions, limitations, and future work.

II. LITERATURE REVIEW

From a whole system perspective, a previous study [7] classified motion planning for a mobile manipulator into three approaches. In the first approach, the mobile manipulator is considered as a separate system. In this approach, only the mobile robot moves near the goal, and the manipulator reaches the goal sequentially. In [8], the researchers performed a pick-and-place task with automotive components by first identifying the closest feasible pose for grasping the component. They selected the A* algorithm to plan the mobile robot path and used the rapidly exploring random tree (RRT) algorithm as a sampling-based method for planning the motion of the manipulator. However, this method has the limitation of being unable to fully use the mobile manipulator because it cannot operate both systems simultaneously.

The second approach considers a mobile manipulator as a single high-degree-of-freedom system. It mainly uses a sampling method to find a solution that considers both the mobile robot and manipulator simultaneously in the configuration space of the whole system. In [9], the RRT algorithm was applied to achieve collision-free motion and manipulation in a 3D space. They used the RRT-GoalBias algorithm and postprocessed the initial path to smoothen it. In [10], motion planning was performed using a reinforcement learning approach based on the proximal policy optimization (PPO) algorithm for grasping objects. However, the high computational cost of this method makes it difficult to perform other tasks such as online obstacle avoidance.

The third method considers the different characteristics of the mobile robots and manipulators. In [11], a sampling-based approach called the Hierarchical and Adaptive mobile Manipulator Planner (HAMP) was proposed. This planner initially planned the motion of the mobile robot for collision-free motion using a probabilistic roadmap (PRM) algorithm. When a collision occurs with the manipulator, additional planning for the motion of the manipulator is performed using the PRM algorithm. This approach considers the different characteristics of both systems by planning the mobile robot hierarchically, prioritizing it, and then checking for collisions with the mobile manipulator. In [12], the motion planning for a mobile base was implemented using a hybrid sampling strategy, which is an optimized variant of the PRM. In this case, as in the previously described HAMP, the manipulator replanned its path only when a collision was detected. In [13], a real-time adaptive motion planner (RAMP) algorithm was proposed based on optimization methods. This algorithm is aimed at performing both path planning and plan execution simultaneously for real-time motion planning in dynamic environments. Each path was optimized using an evolutionary algorithm and the random selection method to ensure diversity. By assuming a loosely coupled relationship between the mobile robot and manipulator, the optimization process is primarily focused on minimizing the execution time, reducing energy consumption, and maximizing the manipulability. The goal of this approach is to select paths that meet the different characteristics of both the mobile robot and manipulator. By reflecting the different characteristics of both subsystems, these studies achieved the optimal utilization of a mobile manipulator while minimizing the computational costs. This paper proposes a virtual impedance relationship that connects the mobile robot and manipulator, thus allowing the virtual impedance force generated by the manipulator's motion to affect the mobile robot, and vice versa. The virtual impedance concept was used to integrate the characteristics of the two subsystems.

In addition to the previously mentioned system perspective, previous research on motion planning for mobile manipulators can be categorized according to the constraints of the motion planning methods. Among the various constraints, we focused on task constraints because they have a significant impact on the motion planning method. In previous studies, motion planning methods with task constraints were classified into sampling, optimization, and potential field-based methods. In [18], a sampling-based approach with kinematic redundancy was used to track the desired poses of an end-effector. Considering the link lengths of the manipulator, they iteratively generated circular regions centered around the end-effector. Within these regions, the pose of the mobile robot was determined using an RRT-based sampling method. For the manipulator, motion planning was achieved by generating a path using inverse kinematics for the target pose of the end-effector. In [19], an optimization-based approach was used that employed model predictive control (MPC) to plan the motion of a mobile manipulator when a path was provided. To enable the implementation of the MPC, a Jacobian matrix and kinematic model of the robot were derived. Validation tests were conducted to demonstrate the effectiveness of the proposed motion planning method and dynamic obstacle avoidance. In [20], the optimization-based approach was introduced, which uses a given trajectory to determine the maximum allowable load of a mobile manipulator. The augmented Jacobian method is employed to address task space constraints. In the previously mentioned study [14], the potential-field method was used. As in this study, the mobile robot in the previous study was exerted by an attractive force to track the end-effector and a repulsive force caused by nearby obstacles. The mobile robot created a vector field to maintain a specific distance from the manipulator. However, a limitation of the previous study is that the potential field was only applied to holonomic mobile robots. In the study [21], a potential field-based method was introduced that adjusts a given path in response to environmental changes or the appearance of obstacles. This approach uses the potential field method to attract robots to the original path of the end-effector while moving away from obstacles using repulsive forces, thereby facilitating path adaptation. Owing to their simplicity and lower computational complexity compared with other methods, potential field-based methods have been widely used in cases comprising task constraints. This study proposes a VIEF that shares similarities with potential fields. Like how a mobile robot moves toward a goal with lower potential in potential field methods, a mobile manipulator moves toward a target point guided by a virtual impedance field. However, because the virtual impedance force includes a nonconservative damping force, it cannot be considered as a conservative force and can be explained from a potential energy perspective. Therefore, we present this from a total-energy perspective. This study considered task constraints, thus allowing the mobile robot to follow the end-effector. Furthermore, it adapts the virtual impedance field to account for nonholonomic constraints.

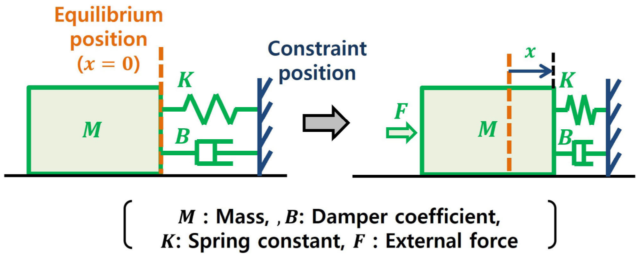
Equilibrium position $(x = 0)$
Constraint position
K
M
B
x
K
F
M
B
M : Mass, , B: Damper coefficient,
K: Spring constant, F : External force

FIGURE 2. Example of mass–damper–spring system.

III. MOTION PLANNING USING VIRTUAL IMPEDANCE ENERGY FIELD

A. VIRTUAL IMPEDANCE

Impedance is a concept applicable to both electrical and mechanical systems. In this study, we used mechanical impedance, which includes a mass, damper, and spring. Mechanical impedance Z is defined as the ratio of the force F to velocity $\dot{x}$ of an object, as shown in (1). This is represented as a mass–damper–spring system, as illustrated on the left side of Fig. 2. Therefore, (1) represents the relationship between the system with mass M, damping constant B, and spring constant K, located between the constraint position and equilibrium position, and the external force F, as shown on the right side of Fig. 2.

$$Z(s) = \frac{F(s)}{\dot{x}(s)} = Ms + B + \frac{K}{s} \tag{1}$$

$$M_d \Delta\ddot{p} + B_d \Delta\dot{p} + K_d \Delta p = F \quad (\text{where, } \Delta p = p_d - p) \tag{2}$$

One representative research comprising the application of mechanical impedance to robots is the impedance control of a manipulator [15]. As shown in Fig. 3, this method involves setting a virtual impedance between the end-effector and constraint position, which indirectly controls the force. When the user moves the constraint position or equilibrium position p_d of the manipulator's end-effector to the desired position, an error Δp is generated between the current pose p and p_d. The end-effector can be controlled using the virtual impedance from (2), which is an error equation. Therefore, the user can adjust the desired mass M_d, spring constant K_d, and damping constant B_d set for each axis to set up the desired motion characteristics of the end-effector. In this study, we used virtual impedance as a method for a mobile robot to follow the trajectory of an end-effector effectively. This approach has two advantages. Firstly, it enables the mobile robot to track the end-effector smoothly. Secondly, it facilitates the analysis and calculation of dynamic behaviors such as impedance control.

B. DYNAMIC WINDOW APPROACH

The DWA [17] is an algorithm widely used in local path planning for mobile robots. It performs path planning and velocity control to reach a destination while avoiding obstacles. This algorithm comprises four steps, as depicted in Fig. 4, and repeats them. In the first step, the current linear velocity v and angular velocity ω of the mobile robot are acquired. In the second step, a velocity boundary called the dynamic window is formed in the velocity space centered around the current linear and angular velocities. A certain number of velocity samples are extracted from the velocities within the boundary. The dynamic window range is set by considering the maximum and minimum velocities and accelerations of the robot, thus ensuring that only feasible velocity samples for the robot are selected. In the third step, a forward simulation is used to calculate the trajectory generated from the selected velocity samples when the robot moves at a constant velocity for a certain period. In the fourth step, the cost is calculated for each trajectory by summing the weights of several cost functions, and the sample with the lowest cost, that is, the optimal cost, is output as the next commanded velocity. The cost functions consider factors such as distance to the surrounding obstacles, distance to the destination, and distance to the global path. Once the next velocity is output in this manner, the entire process is repeated iteratively by returning to the first step according to the algorithm cycle.

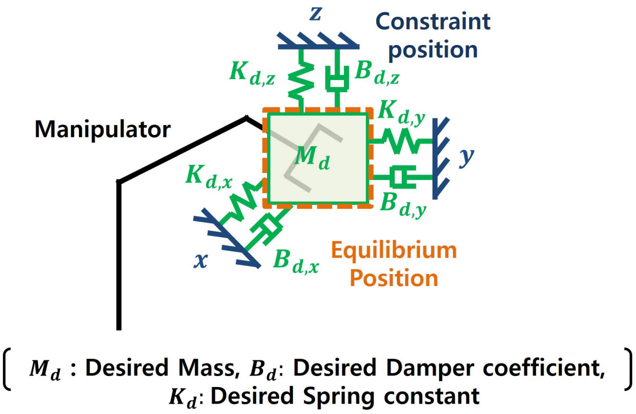

FIGURE 3. Impedance control of manipulator.

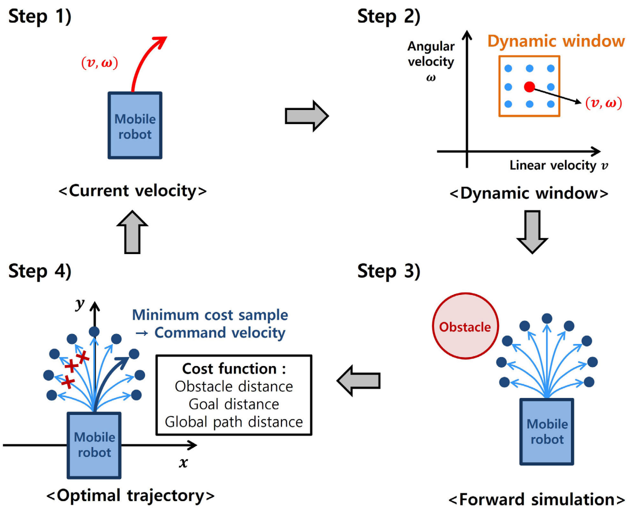

FIGURE 4. DWA algorithm.

In this study, the DWA algorithm used for the motion planning of mobile robots has three major advantages. First, it considers the actual dynamic behavior of mobile robots. By setting the dynamic window in the velocity space, only the achievable velocity samples based on the current velocity state are automatically considered. Therefore, it can be applied not only to holonomic robots but also to the differential-drive non-holonomic mobile robots used in this study. The second advantage is the existence of an optimization process for multiple velocity samples using a cost function. When planning the trajectory of a mobile robot, various objectives can be easily reflected by transforming them into cost functions. The third advantage is that, because the output velocity of the mobile robot is calculated during the optimization process, trajectory planning and velocity control of the mobile robot are performed simultaneously. In addition, obstacle avoidance can be performed online, even in environments comprising obstacles.

C. MOTION PLANNING BASED ON VIRTUAL IMPEDANCE ENERGY FIELD (VIEF)

This study proposes a motion planning method that takes into consideration the differences between the characteristics of the mobile robot and manipulator when the trajectory of the end-effector is given for a specific task. The goal of controlling the manipulator is to track the given trajectory, while the mobile robot is designed to simultaneously follow the end-effector using a virtual impedance relationship and avoid obstacles in the surroundings. To achieve this, a virtual impedance relationship is formed from the end-effector to the center of the mobile robot, thus allowing the mobile robot to follow the motion of the end-effector.

$$\tau_{total} = J^T\left(K_P\Delta x + K_D\Delta\dot{x}\right) + \tau_{gravity}$$
$$(\text{where, } \Delta\dot{x} = \dot{x}_d - \dot{x},\ \Delta x = x_d - x) \tag{3}$$

The control methods for both systems can be explained as follows. First, for the manipulator, a motion planning and control approach is used with a Cartesian proportional derivative (PD) controller. High P gain allows the end-effector to converge to the desired target poses in the Cartesian space. Eq. (3) is used to determine the pose error Δx, which represents the difference between the end-effector current pose x and the desired target pose x_d. Furthermore, we calculate the velocity error $\Delta\dot{x}$, which represents the difference between the current velocity $\dot{x}$ and the desired target velocity $\dot{x}_d$. Subsequently, the errors Δx and $\Delta\dot{x}$ are multiplied by their respective PD gains K_P, K_D. This multiplication transforms the errors into static forces in the Cartesian space. These forces are then multiplied by the Jacobian transpose matrix J^T to convert them into joint space torques. To compensate for gravity, the pre-calculated $\tau_{gravity}$ is considered, thus resulting in the final manipulated torque τ_{total}.

For the mobile robot, the concepts of virtual impedance and the DWA algorithm are used for motion planning to achieve both tracking of the end-effector and avoidance of the surrounding obstacles. Initially, as illustrated in Fig. 5, a straight line is drawn in the $x_G - y_G$ plane connecting the center of the mobile robot and the position of the end-effector. A virtual impedance relationship is formed on the $x_G - y_G$ plane, and the equilibrium position p_d is defined as the specific equilibrium distance behind the end-effector towards the base of the manipulator. Therefore, this relationship forces the mobile robot to converge to an equilibrium position by exerting a force f_{cmd}. The process of computing the output force f_{cmd} derived from the virtual impedance is as follows. First, as shown in (4), the simple 2D dynamic equation of the mobile robot is represented with the robot mass M_B, acceleration $\ddot{p}_B$, and output force f_{cmd}. In this case, the $\ddot{p}_B$ in (6) is derived from the impedance error equation in (5). Therefore, by substituting the derived acceleration $\ddot{p}_B$ into (4), the final output force f_{cmd} of the mobile robot can be expressed as (7). It is assumed that mass M_d used in the impedance error equation is replaced by the mobile robot mass M_B. Furthermore, M_d, B_d, and K_d are assumed to be diagonal and positive-definite matrices. Additionally, M_B and M_d are assumed to be identical, thereby ensuring that $M_B M_d^{-1} = I$ is satisfied.

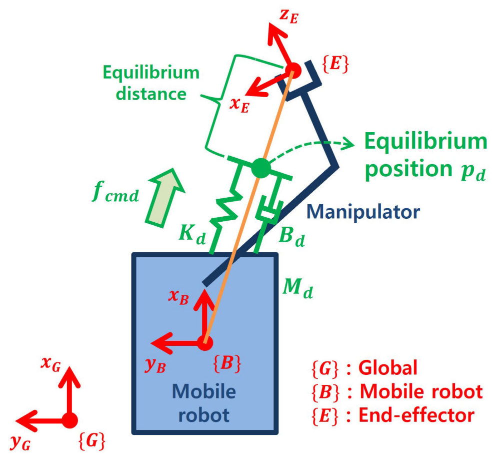

FIGURE 5. Virtual impedance relationship of mobile manipulator.

$$M_B\ddot{p}_B = f_{cmd} \quad \left(\text{where, } p_B = \begin{bmatrix} x_B \\ y_B \end{bmatrix}\right) \tag{4}$$

$$M_d\Delta\ddot{p} + B_d\Delta\dot{p} + K_d\Delta p = F \tag{5}$$

$$\left(\text{where, } M_d = \begin{bmatrix} M_{d,x} \\ M_{d,y} \end{bmatrix}, B_d = \begin{bmatrix} B_{d,x} \\ B_{d,y} \end{bmatrix}, K_d = \begin{bmatrix} K_{d,x} \\ K_{d,y} \end{bmatrix},\right.$$
$$\left.\Delta p = p_d - p_B, p_d = \begin{bmatrix} x_d \\ y_d \end{bmatrix}\right)$$

$$\ddot{p}_B = \ddot{p}_d + M_d^{-1}\{B_d\left(\dot{p}_d - \dot{p}_B\right) + K_d(p_d - p_B) - F\} \tag{6}$$

$$f_{cmd} = M_B\ddot{p}_d + B_d\left(\dot{p}_d - \dot{p}_B\right) + K_d\left(p_d - p_B\right) - F \tag{7}$$

However, the direct application of the force obtained from the virtual impedance to the mobile robot has two limitations, as in a previous study [16], where the virtual impedance was applied to the control of the mobile manipulator. The first limitation is the lack of collision-avoidance strategies

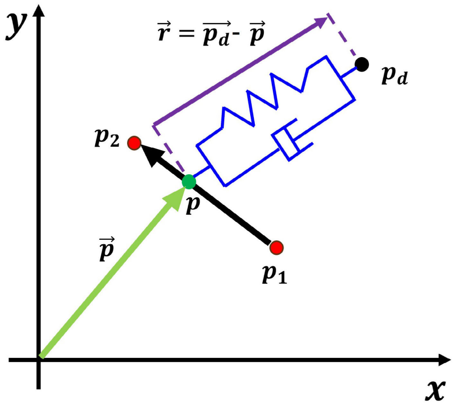

FIGURE 6. Impedance relationship between p and p_d.

for mobile robots. Second, for holonomic mobile robots, it is possible to output the velocity in the same direction as the force f_{cmd} in all directions. However, in the case of non-holonomic mobile robots, where motion in the y_B direction is mostly restricted, it is infeasible to move directly in the force direction. One way to avoid directly applying force to a robot involves using a 'field.' A field represents the effects of forces on an object by assigning cost to the surrounding space. This method expresses the influence of forces on a mobile robot. For instance, as introduced in Section II, [14] suggests that the robot navigates using a potential field that reflects these forces instead of applying attractive and repulsive forces directly. This study proposes a new method named VIEF, which transforms the concept of f_{cmd}, derived from the virtual impedance, into a field.

The following provides two perspectives on transforming force into the VIEF field. Firstly, from the perspective of work, we have constructed the field based on the work done by the impedance force. If we consider the mobile robot moving to hypothetical future positions over short periods, the distance and speed of the movement will vary depending on the future positions. By simplifying this process, we can calculate the work done by the impedance force at future positions, and the most stable position that conforms to the impedance force is where the work is minimized. Secondly, from the energy perspective, our virtual impedance relationship between the mobile robot and the end effector is based on the mass-spring-damper system. In this system, the robot's destination is an equilibrium position where the system stabilizes, and the total energy is zero due to the energy dissipation by the damper. Consequently, our goal is to iteratively move the robot to a position where the total energy is minimized, ultimately aiming to reach an equilibrium position where the total energy is zero. In summary, we calculate the minimum of ΔE, which represents the change in total energy and is equivalent to the work, W. We then configure our field with the W value at each position. By identifying and moving toward the position where W is minimized, the robot can quickly reach the equilibrium position.

Applying the above to our system and explaining it through formulas is as follows. In Fig. 6, the current position of the mobile robot is denoted by p_1. The position variable p is set on a straight line towards the next position p_2. The vector from p to p_d is represented as $\vec{r}$. Therefore, as in (8), the force f_{cmd} can be calculated by integrating with respect to $\vec{r}$. We assume that over a short period, the acceleration $\ddot{p}_d$ at the equilibrium position p_d is zero, indicating that the velocity $\dot{p}_d$ remains constant. Similarly, we assume that the velocity $\dot{p}_B$ from p_1 to p_2 remains constant over a short period. As a result, the work W is expressed by (9), representing the value at position p_2 in the VIEF. Since W is formulated as a quadratic function of position p_2, the minimum value of W at $p_{2,min}$ can be computed by differentiating W with respect to p_2 as shown in (10). In (11), it is confirmed that if there are no external forces and dampers in the system, $p_{2,min}$ becomes p_d. However, if a damper is added, $p_{2,min}$ represents a different position from p_d depending on the ratio of the damping constant B_d to the spring constant K_d. The larger the damping constant B_d is, the more $p_{2,min}$ is pulled from p_d towards the opposite direction of the difference between the target velocity $\dot{p}_d$ and current velocity $\dot{p}_B$. Similarly, when external forces exist, $p_{2,min}$ is pulled in the opposite direction of the external force to maintain the equilibrium distance. In summary, the mobile robot aims to move towards the new equilibrium position $p_{2,min}$, considering the target position, relative velocity, and external forces.

$$W = \Delta E = \int_{\vec{r}_1}^{\vec{r}_2} f_{cmd} d\vec{r} \quad (\text{where, } \vec{r} = p_d - p) \tag{8}$$

$$\begin{aligned} W = &\int_{p_1}^{p_2} \left\{ -M_B \ddot{p}_d - B_d \left(\dot{p}_d - \dot{p}_B\right) - K_d \left(p_d - p\right) + F \right\} dp \\ &= -M_B \ddot{p}_d \left(p_2 - p_1\right) - B_d \left(\dot{p}_d - \dot{p}_B\right)\left(p_2 - p_1\right) \\ - K_d &\left\{ p_d \left(p_2 - p_1\right) - \frac{1}{2}\left(|p_2|^2 - |p_1|^2\right)\right\} + F\left(p_2 - p_1\right) \end{aligned} \tag{9}$$

$$(Assumption, \ddot{p}_d = 0, \dot{p}_d \ \& \ \dot{p}_B \ is \ Constant)$$

$$\frac{dW}{dp_2} = -B_d\left(\dot{p}_d - \dot{p}_B\right) - K_d\left(p_d - p_2\right) + K \tag{10}$$

$$p_{2,min} = p_d - \frac{B_d}{K_d}\left(\dot{p}_d - \dot{p}_B\right) - \frac{F}{K_d} \tag{11}$$

As shown in Fig. 7, to overcome the limitations of obstacle avoidance and non-holonomicity in previous studies, we used the VIEF as a costmap for the DWA algorithm, which is the local planner of the mobile robot. The cost function used for the costmap of the DWA algorithm includes the VIEF cost function and the obstacle cost function. Using the DWA algorithm, the robot generates velocity samples that can be output. For each sample, cost functions are calculated based on the VIEF and obstacle cost functions. Because nonholonomic mobile robots cannot move in the y_B direction, velocity samples are created by combining the velocities in the x_B direction and the z_B rotation directions. If a collision occurs with an obstacle while moving at the velocity of the sample, the velocity sample is discarded. The velocity command of

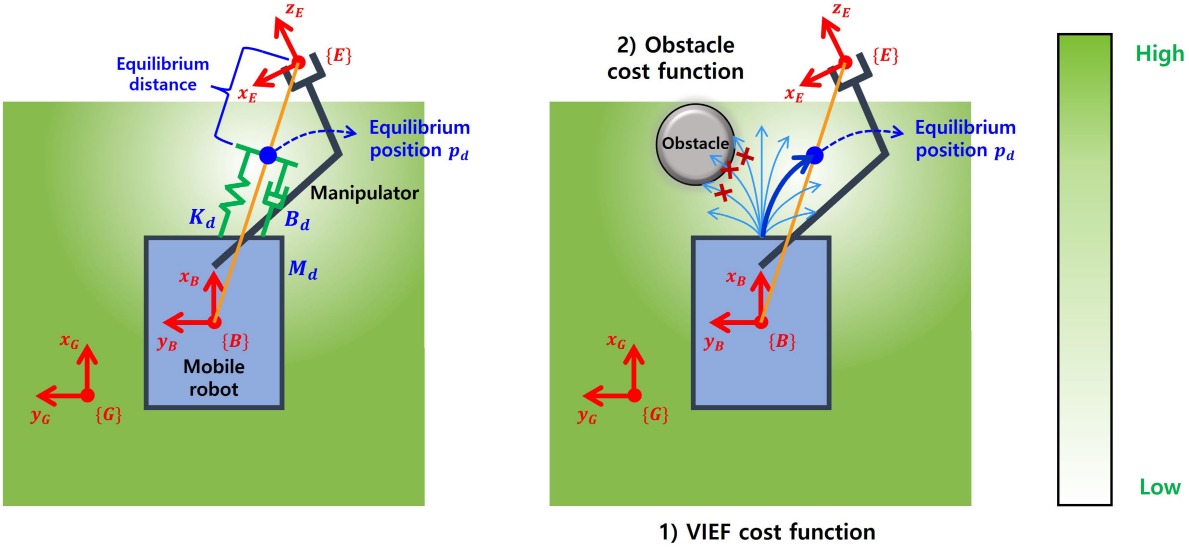

FIGURE 7. Using VIEF as cost function in DWA algorithm.

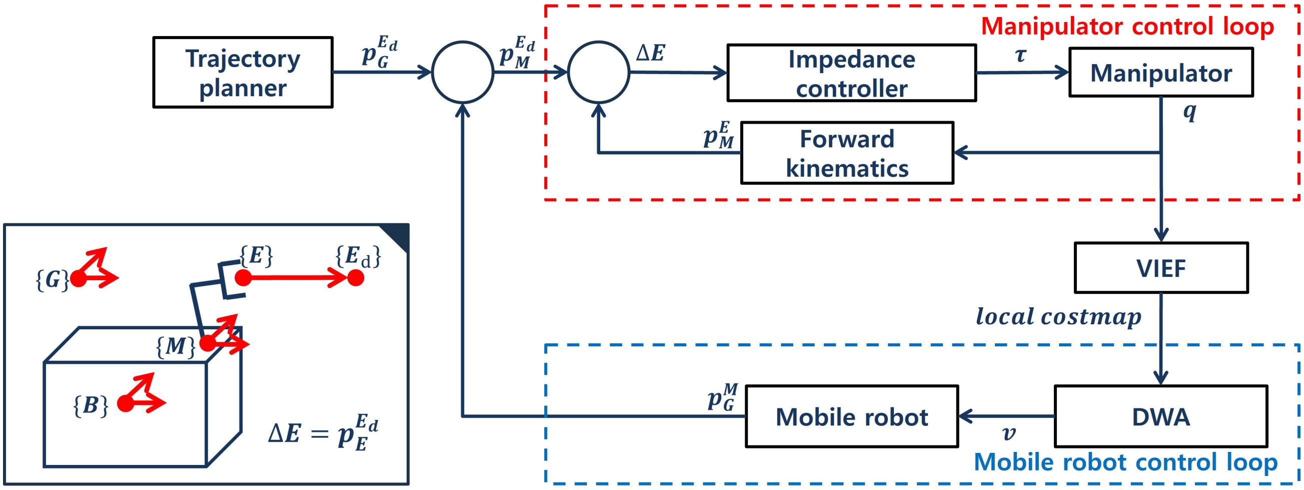

FIGURE 8. Motion planner structure using VIEF.

the mobile robot is determined by selecting the velocity sample with the lowest cost from the remaining feasible samples.

Fig. 8 presents the structure of the proposed motion planner. The left part of Fig. 8 depicts the coordinates of the global frame $\{G\}$, mobile base $\{B\}$, manipulator base $\{M\}$, and end-effector $\{E\}$, and the target coordinates of the end-effector $\{E_d\}$. When the target coordinate $p_G^{E_d}$ of the end-effector in the global coordinate system is provided by the trajectory planner, the target position $p_M^{E_d}$ in the current manipulator base frame $\{M\}$ is determined. p_M^E is also the current end-effector position in frame $\{M\}$. The Cartesian space error ΔE is calculated through the difference between $p_M^{E_d}$ and p_M^E. The PD controller of the manipulator responds to the error ΔE by calculating the torque τ, which prompts the manipulator to move as a result. Subsequently, a feedback control loop is established by employing forward kinematics to determine the position of the end-effector p_M^E based on the base frame of the manipulator. In this case, the virtual impedance relationship between the mobile robot and the moved end-effector results in the formation of a VIEF. This field provides a local costmap for a mobile robot, connecting two systems with different characteristics. The movement of the mobile robot is determined by the DWA planner, which determines the velocity v based on the local costmap. As the mobile robot moved, the $\{M\}$ coordinate moves along with it. Consequently, the position controller of the manipulator is adjusted again in response to the error ΔE of the end-effector.

In summary, the proposed motion planning method based on the VIEF uses virtual impedance relationships. The manipulator, which is controlled by a Cartesian PD controller, allows the end-effector to continuously track along the desired trajectory. Owing to the changed position of the

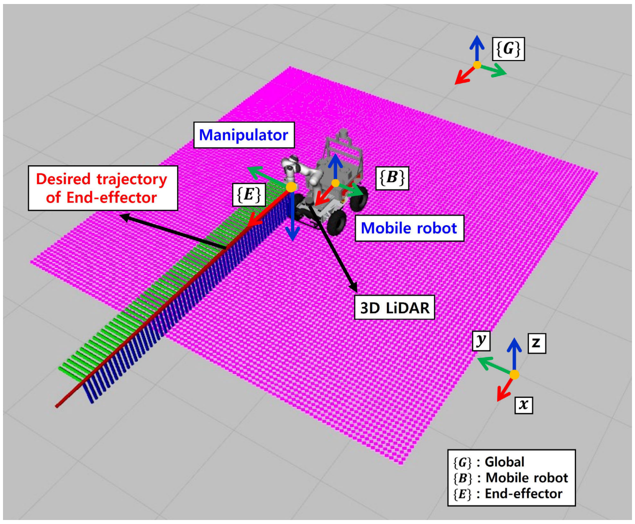

FIGURE 9. Simulation environment.

end-effector, the mobile robot receives virtual impedance forces, thereby forming a VIEF. The mobile robot employs the VIEF as a local costmap, thus allowing the DWA algorithm to plan the motion of the mobile robot. This allows the robot to avoid obstacles in its surroundings while simultaneously tracking the end-effector. In conclusion, this motion planning approach, when given the target trajectory for the end-effector, can generate movements that consider the different characteristics of both the mobile robot and manipulator subsystem.

IV. SIMULATION

A. SIMULATION ENVIRONMENT AND FRAMEWORK

To validate the motion planner proposed in this study, we introduced a mobile manipulator system and 3D LiDAR sensor into the simulation environment, as shown in Fig. 9. We generated the desired trajectory of the end-effector and conducted simulations. The simulation carried out on an Intel i9-13900F 24-core 5.6GHz CPU, and the simulation was conducted in Ubuntu 20.04 environment with a robot operating system (ROS) noetic and the Gazebo simulator. For the hardware, we used Scout 2.0 [22], which is a nonholonomic differential drive robot, for the mobile robot and Panda [23] with seven degrees of freedom for the manipulator. In addition, the VLP-16 [24], a 3D LiDAR, was used to detect obstacles surrounding the mobile robot. For the software, we used the open-source ROS framework and customized the relevant components for our motion planner. To control the manipulator, we used the franka_ros package [25] and performed Cartesian PD control. For the mobile robot, we modified the key components of the navigation [26] and move_base_flex packages [27], including the DWA.

The simulation system structure diagram is presented in Fig. 10, where each square box represents a node. First, the *3D LiDAR* node acquires *laser scan* data from the sensor, which are then used to recognize the surrounding obstacles in the *mobile manipulator planner* node. At the *end-effector planner* node, when the destination of the end-effector is set, the manipulator performs Cartesian trajectory planning to reach the destination from its current position. In this simulation, the linear segments with parabolic blends (LSPB) method were used for Cartesian trajectory planning. This allowed for smooth movements at the start and end of the trajectory. When a trajectory consisting of the desired position, velocity, and acceleration is generated, as shown in Fig. 11, it is passed to the *global planner* node. Subsequently, each desired pose of the end-effector corresponding to the current time in the trajectory is transmitted to the *local planner* node. The local planner node continuously receives the desired pose of the end-effector and sends the calculated *joint torque* and *command velocity* to the *manipulator* and *mobile robot* nodes, respectively, during every control cycle. The mobile manipulator system was controlled through this process.

Fig. 12 presents an image of the motion planner during the simulation. Motion planning and manipulator control are implemented to track the desired trajectory of the end-effector. Simultaneously, within the mobile robot, the DWA algorithm generates velocity samples, taking into account the dynamic behavior of the robot. Each trajectory derived from these velocity samples is evaluated using cost functions. The obstacle cost function, C_{obs}, is defined as in (12), assigning a value of 1 if there is an obstacle and 0 otherwise. The impedance cost function, C_{imp}, utilizes a normalized cost that ranges between 0 and 1, as described in (9). The total of the cost functions, C_{total} as indicated in (13), is expressed as the sum of various cost functions, each of which can be represented as a layer on the cost map. Additional cost functions, such as an oscillation function or a goal function, can be incorporated as needed. Consequently, the velocity samples with the minimum values among the trajectories are selected as the command velocities for the mobile robot.

$$C_{obs}(x,y) = \begin{cases} 1 & \textit{if there is an obstacle} \\ 0 & \textit{otherwise} \end{cases} \tag{12}$$

$$C_{total}(x,y) = C_{imp}(x,y) + C_{obs}(x,y) \tag{13}$$

The parameter sets for the simulation are listed in Table 1. The parameters in the first group are associated with the VIEF. The *grid resolution* indicates the length of each grid in the field and is set 0.05 m per grid. The parameters $M_{d,x}$ and $M_{d,y}$ of the impedance relationship in (9) are both set M_B. K_d and B_d are heuristically set as 5000 and 350, respectively. The *equilibrium distance* that the robot system aims to converge to at the end is set as 0.55 m, which is the distance between the end-effector and the center of the mobile robot. The parameters in the second group are associated with DWA. The *size of the local costmap* is 5 × 5 m, and *the number of velocity sample* is set as 20 in the x- and z-directions. The forward simulation lasts 2 s, and the *control frequency* of the algorithm is 20 Hz. These are the parameters in the last group for the data acquisition frequency. The data acquisition frequency for the point cloud of the 3D LiDAR,

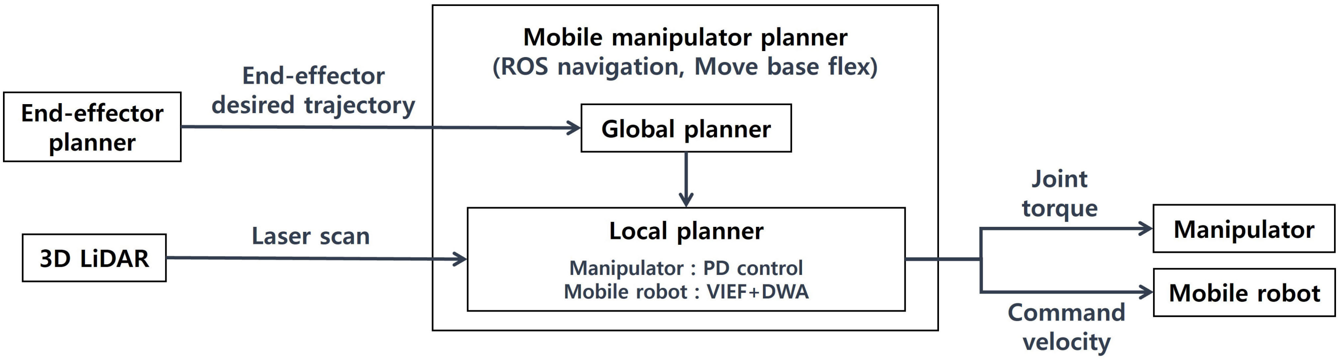

FIGURE 10. Simulation system structure.

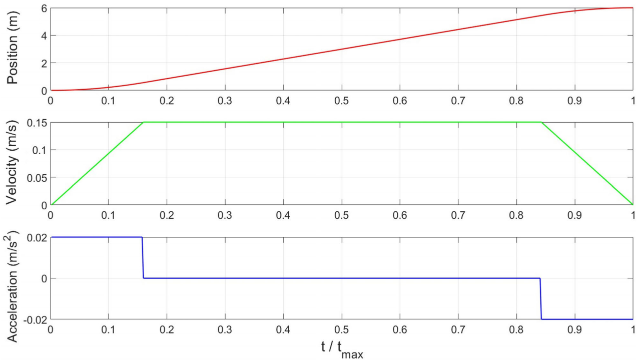

FIGURE 11. Planned result obtained using the LSPB.

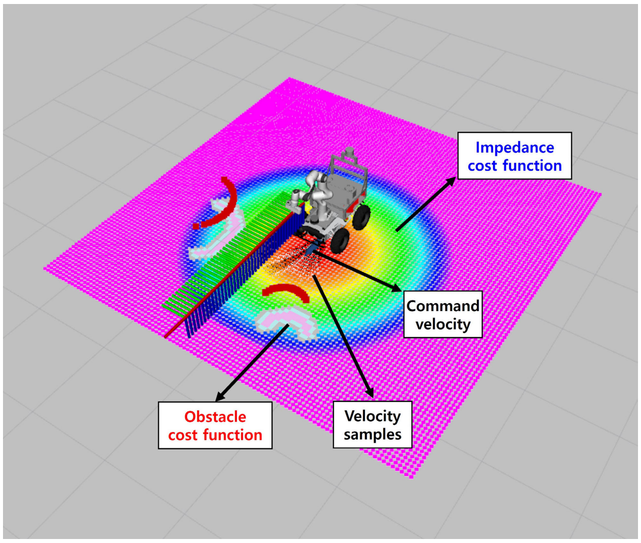

FIGURE 12. Running proposed motion planner in simulation.

pose of the manipulator and mobile, and the running time are respectively set as 10 Hz, 10 Hz, and 5 Hz.

B. SIMULATION CONTENTS

TABLE 1. Simulation parameter setting.

Group	Parameter	Value
Virtual impedance energy field	Grid resolution	0.05m
	$M_{d,x}, M_{d,y}$	M_B, M_B
	$B_{d,x}, B_{d,y}$	350, 350
	$K_{d,x}, K_{d,y}$	5000, 5000
	Equilibrium distance	0.55m
DWA	Size of local costmap	5m X 5m
	The number of velocity samples v_x, v_y, v_z	20, 0, 20 EA
	Forward simulation time	2s
	Control frequency	20Hz
Data acquisition frequency	Point cloud of 3D LiDAR	10Hz
	Pose of manipulator and mobile	10Hz
	Running time	5Hz

In this study, three simulations (SIMULATION1 SIMULATION2 and SIMULATION3) were conducted. SIMULATION1 validates the proposed motion planning method for the straight (SIMULATION1-1) and curved (SIMULATION1-2) trajectories of the end-effector in obstacle-free environments. SIMULATION2 validates the motion planning method for straight (SIMULATION2-1) and curved (SIMULATION2-2) end-effector paths in environments comprising cylindrical obstacles. SIMULATION3 validates the motion planning method for a straight end-effector path within a complex environment with a narrow gate, passage, and various obstacles. In addition, the algorithm execution times for the preceding simulations were measured. We measured the execution time of the algorithm in two parts. The first is the time required to compute the cost of the virtual impedance field and update the local cost function. The second part is the time required to compute the total cost for all the velocity samples based on the updated local cost function and derive the optimal velocity sample with the minimum cost.

First, in SIMULATION1-1, a straight trajectory was generated for the end-effector with fixed poses up to a position 6 m ahead of the mobile robot, as shown in Fig. 13. Next, in SIMULATION1-2, a curved trajectory with a radius of 3 m, wherein the poses were changed for 90° along a circular trajectory, was generated for the end-effector, as shown in

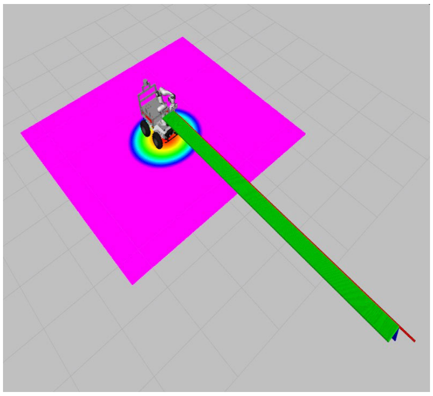

FIGURE 13. Straight path (SIMULATION 1-1).

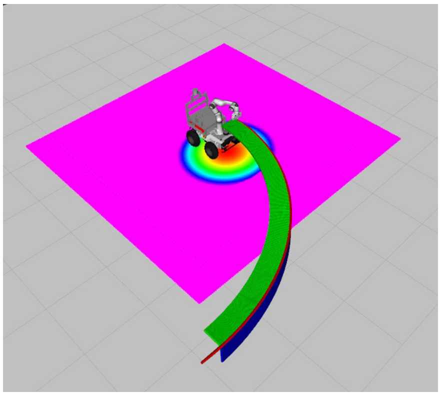

FIGURE 14. Curved path (SIMULATION 1-2).

Fig. 14. Three data were obtained from the simulations. First, a pose change graph was created by accumulating the overall pose variations of the mobile robot and end-effector. Second, we calculated the position and angle errors by comparing the output poses of the end-effector with the desired poses. The third set of data represented the linear and angular velocity graphs of the mobile robot, thus outputting the commanded velocity and resulting in the output velocity. Moreover, for SIMULATION1-1 exclusively, we generated a velocity graph of the end-effector in the x-direction of {G} and compared it with the planned velocity.

In SIMULATION2, the straight and curved trajectories were identical to those in SIMULATION1. In the case of the straight trajectory, the obstacles with heights of 1 m were placed at (1.75, 0.6), (3.5, -0.6), and (5.25, 0.7), and the radius of each obstacle was set as 0.25 m, 0.25 m, and 0.5 m in the XY plane relative to the mobile robot, as shown in Fig. 15. For the curved trajectory, the obstacles were located at (1.8, 0.2) and (2.8, -1.9), both at a height of 1 m and radius of 0.25 m in Fig. 16. During this process, we verified whether the end-effector followed the desired trajectory and whether the mobile robot avoided obstacles. The contents of simulation results were identical to those of SIMULATION1. Firstly, we recorded a pose graph in which the overall poses of the mobile robot and end-effector were accumulated. Secondly, we analyzed the pose error of the end-effector by comparing its output poses with the desired poses. Finally, we verified the velocity graph for the mobile robot.

In SIMULATION3, a straight trajectory for the end-effector was generated on a more complex map. Various obstacles were present, as depicted in Fig. 17. Initially, a narrow gate, 1 m wide, was positioned at the map's entrance. Following this, a short passage, 1.5 m in length, was diagonally located at (2.0, −0.15). Subsequently, sphere and cylindrical obstacle with heights of 1.5 m were placed at (5.6, 0.6) and (6.9, −0.55), with radii of 0.5 m and 0.25 m, respectively. Finally, a straight passage, 1 m wide and 2 m long, allowed the robot to exit the map.

C. CSIMULATION RESULT

1) SIMULATION 1

In SIMULATION1, we validated our motion planning method for the straight and curved trajectories of the end-effector in an obstacle-free environment. The simulation yielded three types of data: the pose change graph of the mobile robot and end-effector, pose error of the end-effector, and velocity of the mobile robot. Furthermore, we analyzed the velocity graph of the end-effector in the x-direction of {G} only for SIMULATION1-1.

a: SIMULATION 1-1

Figs. 18–21 present the results obtained when the trajectory of the end-effector was straight. In Fig. 18, the pink and light-blue dots represent the positions of the end-effector and mobile robot, respectively, and the yellow dashed lines connect the corresponding pink and light-blue dots. Consequently, as the end-effector followed a straight trajectory, the mobile robot also moved along a straight trajectory while maintaining a certain distance from the end-effector. In Fig. 19, the pose error of the end-effector is represented by dividing it into position and angle errors. The angle error illustrates the errors in the angles for the x, y, and z axes of the end-effector coordinates {E}, as indicated by the red, green, and blue lines, respectively. The average and maximum position errors were 0.00519 m and 0.0184m, respectively, while the absolute values of the angle error peaked at 0.51°, 1.36°, and 4.01° along the x-, y-, and z-axes, respectively. The pose error remained small until the end-effector reached the final target position (approximately 46 s), after which it rapidly increased to its maximum value. This was attributed to the residual velocity of the end-effector, which caused it to overshoot the final target position when attempting to stop. Fig. 21 presents a graph representing the velocity of the end-effector along the x-axis of {G}. The desired velocity, referred to as the "command velocity," and the actual output velocity, referred to as the "output velocity," are indicated in red and blue, respectively. The results confirmed that the end-effector moved in accordance with the planned velocity profile obtained using the LSPB method. Fig. 20 presents the linear and angular velocities of the mobile robot as a result of the DWA. As a result, by following a trajectory similar to

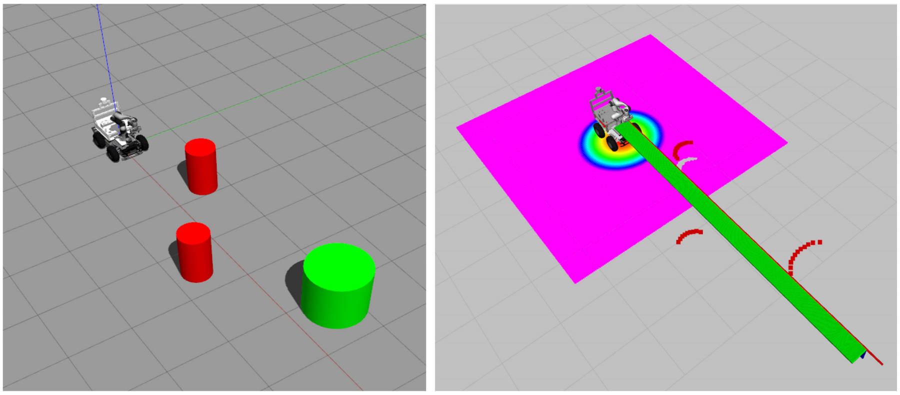

FIGURE 15. Straight path with obstacles (SIMULATION2-1).

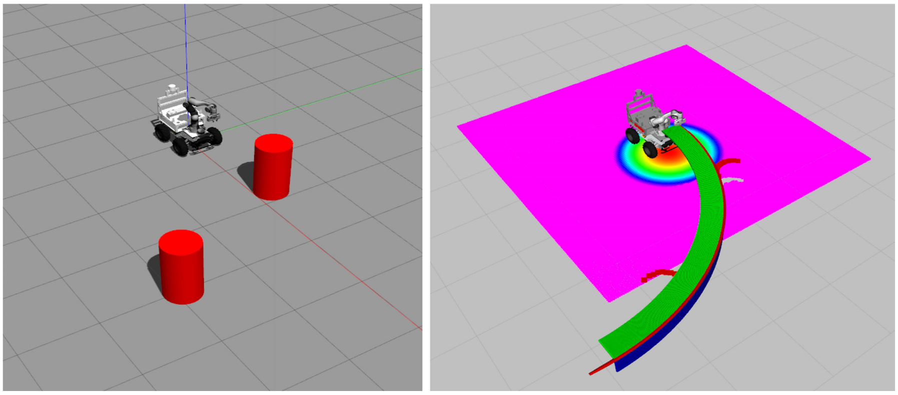

FIGURE 16. Curved path with obstacles (SIMULATION2-2).

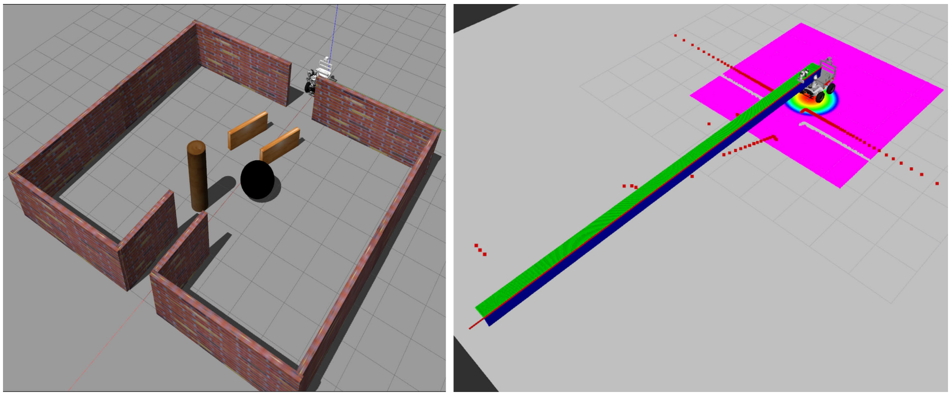

FIGURE 17. Straight path with complex obstacles (SIMULATION3).

that of the end-effector, the mobile robot increased its linear velocity from 0, maintained a constant velocity for some time, and then gradually decreased before stopping at the destination. The output velocity graph presented a trapezoidal shape similar to the velocity profile planned by the LSPB. Therefore, it was confirmed that, using our motion planner, the mobile robot moved in a profile similar to the planned velocity of the end-effector. Moreover, in comparison with the constant-velocity segment from 7.9–41.5 s in Fig. 21, the corresponding segment in Fig. 20 also lasted for almost the same duration. This indicates that the trapezoidal velocity profile of the mobile robot was output after the end-effector, as the mobile robot generated a velocity in response to the movement of the end-effector. The angular velocity was close to zero, except for the period following the mobile robot's stop.

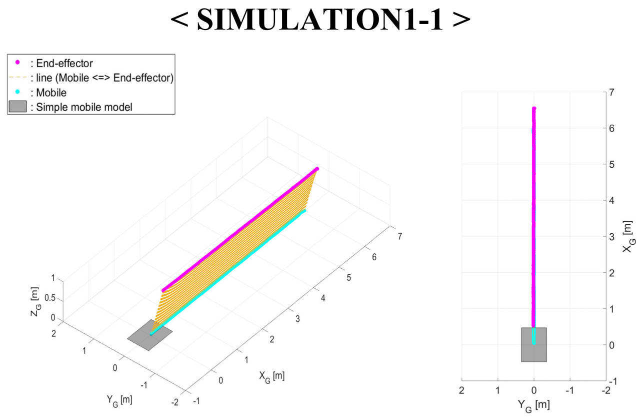

FIGURE 18. Pose graph (SIMULATION1-1).

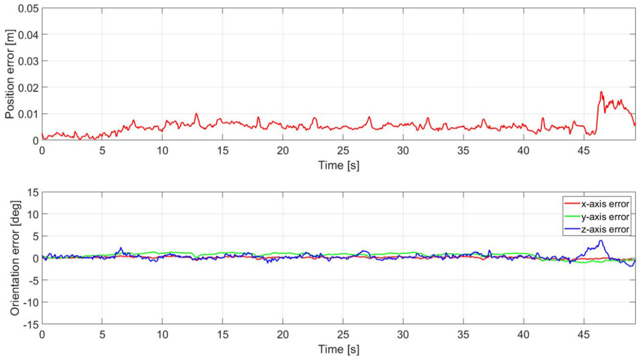

FIGURE 19. Pose error of end-effector (SIMULATION1-1).

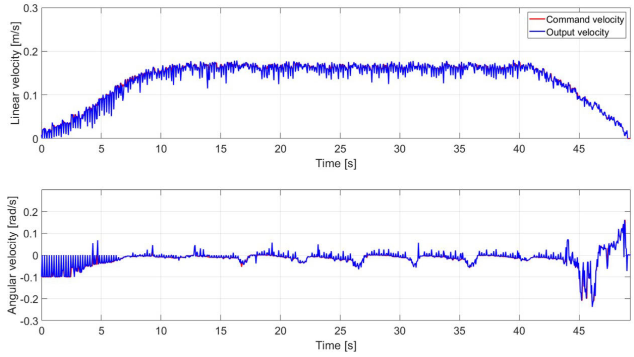

FIGURE 20. Command and output velocity of mobile robot (SIMULATION1-1).

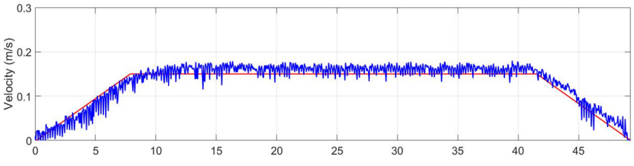

FIGURE 21. Command and output velocity(x) of end-effector (SIMULATION1-1).

b: SIMULATION 1-2

Figs. 22–24 present the results obtained when the trajectory of the end-effector is curved. As shown in Fig. 22, the mobile robot moved along a curved trajectory while maintaining an appropriate distance from the end-effector. In Fig. 23, the average and maximum position errors were 0.00658 m and 0.0235 m, while the absolute values of the angle error peaked at 1.00°, 2.59°, and 3.37° along the x-, y-, and z-axes, respectively. Overall, both the position error and angle error had larger values compared to SIMULATION 1-1, and this result was due to the motion of the mobile robot rotating along the curved trajectory. In Fig. 24, the output linear velocity of the end-effector presented a trapezoidal shape similar to that of the command velocity profile, while the angular velocity had an overall negative value owing to the clockwise motion of the mobile robot along the curved trajectory.

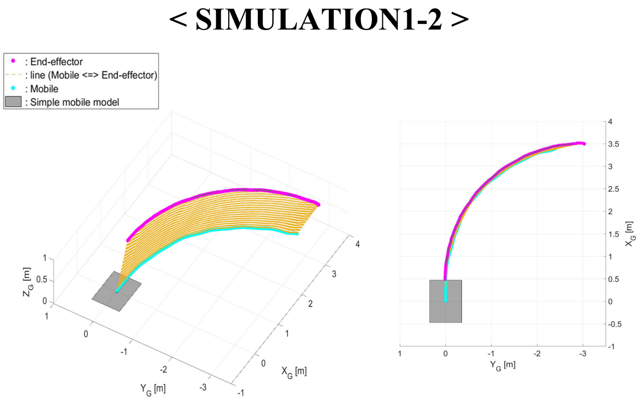

FIGURE 22. Pose graph (SIMULATION1-2).

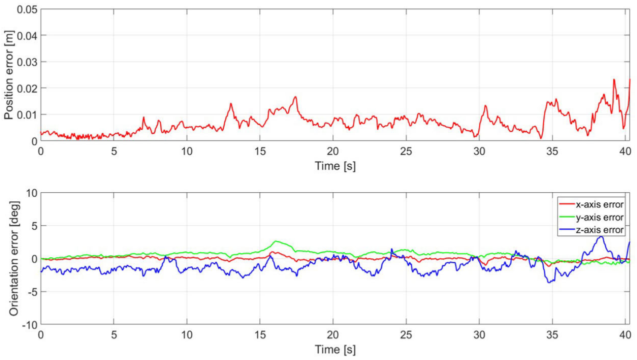

FIGURE 23. Pose error of end-effector (SIMULATION1-2).

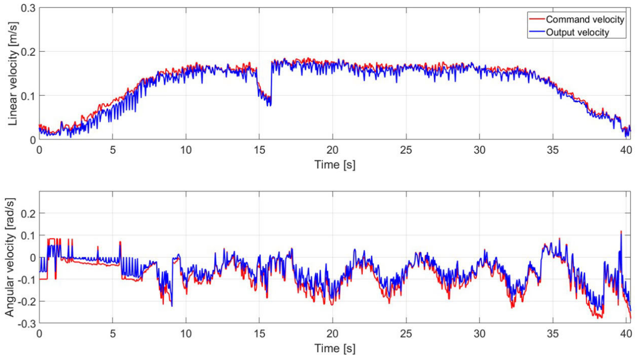

FIGURE 24. Command and output velocity of mobile robot (SIMULATION1-2).

< SIMULATION2-1 >

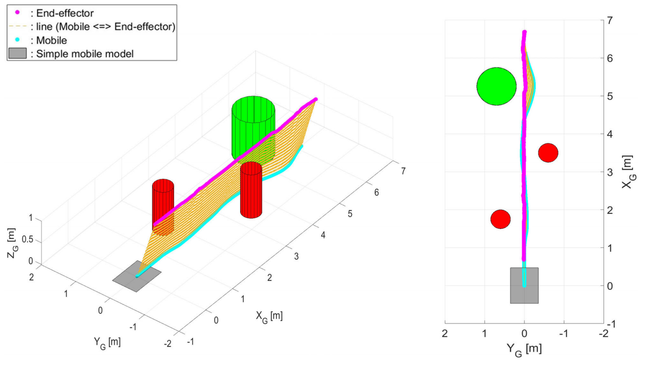

FIGURE 25. Pose graph (SIMULATION2-1).

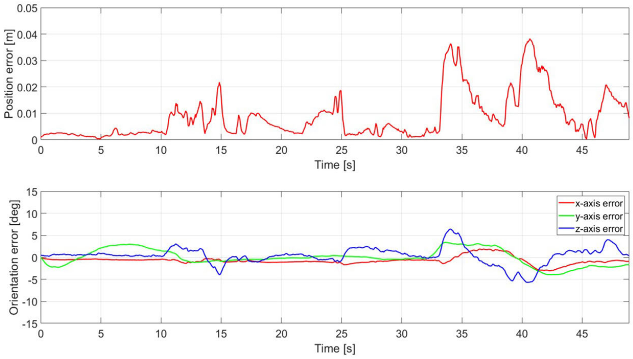

FIGURE 26. Pose error of end-effector (SIMULATION2-1).

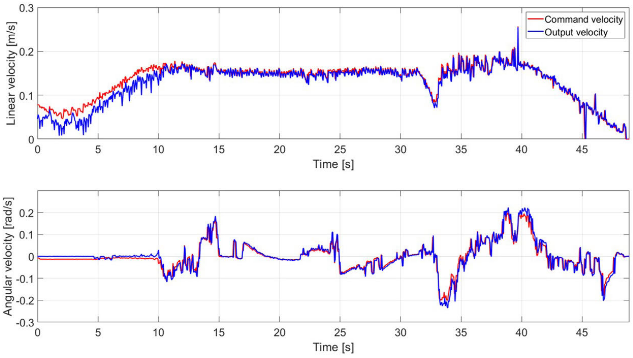

FIGURE 27. Command and output velocity of mobile robot (SIMULATION2-1).

< SIMULATION2-2 >

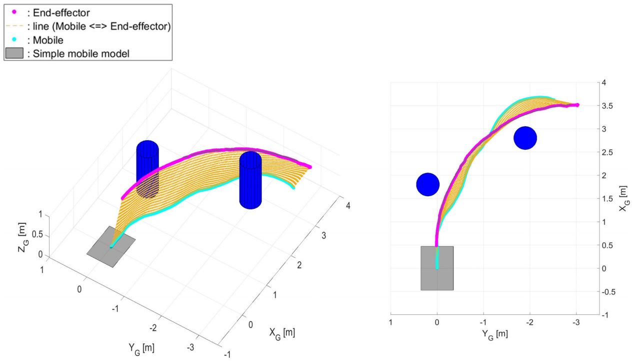

FIGURE 28. Pose graph (SIMULATION2-2).

In conclusion, in SIMULATION1, the proposed motion planning method successfully enabled the end-effector to track both straight and curved trajectories while maintaining the distance between the mobile robot and end-effector. The pose error exhibited larger values in SIMULATION1-2, where the robot drove along a curved trajectory, compared to the case of SIMULATION1-1. Thus, it was confirmed that the pose error increased owing to the angular velocity of the mobile robot. The velocity of the mobile robot in both SIMULATION1-1 and SIMULATION1-2 exhibited a shape similar to that of the velocity profile generated by the trajectory planning of the end-effector.

2) SIMULATION 2

In SIMULATION2, the motion planning method for the straight and curved trajectories of the end-effector was validated in an environment with obstacles. The primary focus at this point was to verify both the obstacle avoidance capability of the mobile robot and the tracking capability of the end-effector. The simulation results yielded three sets of data that were identical to those obtained in SIMULATION1.

a: SIMULATION 2-1

Figs. 25–27 present the results obtained when the three obstacles were placed near the straight trajectory of the end-effector. In Fig. 25, the mobile robot smoothly avoided each obstacle, making three large rotations, and the end-effector successfully followed the target trajectory. As shown in Fig. 26, the average and maximum position errors were 0.00864 m and 0.0383 m, respectively. The absolute angle errors reached peaks of 1.89°, 3.40°, and 6.42° along the x-, y-, and z-axes, respectively. Compared to the obstacle-free SIMULATION1-1, SIMULATION2-1 presented larger pose errors overall and approximately twice the maximum position error. This is attributed to the large angular velocity of the mobile robot caused by obstacle avoidance. In the velocity graph in Fig. 27, it can be observed that the mobile robot exhibited a pattern of alternating positive and negative angular velocities, as it followed an S-shaped trajectory for obstacle avoidance. The linear velocity was similar to the planned velocity of the end-effector while maintaining an appropriate distance from it.

b: SIMULATION 2-2

Figs. 28–30 present the results obtained when two obstacles were positioned near the straight trajectory of the end-effector. In Fig. 28, the end-effector successfully tracked the desired trajectory, and the mobile robot avoided the obstacles effectively while following the end-effector. In Fig. 29, the average and maximum position errors were 0.0148 m

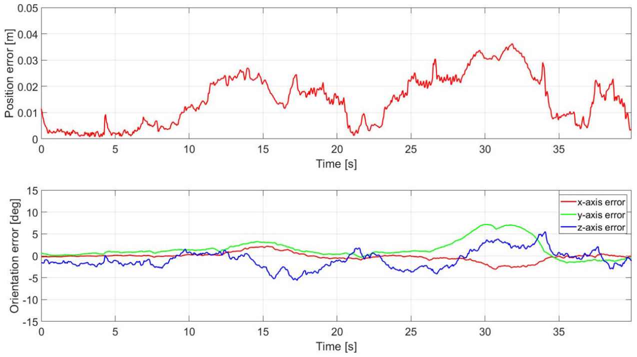

FIGURE 29. Pose error of end-effector (SIMULATION2-2).

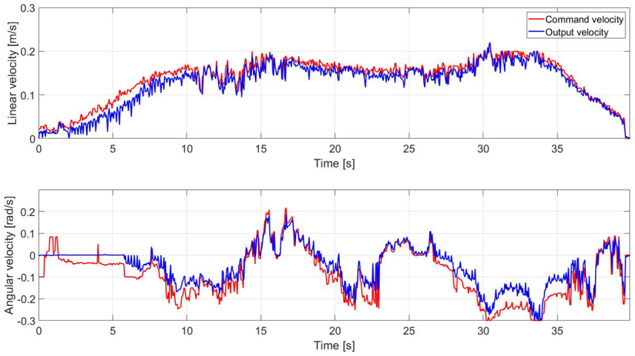

FIGURE 30. Command and output velocity of mobile robot (SIMULATION2-2).

and 0.0363 m, respectively, while the absolute values of the angle error peaked at 2.18°, 7.16°, and 5.53° for the x-, y-, and z-axes, respectively. Based on the results, when compared to the obstacle-free curved environment in SIMULATION1-2, a larger overall pose error was observed, particularly in terms of the orientation error. In the velocity profile of the mobile robot in Fig. 30, the linear velocity was similar to the planned velocity of the end-effector and the angular velocity represented the values generated to avoid obstacles while moving along the curved trajectory.

Through SIMULATION2, it was confirmed that the proposed motion planning method successfully achieved the trajectory following the end-effector and obstacle avoidance of the mobile robot. An analysis of the pose error revealed that a larger pose error was observed owing to obstacle avoidance compared to that in SIMULATION1. A noteworthy observation for both SIMULATION1 and SIMULATION2 was that as the angular velocity of the mobile robot increased, the overall pose error increased. Furthermore, the graph of the angle error for the z-axis of the end-effector presented a shape very similar to that of the angular velocity graph, but in an inverted form. The reason for the inverted shape of the graph is that the z-axis of the mobile robot and end-effector are opposite to each other. This result indicates that the end-effector was unable to maintain its position and rotated along with the mobile robot owing to the angular velocity of the mobile robot. It will be possible to improve the pose error in the future by enhancing the control performance of both the manipulator and mobile robot, particularly by achieving robust control of the manipulator base even during the movement caused by the velocity of the mobile robot.

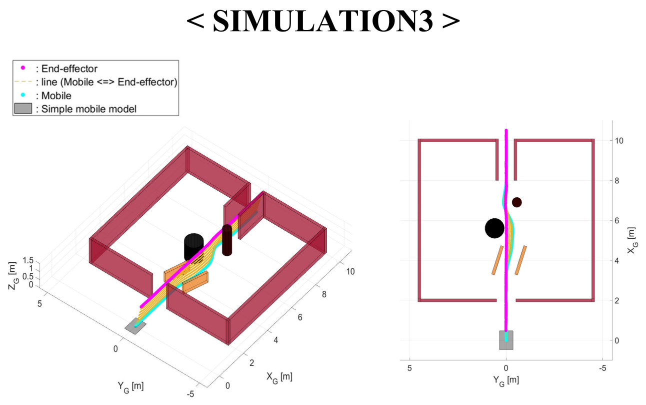

FIGURE 31. Pose graph (SIMULATION3).

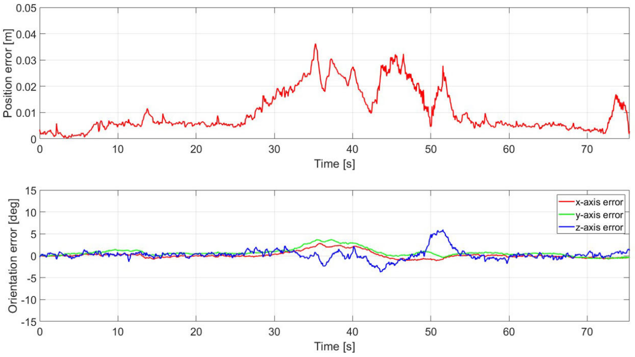

FIGURE 32. Pose error of end-effector (SIMULATION3).

3) SIMULATION 3

Fig. 31-33 present the results obtained when various obstacles were positioned near the straight trajectory of the end-effector. In Fig. 31, the mobile robot successfully navigated the short passage and avoided the sphere and cylindrical obstacles while the end-effector closely followed the desired trajectory. Fig. 32 displays the average and maximum position errors as 0.0104 m and 0.0362 m, respectively, and shows the absolute values of the angular errors peaking at 2.81°, 3.68°, and 5.97° for the x-, y-, and z-axes, respectively. In the velocity graph in Fig. 33, the linear velocity closely matched the planned velocity of the end-effector, and the angular velocity was generated to circumvent the obstacles.

In SIMULATION3, we assessed whether the mobile manipulator could navigate through the narrow passage and various obstacles on a complex map. The results confirmed that the proposed motion planning method successfully enabled the robot to follow the planned trajectory while overcoming diverse obstacles. The pose error was relatively small and comparable to the results from SIMULATION2.

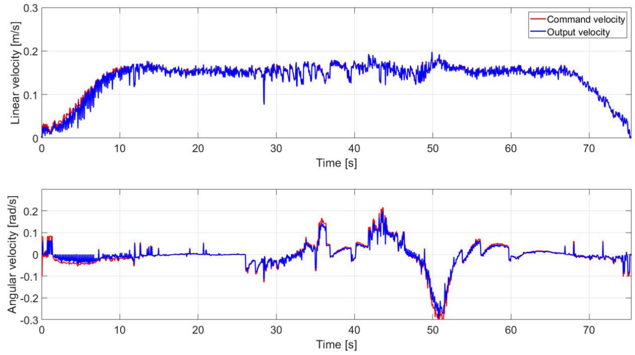

FIGURE 33. Command and output velocity of mobile robot (SIMULATION3).

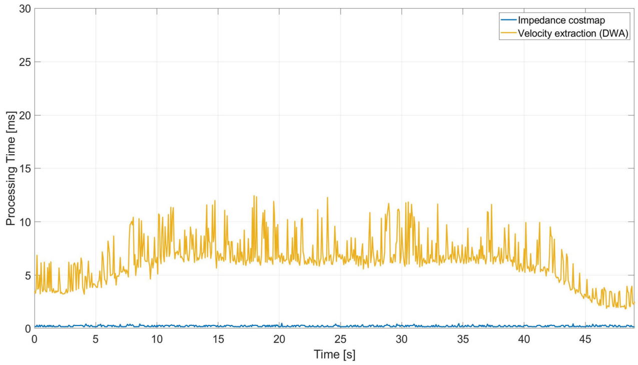

FIGURE 34. Algorithm execution time (SIMULATION1-1).

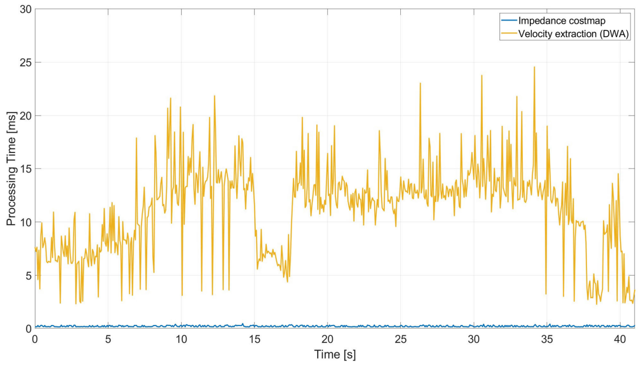

FIGURE 35. Algorithm execution time (SIMULATION1-2).

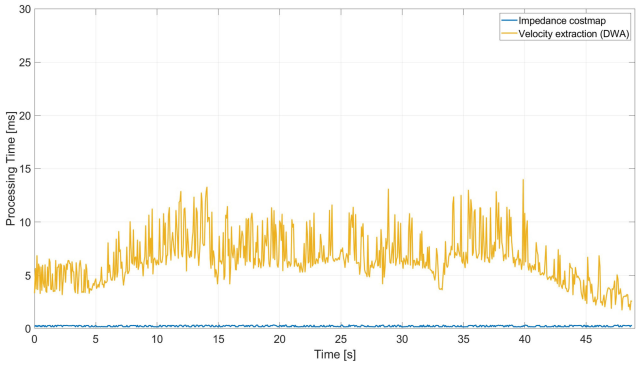

FIGURE 36. Algorithm execution time (SIMULATION2-1).

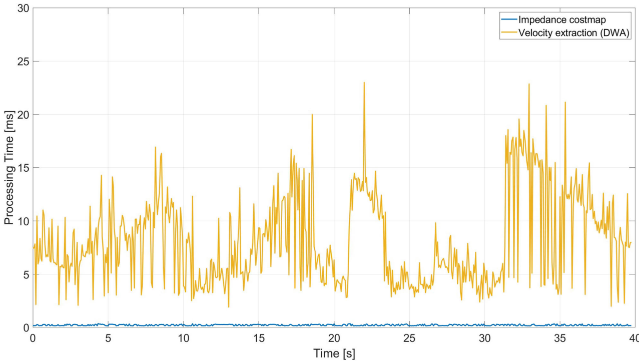

FIGURE 37. Algorithm execution time (SIMULATION2-2).

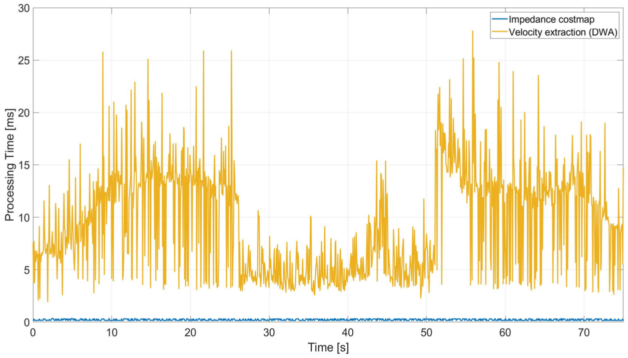

FIGURE 38. Algorithm execution time (SIMULATION3).

4) COMPUTATIONAL TIME OF ALGORITHM

We measured the algorithm processing time required for the main algorithms used in our motion planning method proposed in SIMULATION1, SIMULATION2 and SIMULATION3. Time measurements were conducted in two parts. The first part is the time required to update the energy cost of the local costmap centered at the mobile robot, which is used as the cost function of the VIEF. The second part is the time required to select the optimal velocity sample with the minimum value among all the velocity samples in the DWA algorithm by calculating the total cost for all the velocity samples based on the cost function of the updated local costmap. The measurement data for the algorithm processing times of SIMULATION1-1, SIMULATION1-2, SIMULATION2-1, SIMULATION2-2, and SIMULATION3 are presented in Figs. 34–38 and Tables 2–6. The averaged update time of the VIEF, indicated by blue dots, was 0.212 ms, 0.205 ms, 0.213 ms, 0.217 ms and 0.215 ms for SIMULATION1-1, SIMULATION1-2, SIMULATION2-1, SIMULATION2-2, and SIMULATION3, respectively. The respective average time to sort the optimal velocity sample in the DWA, indicated by yellow dots, was 6.33 ms, 11.11 ms, 6.51 ms, 8.10 ms and 9.40 ms. The update time of the VIEF was approximately 1/54-1/30 of the calculation time for selecting the optimal velocity sample within the DWA. This confirmed that the update time of the VIEF was negligible, exerting minimal impact on the overall processing time.

TABLE 2. Algorithm execution time analysis (SIMULATION1-1).

(737 data)		Min	Max	Aver.	Std. Dev.
Impedance costmap	[ms]	0.135	0.477	0.212	0.065
Velocity extraction(DWA)	[ms]	1.836	12.460	6.327	2.100

TABLE 3. Algorithm execution time analysis (SIMULATION1-2).

(621 data)		Min	Max	Aver.	Std. Dev.
Impedance costmap	[ms]	0.140	0.453	0.205	0.061
Velocity extraction(DWA)	[ms]	2.275	24.551	11.109	4.105

TABLE 4. Algorithm execution time analysis (SIMULATION2-1).

(732 data)		Min	Max	Aver.	Std. Dev.
Impedance costmap	[ms]	0.138	0.336	0.213	0.059
Velocity extraction(DWA)	[ms]	1.749	13.976	6.510	2.266

TABLE 5. Algorithm execution time analysis (SIMULATION2-2).

(597 data)		Min	Max	Aver.	Std. Dev.
Impedance costmap	[ms]	0.143	0.361	0.217	0.0609
Velocity extraction(DWA)	[ms]	1.918	22.995	8.104	4.152

TABLE 6. Algorithm execution time analysis (SIMULATION3).

(1131 data)		Min	Max	Aver.	Std. Dev.
Impedance costmap	[ms]	0.138	0.343	0.215	0.067
Velocity extraction(DWA)	[ms]	1.953	27.776	9.401	4.967

Furthermore, the respective measured total processing time of our motion planner, which added both the time data, was as follows:6.54 ms, 11.32 ms, 6.73 ms, 8.32 ms and 9.62ms. This indicates that the average time from updating the VIEF to obtaining the optimal velocity using the DWA was at most a maximum of 11.32 ms. As a result of comparing the time measurements in all simulations, the update time for the VIEF showed nearly identical values, and the DWA time did not increase in SIMULATION2 and SIMULATION3, even with obstacles to the map. Notably, even in SIMULATION3, which features complex obstacles, there was no significant increase in computing the time to select the optimal velocity sample in the DWA.

V. CONCLUSION

This paper proposed a motion planner based on a VIEF for mobile manipulator systems. When the desired trajectory of the end-effector for a specific task is planned, the motion planner considers the differences in the characteristics of the manipulator and mobile robot. The manipulator was controlled using a Cartesian PD controller to track the velocity and acceleration profiles that were planned using the LSPB. Impedance forces were applied to the mobile robot through a virtual impedance relationship between the end-effector and mobile robot. By converting these forces into the VIEF, it was possible to consider even nonholonomic mobile robots as applicable robots. The mobile robot performed motion planning and velocity control using the DWA algorithm as a local planner, which reflected the VIEF as a cost function. Consequently, the manipulator successfully followed the desired trajectory, while the mobile robot maintained a distance from the end-effector and avoided obstacles in its surroundings. Our motion planner is adaptable to the other types of mobile manipulator and various tasks, such as opening doors or moving objects, and can be further developed as a compliant motion planner for human-guided mobile manipulators in industrial settings. This is particularly useful for transporting large sheets of glass or other sizable panels.

To validate the performance of our motion planner, we conducted simulations to confirm whether the proposed motion planner worked well for both the straight and curved trajectories of the end-effector. As a result, we comprehensively checked the pose change graph of the mobile robot and end-effector, pose error of the end-effector, and velocity of the mobile robot. Specifically, the maximum position error was measured at 0.0383 m and the orientation error at 7.16° (y-axis). Thus, we validated that the end-effector tracked the desired trajectory, and that the mobile robot followed the end-effector while successfully avoiding obstacles. The pose error was greater in the case of curved trajectories than the straight trajectories and was greater when obstacles were present than when there were no obstacles.

In addition, we measured the processing time required for the main algorithms used in our motion planning method. The average update time of the VIEF was 0.212 ms. The average time to calculate the optimal velocity sample within the DWA was 8.29 ms. The update time of the VIEF was short and did not significantly affect the total processing time compared to the time required to select the optimal velocity sample. The average total processing time of our motion planner was sufficiently short at approximately 8.51 ms. Therefore, it was confirmed that online motion planning was possible.

Our motion planner has the following limitations. Firstly, because we assumed $\ddot{p}_d$ to be 0 for a short period of time during the derivation of the VIEF-related equation, the movement of p_d did not exactly match the intended impedance relationship between the manipulator and mobile robot owing to the robot's inertia. Secondly, this study was focused on applying the VIEF to a mobile manipulator platform, and an analysis on the performance variation of the motion planner based on the ratio of the spring constant K_d and damping constant B_d was not performed. Thirdly, this study did not address mathematical optimality and smoothness. Therefore, future work will involve evaluating and comparing these aspects with other existing motion planning models, particularly in environments with more complex dynamic obstacles. Additionally, we aim to conduct experiments in real-world environments to address challenges such as sensor noise, modeling inaccuracies, and the demands of real-time performance.

ACKNOWLEDGMENT

(Jung Hyun Choi and Ui Hun Sagong contributed equally to this work.)

REFERENCES

[1] U. Nehmzow, "Mobile robotics: Research, applications and challenges," in *Proc. Inst. Mech. Eng.*, London, U.K., 2001, pp. 1–4.

[2] M. H. Ali, K. Aizat, K. Yerkhan, T. Zhandos, and O. Anuar, "Vision-based robot manipulator for industrial applications," *Proc. Comput. Sci.*, vol. 133, pp. 205–212, Jan. 2018, doi: 10.1016/j.procs.2018.07.025.

[3] J. L. Outón, I. Villaverde, H. Herrero, U. Esnaola, and B. Sierra, "Innovative mobile manipulator solution for modern flexible manufacturing processes," *Sensors*, vol. 19, no. 24, p. 5414, Dec. 2019, doi: 10.3390/s19245414.

[4] A. Pratkanis, A. E. Leeper, and K. Salisbury, "Replacing the office intern: An autonomous coffee run with a mobile manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, May 2013, pp. 1248–1253.

[5] Z. Li, P. Moran, Q. Dong, R. J. Shaw, and K. Hauser, "Development of a tele-nursing mobile manipulator for remote care-giving in quarantine areas," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3581–3586.

[6] E. Zereik, A. Sorbara, G. Casalino, and F. Didot, "Autonomous dual-arm mobile manipulator crew assistant for surface operations: Force/vision-guided grasping," in *Proc. 4th Int. Conf. Recent Adv. Space Technol.*, Jun. 2009, pp. 710–715.

[7] T. Sandakalum and M. H. Ang, "Motion planning for mobile manipulators—A systematic review," *Machines*, vol. 10, no. 2, p. 97, Jan. 2022, doi: 10.3390/machines10020097.

[8] A. Rastegarpanah, H. C. Gonzalez, and R. Stolkin, "Semi-autonomous behaviour tree-based framework for sorting electric vehicle batteries components," *Robotics*, vol. 10, no. 2, p. 82, Jun. 2021, doi: 10.3390/robotics10020082.

[9] J. Shao, H. Xiong, J. Liao, W. Song, Z. Chen, J. Gu, and S. Zhu, "RRT-GoalBias and path smoothing based motion planning of mobile manipulators with obstacle avoidance," in *Proc. IEEE Int. Conf. Real-time Comput. Robot. (RCAR)*, Xining, China, Jul. 2021, pp. 217–222.

[10] C. Wang, Q. Zhang, Q. Tian, S. Li, X. Wang, D. Lane, Y. Petillot, and S. Wang, "Learning mobile manipulation through deep reinforcement learning," *Sensors*, vol. 20, no. 3, p. 939, Feb. 2020, doi: 10.3390/s20030939.

[11] V. Pilania and K. Gupta, "A hierarchical and adaptive mobile manipulator planner," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Madrid, Spain, Nov. 2014, pp. 45–51.

[12] Q. Li, Y. Mu, Y. You, Z. Zhang, and C. Feng, "A hierarchical motion planning for mobile manipulator," *IEEJ Trans. Electr. Electron. Eng.*, vol. 15, no. 9, pp. 1390–1399, Jul. 2020, doi: 10.1002/tee.23206.

[13] J. Vannoy and J. Xiao, "Real-time adaptive motion planning (RAMP) of mobile manipulators in dynamic environments with unforeseen changes," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1199–1212, Oct. 2008, doi: 10.1109/TRO.2008.2003277.

[14] P. Ogren, N. Egerstedt, and X. Hu, "Reactive mobile manipulation using dynamic trajectory tracking," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, San Francisco, CA, USA, Apr. 2000, pp. 3473–3478.

[15] N. Hogan, "Impedance control: An approach to manipulation: Part I—Theory," *J. Dyn. Syst., Meas., Control*, vol. 107, no. 1, pp. 1–7, Mar. 1985.

[16] T. Takubo, H. Arai, and K. Tanie, "Control of mobile manipulator using a virtual impedance wall," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 4, Washington, DC, USA, May 2002, pp. 3571–3576.

[17] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997, doi: 10.1109/100.580977.

[18] G. Oriolo and C. Mongillo, "Motion planning for mobile manipulators along given end-effector paths," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, pp. 2154–2160.

[19] W. Li and R. Xiong, "Dynamical obstacle avoidance of task-constrained mobile manipulation using model predictive control," *IEEE Access*, vol. 7, pp. 88301–88311, 2019, doi: 10.1109/ACCESS.2019.2925428.

[20] M. H. Korayem and H. Ghariblu, "Maximum allowable load on wheeled mobile manipulators imposing redundancy constraints," *Robot. Auto. Syst.*, vol. 44, no. 2, pp. 151–159, Aug. 2003, doi: 10.1016/s0921-8890(03)00043-5.

[21] D. Aarno, F. Lingelbach, and D. Kragic, "Constrained path planning and task-consistent path adaptation for mobile manipulators," in *Proc. 12th Int. Conf. Adv. Robot.*, Seattle, WA, USA, Oct. 2006, pp. 268–273.

[22] AgileX Robotics. (0551). *Scout 2.0 E-Book*. Accessed: Jan. 10, 2024. [Online]. Available: https://cdn.shopify.com/s/files/1/0551/0630/6141/files/SCOUT_2.0_E-BOOK.pdf?v=1637235031

[23] *FRANKA EMIKA Panda, Franka Control Interface*. Accessed: Jan. 10, 2024. [Online]. Available: https://www.frankaemika.github.io/docs/

[24] Velodyne LiDAR. (9243). *VLP-16 User Manual*. Accessed: Jan. 10, 2024. [Online]. Available: https://velodynelidar.com/wp-content/uploads/2019/12/63-9243-Rev-E-VLP-16-User-Manual.pdf

[25] Franka Emika. *Franka_Ros*. Accessed: Jan. 10, 2024. [Online]. Available: https://github.com//frankaemika/franka_ros

[26] *Ros Planning, Navigation*. Accessed: Jan. 10, 2024. [Online]. Available: https://github.com/ros-planning/navigation

[27] *Magazine*. Accessed: Jan. 10, 2024. [Online]. Available: https://github.com/magazino/move_base_flex

JUNG HYUN CHOI (Graduate Student Member, IEEE) received the B.S. degree in mechanical and information engineering from the University of Seoul, Seoul, South Korea, in 2023, where he is currently pursuing the Ph.D. degree in mechanical and information engineering. His research interests include motion planning and control of mobile manipulators.

UI HUN SAGONG (Graduate Student Member, IEEE) received the B.S. degree in mechanical and information engineering from the University of Seoul, Seoul, South Korea, in 2023, where he is currently pursuing the M.S. degree in mechanical and information engineering. His research interests include motion planning of a mobile manipulator and the control of a redundant manipulator.

JONG HOON PARK received the B.S. and M.S. degrees in mechanical and information engineering from the University of Seoul, Seoul, South Korea, in 2021 and 2023, respectively. He is currently a Research Engineer with the Robot Navigation Project, Robot Business Division, LG Electronics, Seoul.

MURIM KIM (Senior Member, IEEE) received the B.S. degree in material science and mechanical engineering from the Yanbian University of Science and Technology, Jilin, China, in 1999, and the M.S. and Ph.D. degrees in mechanical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2004 and 2008, respectively. He was a Postdoctoral Researcher with the Mechanical Engineering Research Institute, KAIST, in 2008. He was a Senior Researcher with the Research Institute of Industrial Science and Technology, Pohang, South Korea, from November 2008 to February 2016. He is currently the Director and the Chief Researcher of the Human-Centered Robotics Research and Development Division, KIRO, Pohang. His former name was Maolin Jin (Korean pronunciation is Murim Kim). His research interests include robust control of nonlinear plants, time-delay control, robot motion control, electro-hydraulic actuators, winding machines, disaster robotics, and factory automation. He serves as an Associate Editor for the *International Journal of Control, Automation, and Systems*, *Journal of Drive and Control*, and *Journal of the Korean Society for Precision Engineering*.

MYUN JOONG HWANG (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in mechanical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2001, 2003, and 2007, respectively. He has been a Professor with the Department of Mechanical and Information Engineering, University of Seoul, South Korea, since 2021. He was a Research Associate with the Mechanical Engineering Research Institute, KAIST, in 2007. From 2008 to 2009, he was a Research Associate with the Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH, USA. He was a Senior Research Engineer with the Manufacturing Technology Center, Samsung Electronics Company Ltd., Suwon, South Korea, from 2010 to 2013. He was an Assistant Professor with the School of Mechanical and Automotive Engineering, Halla University, Wonju, South Korea, from 2013 to 2015. He was an Assistant/Associate Professor with the Department of Mechanical Engineering, Korea National University of Transportation, Chungju, South Korea, from 2015 to 2021. His research interests include robot motion planning and control, manipulation, autonomous driving, and robot vision.