electronics

MDPI

Article

End Effector Driven Whole Body Trajectory Tracking for Mobile Manipulator Based on Linear and Angular Motion Decomposition

Ji-Wook Kwon *, Taeyoung Uhm, Ji-Hyun Park, Jongdeuk Lee and Jeong Hwan Hwang

Smart Mobility R&D Division, Korea Institute of Robotics and Technology Convergence, Pohang 37666, Republic of Korea
* Correspondence: jwkwon@kiro.re.kr

Abstract

This paper proposes an end-effector (EE) driven whole-body trajectory tracking control algorithm for wheeled mobile manipulators based on linear and angular motion decomposition. Instead of solving a high-dimensional optimization problem across all degrees of freedom, the proposed method formulates the control objective directly in the EE space and decomposes the required motion into planar linear, vertical, and angular components. To address redundancy between the mobile base and the manipulator under non-holonomic constraints, a control authority switching strategy with a radial blending function is introduced. This approach eliminates ambiguity in control allocation while preventing abrupt switching near workspace boundaries. The kinematic controller guarantees exponential convergence of position and orientation errors without requiring a full dynamic model. Numerical simulations demonstrate stable tracking performance in three-dimensional space. Compared with a quadratic programming-based whole-body controller, the proposed method achieves comparable or faster error convergence while reducing computational burden by more than 13 times on average. These results indicate that the proposed EE-driven framework provides a computationally efficient and practically deployable solution for real-time mobile manipulator control.

Keywords: mobile manipulator; whole-body control; end-effector driven control; motion decomposition

Check for updates

Academic Editors: Janos Botzheim, Imran Ghous and Jahanzeb Akhtar

Received: 11 February 2026
Revised: 23 March 2026
Accepted: 24 March 2026
Published: 26 March 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Wheeled mobile manipulators (WMMs) have attracted increasing attention in various industrial fields, including manufacturing, logistics, service, and healthcare, due to their capability to significantly expand the workspace of fixed-base manipulators and to augment mobile robots with manipulation functionalities such as loading and unloading [1–3]. By integrating mobility and manipulation, WMMs can not only extend the operational range of conventional manipulators but also enhance the transportation and logistics capabilities of mobile platforms with dexterous handling functions. Furthermore, WMMs can flexibly adapt to changes in working environments and task requirements, thereby contributing substantially to improved productivity and operational efficiency.

In the early stage of WMM deployment, transportation and manipulation were operated independently and executed in a sequential operation. However, to fully exploit the potential of WMMs, there has been a growing demand for whole-body control (WBC), in

which the mobile base and the manipulator are controlled simultaneously. WBC represents the mobile manipulator as a holistic structure by integrating all degrees of freedom (DOF) of the mobile base and the manipulator, thereby significantly enhancing motion speed and fluidity compared with conventional independent control schemes [4–6]. In other words, since manipulation can be performed while the mobile base is in motion, the overall task cycle time can be reduced, the naturalness of motion can be improved, and behaviors that cannot be realized under independent control architectures can be achieved.

However, WMM inherently exhibits a high system order due to the integration of the mobile platform and the manipulator. Moreover, the redundancy in control inputs must be addressed simultaneously. As a result, the WBC of WMMs poses significant challenges in terms of modeling, coordination, and control design complexity.

To address the complex control problem of highly redundant WMMs, WBC algorithms that integrate all DOF of the mobile manipulator, along with model predictive control (MPC) schemes that compute optimal control inputs under constraints while maintaining the dynamic balance of inherently unstable systems, have been proposed [6–11]. These approaches have demonstrated superior performance in ensuring dynamic stability and enabling the integration of locomotion–manipulation hybrid motions. However, when model-based mathematical optimization techniques have been employed, the associated computational burden often results in long processing times, making it difficult to respond to rapidly varying trajectories in real time. In addition, conventional model-based control strategies require accurate dynamic models of the robot, which necessitates additional compensation mechanisms to account for system uncertainties and modeling errors.

To overcome these limitations, learning-based Physical AI approaches, such as deep reinforcement learning and imitation learning, have emerged to directly learn complex control policies [5,12–16]. In particular, studies on adaptive neural network control for robots equipped with series elastic actuators have contributed to mitigating model uncertainties, demonstrating that adaptive and learning-based control strategies can effectively address modeling inaccuracies. It is worth noting that recent advances in efficient adaptive and learning-based control have demonstrated that control policies can be learned offline from collected experiences, substantially alleviating the dependence on high-performance computing hardware during deployment [15,16]. For mechatronic applications utilizing compact network architectures such as MLPs and LSTMs, real-time inference can be achieved well within the millisecond range even without dedicated GPU hardware. However, such deployment assumes that the training data sufficiently covers the operational conditions, and frequent task reconfiguration in industrial settings—such as changes in kinematic parameters, payload, and workspace geometry—may necessitate retraining. Furthermore, in multi-process robotic systems, shared CPU resources can increase inference latency beyond isolated benchmark levels. These practical considerations, together with the absence of formal stability guarantees for neural network-based controllers, motivate the development of analytical control frameworks that ensure tracking performance a priori without dependence on training data or computational environment assumptions. Furthermore, advanced optimization-based integrated methodologies—such as hierarchical optimization, holistic system integration design, and real-time reactive control—have been investigated for diverse environments and robotic platforms [17–21]. Among these approaches, algorithms based on quadratic programming (QP) have been proposed to generate optimal control commands for integrated models of mobile bases and manipulators [19–21]. These studies have progressed beyond improving model accuracy and have aimed at enhancing the strategic intelligence and task execution capability of robotic systems. However, since such methods must still solve optimization problems for high DOF systems with significant control redundancy, they inherently require high-performance optimization solvers and

dedicated computational hardware. Consequently, these approaches impose substantial computational demands, limiting the development of cost-effective and widely deployable solutions. Moreover, the iterative nature of these optimization-based solvers introduces inherent uncertainty in computation time, as the number of iterations required for convergence varies depending on the current system state and constraint configuration. This time variability poses a fundamental challenge for deterministic real-time control, particularly in high-frequency applications (above 1 kHz) where strict cycle-time budgets must be consistently met. Consequently, ensuring real-time feasibility with optimization-based approaches often necessitates dedicated high-performance computing hardware or careful worst-case timing analysis, both of which increase system cost and integration complexity.

Considering the existing research outcomes and the practical demands in real-world applications, it is essential to reduce the influence of high DOF and to decrease the computational complexity and processing load of control algorithms in order to facilitate the rapid deployment of WBC for WMMs in practical environments. In particular, the control framework should be designed to operate independently of high-performance processors, thereby enabling cost-effective and scalable implementation in real industrial settings.

Therefore, this paper proposes an end-effector (EE)-driven whole-body tracking control (WBTC) framework for a mobile manipulator based on linear and angular motion decomposition. Regardless of whether WBC is employed, the fundamental objective of WMM control is to enable the EE to perform tasks at the desired position. Accordingly, the EE can be regarded as the primary reference for representing the motion of the WMM. By adopting this perspective, it is no longer necessary to separately define a target pose for the mobile base, move the base accordingly, and subsequently control the manipulator in a sequential manner. Instead, an EE trajectory tracking algorithm is proposed so that the WMM can be controlled in a unified manner through WBC with respect to a single reference, namely the EE. This paper addresses a trajectory tracking control algorithm that enables the EE to follow a trajectory defined as a set of positions in three-dimensional space. (Note that the orientation of the EE—i.e., roll, pitch, and yaw—is not considered in this study.) To this end, a kinematics-based trajectory tracking control method presented in [22,23] is adopted to design the EE velocity control algorithm. The resulting velocity command is then realized by decomposing the EE motion into linear, vertical, and rotational components using the motion decomposition technique proposed in [24], and reconstructing the overall motion accordingly. During this process, control redundancy arises because the WMM consists of a non-holonomic mobile base and a multi-DOF manipulator. To eliminate the ambiguity caused by redundancy and to minimize control complexity, a control authority switching algorithm is proposed. Although the control authority switching and command blending algorithms in [16] were formulated based on the boundary of the workspace, such a boundary-based approach is not suitable for trajectory tracking, as it defines switching conditions solely with respect to the relationship between the EE position and the workspace boundary, leaving the possibility that the EE may exit the workspace. To resolve this issue, the proposed switching algorithm is reformulated based not on workspace boundaries but on representative reference positions within predefined regions.

The proposed EE-driven WBTC algorithm is summarized in Figure 1.

Admittedly, the proposed control law is derived based on a kinematic model that generates velocity commands, and therefore does not explicitly account for the dynamic characteristics of the WMM. This may be regarded as a limitation in terms of dynamic performance analysis. However, considering that most commercial robotic platforms are equipped with proprietary low-level dynamic controllers and do not allow direct access to torque-level control by external users, the proposed approach can be interpreted as a

practical and platform-agnostic solution that can be readily applied to a wide range of robotic systems.

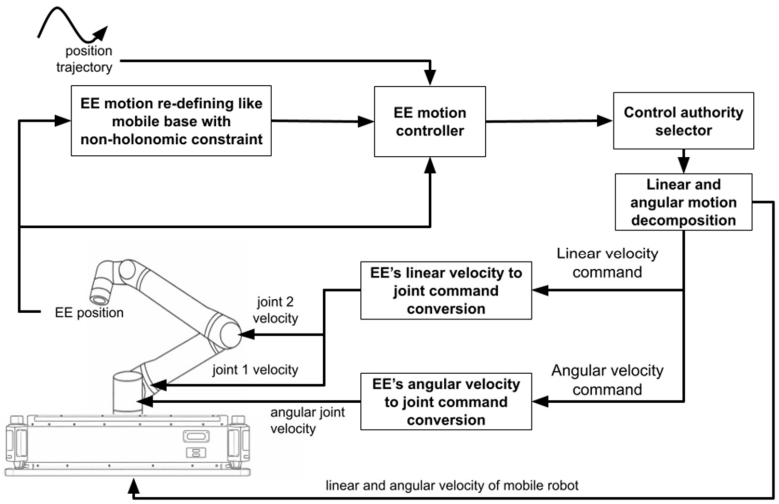


Figure 1. Block diagram of the proposed EE-driven WBTC algorithm.

This paper includes verification results through numerical simulations to demonstrate the performance of the proposed algorithm. To minimize the gap between numerical simulation and real robotic systems, the characteristics of actual robots are reflected in the simulation setup. The performance evaluation of the proposed algorithm is conducted through comparison with a conventional optimization-based algorithm. The results of mission execution under the same scenario on the same processor are compared in terms of error convergence behavior, computation time, and generated control inputs.

The contributions of the proposed method are summarized as follows.

- Reduced control complexity and computational burden: By mitigating the impact of high DOF and minimizing algorithmic complexity and computational load, the proposed method enables processor-independent operation. This facilitates the development of cost-effective and widely deployable solutions applicable to a broad range of robotic platforms.
- Computational efficiency and embedded deployability: The proposed algorithm relies exclusively on elementary arithmetic and trigonometric operations, without requiring linear algebra libraries for large-scale matrix operations or iterative optimization solvers. This characteristic makes the algorithm readily deployable on resource-constrained embedded platforms, such as ARM-based industrial controllers or microcontroller units (MCUs), without the need for GPU acceleration or dedicated optimization hardware. The low and predictable computational footprint of the proposed method thus constitutes a significant practical advantage for real-world mobile manipulator deployment.
- Ease of deployment on commercial robotic systems: Since most commercial robots are equipped with proprietary low-level dynamic controllers that restrict direct external torque-level control, the proposed kinematics-based control law can be readily implemented at the velocity-command level. This enables broad applicability across diverse commercial robotic platforms without requiring modification of internal dynamic controllers.
- EE-driven unified control framework: By adopting the end-effector as the primary reference, the proposed approach eliminates the need to separately define a target pose for the mobile base and subsequently control the manipulator in a sequential manner.

Instead, the WMM is controlled in a unified framework with respect to the EE, thereby enhancing motion fluidity, coordination, and overall operational efficiency.

- Stable control authority switching mechanism: A radial function–based control authority switching algorithm is proposed, where the switching condition is defined with respect to representative reference positions within predefined regions rather than workspace boundaries. This approach mitigates potential instability during control authority transitions, alleviates constraints associated with strict workspace boundary definitions, and prevents the EE from exiting the admissible workspace during trajectory tracking.

The remainder of this paper is organized as follows. Section 2 describes the overall system configuration and presents the kinematic modeling of the proposed framework based on linear and angular motion decomposition. Section 3 introduces the kinematic motion controller designed to ensure stable trajectory tracking of the EE, along with a theoretical proof of error convergence. Section 4 proposes a novel control authority switching algorithm and a blending function scheme to address the redundancy problem between the mobile base and the manipulator. Section 5 presents simulation results that evaluate the trajectory tracking performance and computational efficiency of the proposed method, and compares them with those of a conventional quadratic programming (QP)-based control approach to demonstrate its effectiveness. Finally, Section 6 concludes the paper.

2. Motion Decomposition

2.1. System Description

The WMM consists of a mobile robot subject to non-holonomic constraints and a multi-DOF manipulator, as illustrated in Figure 2. As discussed in the Introduction, both subsystems contribute to the motion of the EE; therefore, it is necessary to explicitly define the motion components associated with each subsystem. To clearly present the proposed motion decomposition strategy, Figure 2 depicts the mobile base and the manipulator separately, highlighting their respective roles in generating the overall EE motion.

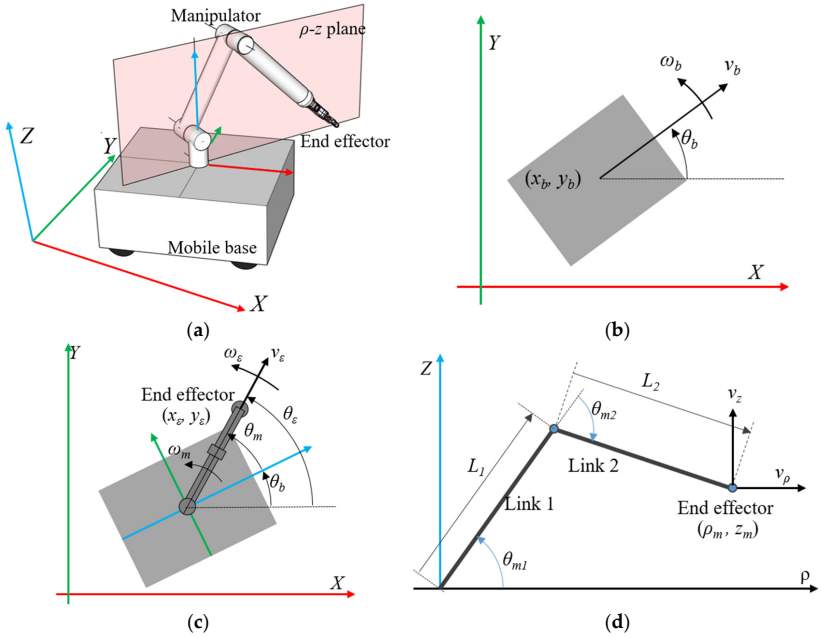


Figure 2. The system parameters of the WMM: (**a**) Overall configuration of the wheeled mobile manipulator and the ρ–z motion representation plane; (**b**) Kinematic model of the non-holonomic mobile base; (**c**) Combined planar motion representation of the mobile base and manipulator at the EE level; (**d**) ρ–z plane kinematic relationship of the manipulator and link geometry for motion decomposition.

Figure 2a presents an overall schematic view of the mobile manipulator. As shown in Figure 2a, for the convenience of algorithmic description, the coordinate frame of the mobile base is aligned with that of the manipulator for the proposed linear and angular motion decomposition. Figure 2b illustrates the kinematic characteristics of the mobile base, highlighting its non-holonomic constraint. Figure 2c describes the kinematic relationship of the manipulator in the ρ–z plane. For the proposed motion decomposition framework, the motion of the EE is expressed in the ρ–z plane, as depicted in Figure 2c. Figure 2d defines the EE coordinates based on the positional relationship between the mobile base and the manipulator. The variables indicated in Figure 2 and used throughout this paper are defined in Table 1.

Table 1. System variables.

Item	Definition	Item	Definition
ρ–z plane	The local plane aligned with the manipulator link orientation.	v_ε	The horizontal velocity of the EE. (m/s)
(x_b, y_b, z_b)	The position of the mobile base. (m, m, m)	$v_{\varepsilon z}$	The vertical velocity of the EE. (m/s)
(x_t, y_t, z_t)	The position of the trajectory. (m, m, m)	ω_ε	The angular velocity of the EE. (rad/s)
θ_b	The orientation of the mobile base. (rad)	(ρ_m, z_m)	The position of EE in the ρ–z plane. (m, m)
v_b	The linear velocity of the mobile base. (m/s)	v_ρ	EE's linear velocity in the ρ direction on the ρ–z plane. (m/s)
ω_b	The angular velocity of the mobile base. (rad/s)	v_z	EE's vertical velocity in the z direction on the ρ–z plane. (m/s)
$(x_\varepsilon, y_\varepsilon, z_\varepsilon)$	The position of the EE. (m, m, m)	ω_m	The angular velocity of the manipulator. (rad/s)
θ_ε	The orientation of the EE. (rad)	L_1, L_2	The lengths of each link of the manipulator. (m)
θ_m	The orientation of the manipulator. (rad)	θ_{m1}, θ_{m2}	The joint angles of the manipulator. (rad)

The variables are related as follows. First, v_ρ and v_ε are defined along the same direction. In addition, $z_\varepsilon = z_m$, where both variables represent the height of the manipulator. As previously mentioned, the mobile base is modeled as a non-holonomic mobile platform, as illustrated in Figure 2b. Accordingly, it exhibits the following kinematic characteristics.

$$\dot{x}_b = v_b \cos\ \theta_b,\ \dot{y}_b = v_b \sin\ \theta_b,\ \dot{\theta}_b = \omega_b. \tag{1}$$

The mobile base cannot generate motion in the z-direction. Therefore, the mobile base cannot influence the vertical movement of the EE. This implies that the redundancy of the control input in the WBTC system appears only in the horizontal motion. Accordingly, the subsequent motion decomposition and control authority switching are performed primarily with respect to the horizontal motion.

As shown in Figure 2b,c the linear velocity component v_ε and the angular velocity component ω_ε of the EE are expressed as the combined contributions of the mobile base and the manipulator, and can therefore be represented as follows.

$$v_\varepsilon = v_b \cos\ (\theta_m) + v_\rho \tag{2a}$$

$$\omega_\varepsilon = \omega_b + \omega_m \tag{2b}$$

To ensure a unified representation of motion, the movement of the manipulator is not expressed in Cartesian coordinates (x, y, z). Instead, it is formulated using v_ε and ω_ε, in a form analogous to the kinematic model of the mobile base given in (1). Using the variables defined in Figure 2 and Table 1, the kinematic behavior of the EE can be expressed as follows.

$$\begin{bmatrix} \dot{x}_\varepsilon \\ \dot{y}_\varepsilon \\ \dot{z}_\varepsilon \\ \dot{\theta}_\varepsilon \end{bmatrix} = \begin{bmatrix} \cos(\theta_\varepsilon) & 0 & -\rho_m \sin(\theta_\varepsilon) \\ \sin(\theta_\varepsilon) & 0 & \rho_m \cos(\theta_\varepsilon) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_\varepsilon \\ v_z \\ \omega_\varepsilon \end{bmatrix} \tag{3}$$

Here, it can be observed that the vertical motion can be realized independently of the planar motion components.

Note: It should be noted that the kinematic formulation in (3) inherently embeds the coordinate transformation between the inertial frame and the EE motion direction. As can be seen in (3), the planar velocity components of the EE in the inertial frame are expressed as $v_\varepsilon \cos(\theta_\varepsilon)$ and $v_\varepsilon \sin(\theta_\varepsilon)$, which correspond to the elements of a standard two-dimensional rotation matrix parameterized by the EE orientation θ_ε. Accordingly, the transformation from the local velocity v_ε of the EE to the inertial-frame velocity components is realized through the orientation angle θ_ε without requiring an explicit rotation matrix. In addition, as confirmed in (3), the vertical velocity component v_z is structurally decoupled from the planar components. Therefore, the proposed formulation enables each motion component to be designed and controlled independently without the need for a full three-dimensional rotation matrix that maps between the inertial and EE coordinate frames. This structural decoupling eliminates inter-axis coupling and thereby allows coordinated trajectory tracking to be achieved through the independent design of each decomposed velocity component.

2.2. Motion Decomposition of Manipulator

For motion decomposition, the linear and angular motion decomposition method presented in [16] is adopted. In [16], the manipulator motion is expressed in terms of (ρ_m, z_m, θ_m) rather than Cartesian coordinates (x, y, z), which allows the decomposition strategy to be directly applied in this study. Although [16] primarily addressed horizontal motion of the mobile manipulator and omitted vertical motion, the present work extends the framework to incorporate vertical motion in order to enable full three-dimensional EE trajectory tracking. Accordingly, we demonstrate that when v_ρ and v_z are designed based on (ρ_m, z_m, θ_m), the resulting motion can be implemented through the joint movements of the manipulator shown in Figure 2d, following the decomposed motion policy proposed in [16].

Note: The decomposition of EE motion of the mobile manipulator into linear and angular components can be supported by (1) and Figure 2c. First, as mentioned in the Introduction, this paper considers the generation of velocity commands based on a kinematic model, under the assumption that the velocity commands are rapidly implemented by the robot manufacturers' low-level dynamic controllers. Next, as shown in (1), the translational and rotational velocity commands of the mobile base are independently commanded. Furthermore, as depicted in Figure 2c, the horizontal position of the EE can be represented by independent translational and angular velocity inputs. Based on these two observations, it can be concluded that the linear and angular motions can be decomposed when generating kinematics-based control inputs for the mobile manipulator.

The variables (ρ_m, z_m) and the joint angles (θ_{m1}, θ_{m2}) satisfy the following relationships [16].

$$\theta_{m2} = -\arccos\left(\frac{\rho_m^2 + z_m^2 - L_1^2 - L_2^2}{2L_1L_2}\right) \tag{4a}$$

$$\theta_{m1} = \operatorname{atan2}(z_m, \rho_m) - \operatorname{atan2}(L_2 \sin\theta_{m2}, L_1 + L_2 \cos\theta_{m2}) \tag{4b}$$

In (4a), θ_{m2} has two potential solutions of $+/-$ arccos(·). Since the solution is fixed to the elbow down configuration to ensure control continuity with avoiding singular position,

the negative sign on the right side of (4a) is chosen. Also, v_ρ, v_z and the time derivatives of θ_{m1}, θ_{m2} have the following relationship.

$$\begin{bmatrix} \dot{\theta}_{m1} & \dot{\theta}_{m2} \end{bmatrix}^T = \mathbf{J}^{-1} \begin{bmatrix} v_\rho & v_z \end{bmatrix}^T, \tag{5}$$

where,

$$\mathbf{J} = \begin{bmatrix} -(L_1 \sin\ \theta_{m1} + L_2 \sin\ (\theta_{m1} + \theta_{m2})) & -L_2 \sin\ (\theta_{m1} + \theta_{m2}) \\ L_1 \cos\ \theta_{m1} + L_2 \cos\ (\theta_{m1} + \theta_{m2}) & L_2 \cos\ (\theta_{m1} + \theta_{m2}) \end{bmatrix}. \tag{6}$$

By (6) time derivatives of θ_{m1}, θ_{m2} are described as follows:

$$\dot{\theta}_{m1} = \frac{1}{L_1 L_2 \sin\ \theta_{m2}} \left[L_2 \cos\ (\theta_{m1} + \theta_{m2}) v_\rho + L_2 \sin\ (\theta_{m1} + \theta_{m2}) v_z \right] \tag{7a}$$

$$\dot{\theta}_{m2} = \frac{1}{L_1 L_2 \sin\ \theta_{m2}} \left[-(L_1 \cos\ \theta_{m1} + L_2 \cos\ (\theta_{m1} + \theta_{m2})) v_\rho - (L_1 \sin\ \theta_{m1} + L_{m2} \sin\ (\theta_{m1} + \theta_{m2})) v_z \right] \tag{7b}$$

The denominator in (7b) indicates the kinematic sensitivity of the manipulator due to sin(·) function. By consistently selecting the elbow-down configuration ($\theta_2 < 0$) as in (4), the singularity at $\theta_2 = 0$ is prevented.

Therefore, the process described in (4)–(7) shows that, once v_ρ, v_z are given, they can be directly transformed into the corresponding joint velocities. Hence, the manipulator can be effectively controlled solely through the design of v_ρ, v_z.

Note: As demonstrated in Figure 2 and (5)–(7), these decomposed velocity components are quantitatively mapped to the individual joint velocities of the manipulator $\dot{\theta}_{m1}$ and $\dot{\theta}_{m2}$ through the inverse operation of the Jacobian matrix **J**. In this process, horizontal and vertical motions are mathematically decoupled through the respective rows of the **J**, allowing for independent control of each motion component without inter-axis interference or reliance on complex numerical optimization. While joint variables can be obtained from the EE position via standard inverse kinematics, directly solving for all angles from a three-dimensional position generally requires handling full system redundancy. In contrast, by representing motion components in this structured form, the proposed decomposition reduces the effective DOF involved in the computation, thereby simplifying the control algorithm while clearly defining the contribution of each component to the final trajectory of the EE.

Note: It should be noted that the proposed decomposition framework does not require an explicit rotation matrix for transforming between the inertial coordinate frame and the EE coordinate frame. This is attributed to the fact that the motion representation adopted in this study, which is based on v_ε, θ_ε, and v_z, inherently encodes the directional information through the trigonometric terms cos (θ_ε) and sin (θ_ε) in (3), which effectively serve as the rotation matrix elements. Moreover, since the vertical motion is structurally independent from the planar motion as confirmed in (3) and (5)–(7), coordinated control of the different motion components is achieved not through a unified rotation matrix based transformation but through the independent and parallel design of each decomposed velocity component. The validity of this structural independence is formally verified by Theorem 1 in Section 3, which guarantees that the position errors in all axes converge to zero under the proposed independent control law.

3. Kinematic Motion Controller of EE

A motion controller is designed to enable the EE to track a given trajectory. To formulate the EE motion control law, the following position error variables are defined.

$$e_z = z_t - z_\varepsilon,\ e_x = x_t - x_\varepsilon,\ e_y = y_t - y_\varepsilon, \tag{8}$$

where, the time derivatives of the position errors are given as follows.

$$\begin{aligned} \dot{e}_z &= \dot{z}_t - \dot{z}_\varepsilon \\ &= \dot{z}_t - v_z \end{aligned} \tag{9a}$$

$$\begin{aligned} \dot{e}_x &= \dot{x}_t - \dot{x}_\varepsilon \\ &= \dot{x}_t - v_\varepsilon \cos(\theta_\varepsilon) + \rho\omega_\varepsilon \sin(\theta_\varepsilon) \end{aligned} \tag{9b}$$

$$\begin{aligned} \dot{e}_y &= \dot{y}_t - \dot{y}_\varepsilon \\ &= \dot{y}_t - v_\varepsilon \sin(\theta_\varepsilon) - \rho\omega_\varepsilon \cos(\theta_\varepsilon) \end{aligned} \tag{9c}$$

The control objective is to ensure that the error variables defined in (8) and (9) asymptotically converge to zero. To achieve this objective, the following control inputs are designed.

$$v_{zc} = \dot{z}_t + k_z e_z \tag{10a}$$

$$v_{\varepsilon c} = X\cos(\theta_\varepsilon) + Y\sin(\theta_\varepsilon) \tag{10b}$$

$$\omega_{\varepsilon c} = \dot{\theta}_d + k_\theta e_\theta \tag{10c}$$

Here, k_z and k_θ are positive constants selected by user, and X, Y are defined as follows.

$$X = \dot{x}_t + k_x e_x,\ Y = \dot{y}_t + k_y e_y, \tag{11}$$

where k_x and k_y are positive constants determined by the user. Theorem 1 verifies that, under the control inputs defined in (10) and (11), the EE is guaranteed to follow the prescribed trajectory.

Theorem 1. *By applying the control inputs in (10a) and (10b) to the EE of the mobile manipulator represented in (3) and Figures 1 and 2, the position errors in (8) converge to zero, allowing the EE to track the given trajectory.*

Proof of Theorem 1. To show that the position error in (8) converges to zero over time, the control input is designed so that position errors in (8) become as follows:

$$\dot{e}_z = -k_z e_z,\ \dot{e}_x = -k_x e_x, \text{ and } \dot{e}_y = -k_y e_y. \tag{12}$$

Accordingly, v_z, $v_\varepsilon cos\,(\theta_\varepsilon)$ and $v_\varepsilon sin\,(\theta_\varepsilon)$ at the end of the right-hand side of (9) are designed as follows:

$$v_z = \dot{z}_t + k_z e_z \tag{13a}$$

$$v_\varepsilon \cos(\theta_\varepsilon) = \dot{x}_t + \rho\omega_\varepsilon \sin(\theta_\varepsilon) + k_x e_x \tag{13b}$$

$$v_\varepsilon \sin(\theta_\varepsilon) = \dot{y}_t - \rho\omega_\varepsilon \cos(\theta_\varepsilon) + k_y e_y \tag{13c}$$

As applying (13) to (8), (12) can be obtained, thereby showing that the position errors converge to zero as time goes on. As can be seen in (13a), v_z, velocity input in vertical direction, is determined directly. To obtain the linear velocity, v_ε, (13b) and (13c) are multiplied by $cos\,(\theta_\varepsilon)$ and $sin\,(\theta_\varepsilon)$ respectively, and then added together as in followings.

$$\begin{aligned} v_\varepsilon \cos^2(\theta_\varepsilon) + v_\varepsilon \sin^2(\theta_\varepsilon) = {} & \cos^2(\theta_\varepsilon)\{\dot{x}_t + \rho\omega_\varepsilon \sin(\theta_\varepsilon) + k_x e_x\} \\ & + \sin^2(\theta_\varepsilon)\{\dot{y}_t - \rho\omega_\varepsilon \cos(\theta_\varepsilon) + k_y e_y\} \end{aligned}$$

$$
\begin{aligned}
v_\varepsilon\{\cos^2(\theta_\varepsilon)+\sin^2(\theta_\varepsilon)\} &= \cos(\theta_\varepsilon)\{\dot{x}_t+k_x e_x\}+\rho\omega_\varepsilon\sin(\theta_\varepsilon)\cos(\theta_\varepsilon)\\
&\quad+\sin(\theta_\varepsilon)\{\dot{y}_t+k_y e_y\}-\rho\omega_\varepsilon\sin(\theta_\varepsilon)\cos(\theta_\varepsilon)\\
v_\varepsilon &= \cos(\theta_\varepsilon)\{\dot{x}_t+k_x e_x\}+\sin(\theta_\varepsilon)\{\dot{y}_t+k_y e_y\}\\
v_\varepsilon &= X\cos(\theta_\varepsilon)+Y\sin(\theta_\varepsilon)
\end{aligned}
$$

Through the process of (12) and (13), it can be shown that when the proposed control algorithm is applied to the mobile manipulator's EE, the position error variables converge to zero. □

Although Theorem 1 guarantees the convergence of the position error to zero, this result presupposes that the EE orientation is properly aligned toward the trajectory. Therefore, the desired orientation of the EE is designed based on [14,15] as follows.

$$\theta_d = \arctan2(Y, X) \tag{14}$$

To ensure that θ_ε converges toward θ_d, the orientation error e_θ is defined as follows.

$$e_\theta = \theta_d - \theta_\varepsilon \tag{15}$$

The time derivative of e_θ is as follows.

$$\dot{e}_\theta = \dot{\theta}_d - \omega_\varepsilon \tag{16}$$

To drive e_θ to zero, ω_ε can be designed as in (10c). Whether the control command in (10c) ensures the convergence of e_θ to zero can be verified by examining if e_θ exhibits the following closed-loop behavior, as shown in (12) of Theorem 1.

$$\dot{e}_\theta = -k_\theta e_\theta \tag{17}$$

By substituting ω_ε defined in (10c), into (16), we obtain (17). Therefore, it can be concluded that, under the control input ω_ε in (10c) converges to θ_d.

4. Control Authority Switching Algorithm

As shown in (2), both v_ε and ω_ε are expressed as combinations of the velocity components of the mobile base and the manipulator. Therefore, redundancy must be addressed when controlling the EE. A control authority switching algorithm is proposed to resolve the redundancy issue.

In the proposed strategy, the manipulator workspace is assumed to be a spherical region centered at the manipulator base mount. Based on this assumption, a control authority allocation algorithm is proposed. The fundamental concept of the algorithm is to determine whether the mobile base or the manipulator should take control authority depending on whether the desired trajectory lies within the manipulator workspace. If the desired position is located outside the workspace, the EE cannot reach the target using the manipulator alone. In this case, the mobile base is commanded to move so that the EE can approach the desired position. Conversely, if the desired position lies within the manipulator workspace, the manipulator alone is sufficient to reach the target, and thus the mobile base does not need to move.

In addition to this fundamental concept, a blending function is introduced to enable smooth switching when the desired position crosses the workspace boundary. This prevents abrupt start–stop motions that would otherwise impose excessive mechanical loads on the system during instantaneous control authority transitions.

Although a similar switching concept was proposed in [16], the switching logic in that work was defined with respect to the workspace boundary, which required multiple

switching conditions across different regions. In contrast, the present study simplifies the structure by defining the switching condition with respect to a single representative point of the workspace, thereby providing a more concise and unified switching framework.

Since the workspace is defined as a sphere centered at the manipulator base mount, it is represented as shown in Figure 3.

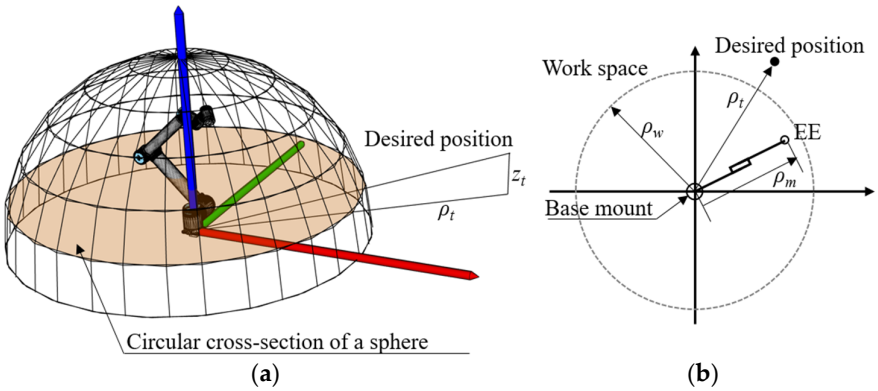


Figure 3. Workspace representation and planar cross-sectional model for control authority switching; (**a**) Three-dimensional spherical workspace centered at the manipulator base mount and its circular cross-section at the desired height; (**b**) Planar cross-sectional representation of the workspace and geometric relationship among the base mount, desired position, and EE.

Figure 3a illustrates the relationship between the workspace and the trajectory in three-dimensional space. Although the actual manipulator workspace is represented as a spherical region in 3D space, this study aims to reduce system dimensionality through motion decomposition. As confirmed in (1), the planar and vertical motions can be implemented independently. Based on this property, the workspace is defined in terms of a cross-sectional representation corresponding to the height of the desired position, as shown in Figure 3b. In Figure 3a,b, ρ_t denotes the planar distance between the desired position and the center of the workspace, while ρ_w represents the radius of the workspace cross-section corresponding to the height of the desired position. The value of ρ_w is computed as follows.

$$\rho_w = \sqrt{R^2 - z_t^2},\ |z_t| \leq R \tag{18}$$

Here, R denotes the radius of the spherical workspace. Once the planar workspace is defined according to (18), the switching function $\mu(\cdot)$ is formulated as illustrated in Figure 4.

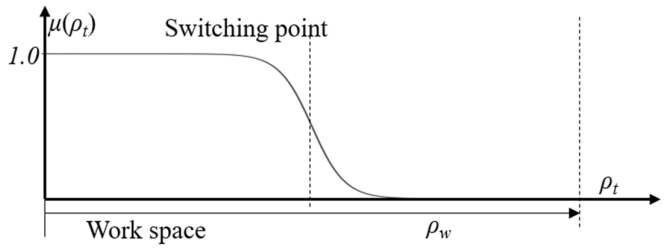


Figure 4. Radial-function-based control authority switching function with respect to ρ_t.

Figure 4 illustrates the switching function $\mu(\rho_t)$ with respect to ρ_t. Here, the switching point denotes the position at which the actual control authority transition occurs. When

the switching point is set closer to ρ_w, the operating range of the manipulator is expanded. Conversely, when it is set farther from ρ_w, the weighting of the mobile base increases.

$$\mu(\rho_t) = \frac{1}{2}\{1 - \tanh(\alpha(\rho_t - s))\}, \tag{19}$$

where, s denotes the switching point, and α is a parameter that determines the slope of the transient region. As α increases, the slope becomes steeper and the blending region becomes narrower; conversely, as α decreases, the blending region becomes wider. Both s and α are user-defined parameters that can be selected according to system requirements. However, s should be chosen with consideration of ρ_w. Based on Figure 4 and (2) and (19), the switching algorithm can be defined as follows.

$$v_\varepsilon = v_b \cos(\theta_m) + v_\rho \text{ where } v_b \cos(\theta_m) = (1 - \mu(\rho_t))v_{\varepsilon c},\ v_\rho = \mu(\rho_t)v_{\varepsilon c} \tag{20a}$$

$$\omega_\varepsilon = \omega_b + \omega_m \text{ where } \omega_b = (1 - \mu(\rho_t))\omega_{\varepsilon c},\ \omega_m = \mu(\rho_t)\omega_{\varepsilon c} \tag{20b}$$

In (20a), $v_b cos(\theta_m)$ is computed. Therefore, when solving for the actual v_b, $cos(\theta_m)$ appears in the denominator, which leads to a singularity at $\theta_m = \pm\pi/2$. To avoid this singular point, a radial function $\kappa(\theta_m)$, as presented in [15,17], is applied to modify v_b as follows.

$$v_b = \kappa(\theta_m)\frac{(1 - \mu(\rho_t))}{\cos(\theta_m)}v_\varepsilon^c, \tag{21}$$

where

$$\kappa(\theta_m) = 1 - \exp\left\{-\left(|\theta_m| - \frac{\pi}{2}\right)^2/\sigma^2\right\},\ \sigma > 0 \text{ is constant.} \tag{22}$$

The radial function $\kappa(\theta_m)$ ensures that the control input v_b avoids very large values or infinity when θ_m goes to $\pm\pi/2$ and that v_b is always defined.

Note: The switching point s determines the balance of control authority between the mobile base and the manipulator. When s is set close to ρ_w, the manipulator-only control region is maximized, but the EE is more likely to approach the workspace boundary, increasing the risk of joint velocity saturation and kinematic singularity. Conversely, when s is set close to zero, the mobile base is activated excessively, leading to unnecessary base motion and energy inefficiency. In this study, s is set to $\rho_w/2$, which equally distributes the control authority between the two subsystems: the manipulator primarily handles tracking when $\rho_t < s$, while the mobile base progressively participates as ρ_t exceeds s. Since ρ_w varies with the target height according to (18), s adapts automatically to the three-dimensional workspace geometry. Combined with α = 5.0, this configuration produces a smooth transition zone of approximately $\pm$20% of ρ_w, preventing both abrupt mechanical shock and excessive response delay. The user may increase s when the trajectory remains within the workspace or decrease it when early base activation is desired. For example, when s is set to a smaller value, the active range of the mobile base is expanded, causing the base to move even when the desired position lies within the reachable workspace of the manipulator. This configuration is advantageous in scenarios where minimizing manipulator joint motion is desired or where maintaining a compact arm posture during base transit is preferred. Conversely, when s is set to a larger value, the operational range of the manipulator is extended, allowing the robot to reach targets by extending its arm while the base is in motion. This is beneficial in confined environments or in situations where precise repositioning of the base incurs a high cost.

5. Simulation Results

This chapter includes the simulation results on the performance of the proposed EE-driven trajectory tracking control algorithm based on the linear and angular motion decomposition. For the numerical simulation, the mobile manipulator capable of moving in a three-dimensional space is modeled in a Python environment. Also, the technical summary and parameters for the simulation are provided in Table 2.

Table 2. Simulation setup and parameter configuration.

System parameters	SW		Python 3.10	
	Operation frequency		1 kHz	
	Numerical integration		1st-order Euler method	
	Inverse kinematics solution selection		Elbow down configuration	
	Link length (upper, lower)		1.5 m, 1.5 m	
Simulation parameters	k_x	6.0	Max ρ_w	3.0
	k_y	6.0	Blending factor α	5.0
	k_z	3.0	Blending factor s	$\rho_w/2$
	k_θ	4.5	σ	6.0
	v_b (min, max)	(−2.5, 2.5) (m/s)	ω_b (min, max)	(−2.5, 2.5) (rad/s)
	$d(\theta_{m1}, \theta_{m2})/dt$ (min, max)	(−2.5, 2.5) (rad/s)	ω_m (min, max)	(−2.5, 2.5) (rad/s)

The control gains and switching parameters listed in Table 2 are selected based on the following design rationale. As established in Theorem 1, the position tracking gains k_x, k_y, k_z and the orientation gain k_θ guarantee exponential error convergence for any positive values; the specific values determine the convergence rate according to the first order decay dynamics $\dot{e} = -k \cdot e$ given in (12). The horizontal gains $k_x = k_y = 6.0$ are set relatively high because both the mobile base and the manipulator contribute to the planar motion. The vertical gain $k_z = 3.0$ is set to a more conservative value, as the vertical motion is realized solely by the manipulator and must respect the joint velocity limits. The orientation gain $k_\theta = 4.5$ is set equal to the horizontal gains to ensure coordinated tracking. For the blending function, the switching point $s = \rho_w/2$ provides a balanced allocation between the manipulator-only and base-assisted control regions, and the slope coefficient $\alpha = 6.0$ is chosen to prevent both abrupt mechanical transitions and excessive response delays.

Note: The control gains in the proposed algorithm directly determine the convergence rate of the first-order differential equation governing the end-effector error dynamics. Since the system is analytically guaranteed to be stable regardless of gain magnitude, the gain selection reduces to a hardware-constrained design choice rather than an optimization problem. Specifically, larger gains yield faster convergence but proportionally increase the commanded velocity, which must remain within the physical joint velocity limits of the mobile base and manipulator. Therefore, the gains are selected as the maximum values that satisfy the actuator velocity constraints, ensuring the fastest achievable convergence without violating hardware limitations.

To show the performance of the proposed algorithm, we assume the following two trajectory scenarios as in Table 3.

To evaluate the performance and real-time feasibility of the proposed algorithm, it is compared with an optimization-based WBC approach employing a conventional QP controller [13]. It should be noted that both the proposed method and the QP-based controller are designed to achieve the same control objective, namely, to minimize the

EE position error with respect to the desired trajectory in three-dimensional space. Both methods operate under the same target trajectory, the same initial conditions, and the same system parameters as specified in Tables 2 and 3. While the proposed method constructs the control inputs directly through motion decomposition and explicit kinematic relationships as described in Sections 2–4, the QP-based controller computes the control inputs by solving an optimization problem over the holistic Jacobian under a weighting matrix.

Table 3. Simulation scenario details.

Initial state	(x_b, y_b, z_b)	$(-3.0, 3.0, 0.0)$ (m, m, m)	(x_t, y_t, z_t)	(2.0, 0.0, 0.25) (m, m, m)
	v_ε	0.0 (m/s)	$v_{\varepsilon z}$	0.0 (m/s)
	θ_b	0.0 (rad)	ω_ε	0.0 (rad/s)
	v_b	0.0 (m/s)	v_ρ	0.0 (m/s)
	ω_b	0.0 (rad/s)	v_z	0.0 (m/s)
	θ_ε	0.0 (rad)	ω_m	0.0 (rad/s)
	θ_m	0.0 (rad)	L_1, L_2	1.5, 1.5 (m)
	θ_{m1}, θ_{m2}	$\pi/4, \pi/2$ (rad)		
trajectory	v_{xt}	$1.2 \sin(t/1.5)$ (m/s)		
	v_{yt}	$-1.22 \cos(t/1.5)$ (m/s)		
	v_{zt}	$-0.25 \cos(t)$ (m/s)		

Note: The QP-based optimization algorithm is selected as the benchmark because it represents the current state of the art in whole-body control of mobile manipulators [6–11,17–21]. In particular, MPC—one of the most widely adopted advanced control strategies—frequently employs QP as its core optimization engine, and thus comparing against a QP-based controller is considered representative of the prevailing optimization-based control paradigm. To enhance the practical relevance of the comparison, velocity level constraints reflecting the physical limitations of the system are imposed on both the mobile base and the manipulator in the simulation.

This formulation ensures that the comparison between the two methods is conducted under a fair and consistent evaluation framework. The QP controller used as a benchmark is formulated with the following cost function.

$$\begin{gathered} Minimize\ C(\mathbf{q}) = \tfrac{1}{2}\dot{\mathbf{q}}^T\mathbf{H}\dot{\mathbf{q}} + \mathbf{f}^T\dot{\mathbf{q}}. \\ \text{Subject to:} \\ \mathbf{J}_\rho\dot{\mathbf{q}} \geq -k_{cbf}(\rho - \rho_{\min}), -\mathbf{J}_\rho\dot{\mathbf{q}} \geq -k_{cbf}(\rho_{\max} - \rho) \\ \mathbf{J}_\rho\dot{\mathbf{q}}t_s + (\rho - \rho_{\min}) \geq 0,\ \mathbf{J}_\rho\dot{\mathbf{q}}t_s + (\rho_{\max} - \rho) \geq 0 \\ |v_b| \leq v_{b_\max},\ |\omega_b| \leq \omega_{b_\max},\ \left|\dot{\theta}_{m1}\right| \leq \dot{\theta}_{m1_\max},\ \left|\dot{\theta}_{m2}\right| \leq \dot{\theta}_{m2_\max} \end{gathered},$$

where

$$\dot{\mathbf{q}} = \begin{bmatrix} v_b & \omega_b & \dot{\theta}_{m1} & \dot{\theta}_{m2} \end{bmatrix}^T,\ \mathbf{v}_{des} = \dot{\mathbf{x}}_t + \mathbf{K}(\mathbf{x}_t - \mathbf{x}_{ee})$$

$$\mathbf{J}_\rho = \begin{bmatrix} 0 & 0 & -L_1 \sin\theta_1 - L_2 \sin(\theta_{m1} + \theta_{m2}) & -L_2 \sin(\theta_{m1} + \theta_{m2}) \end{bmatrix}$$

The above optimization problem is equivalently expressed in the standard quadratic form, C(**q**), where

$$\mathbf{H} = \omega_{track}\mathbf{J}_h^T\mathbf{J}_h + \lambda_{reg}\mathbf{I} + \lambda_\varepsilon\mathbf{e}\mathbf{e}^T,\ \mathbf{f} = -\omega_{track}\mathbf{J}_h^T\mathbf{v}_{des} - \lambda_\varepsilon k_\varepsilon\theta_\varepsilon\mathbf{e}_1,\ \mathbf{e}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}^T$$

$$\omega_{track} = 1 + 5\|\mathbf{e}\|,\ \mathbf{x}_t = \begin{bmatrix} x_t & y_t & z_t \end{bmatrix}^T,\ \mathbf{x}_{ee} = \begin{bmatrix} x_\varepsilon & y_\varepsilon & z_\varepsilon \end{bmatrix}^T,\ \mathbf{K} = diag\begin{pmatrix} 10 & 10 & 8 \end{pmatrix},$$

$$\mathbf{J_h} = \begin{bmatrix} \cos\ \theta_b & -\rho \sin\ \theta_b & -(L_1 \sin\ \theta_{m1} + L_2 \sin\ (\theta_{m1} + \theta_{m2})) \cos\ \theta_b & -L_2 \sin\ (\theta_{m1} + \theta_{m2}) \cos\ \theta_b \\ \sin\ \theta_b & \rho \cos\ \theta_b & -(L_1 \sin\ \theta_{m1} + L_2 \sin\ (\theta_{m1} + \theta_{m2})) \sin\ \theta_b & -L_2 \sin\ (\theta_{m1} + \theta_{m2}) \sin\ \theta_b \\ 0 & 0 & L_1 \cos\ \theta_{m1} + L_2 \cos\ (\theta_{m1} + \theta_{m2}) & L_2 \cos\ (\theta_{m1} + \theta_{m2}) \end{bmatrix}$$

The QP problem is solved numerically at every control cycle using the SLSQP (Sequential Least Squares Programming) algorithm, with both the objective function and its analytical gradient supplied explicitly to eliminate numerical differentiation overhead. The solver is configured with a convergence tolerance is 1.0×10^{-8} and a maximum iteration limit of 500. The workspace feasibility is enforced through two layers of constraints that ensure the planar reach rho remains within $[\rho_{min}, \rho_{max}] = [0.30, 2.70]$ m, where $\rho_{max} = L_1 + L_2 - 0.3$ provides a boundary margin to prevent chattering near workspace limits, and one-step hard constraints that guarantee $\rho(t + t_s)$ within $[\rho_{min}, \rho_{max}]$ at every time step. All numerical computations, including the Jacobian evaluation, QP construction, and SLSQP optimization, are performed using the numerical routines of NumPy and SciPy in a Python 3.10 environment.

Under the environment specified in Table 2, when the scenario described in Table 3 is executed, the resulting motion and error behaviors are obtained as shown in Figure 5. As shown in Figure 5, the proposed algorithm exhibits rapid reduction of the EE position errors e_x, e_y, and e_z, even under large initial error conditions, compared with the QP-based tracking controller. All axes demonstrate stable convergence within approximately 2 s. In particular, stable convergence is maintained without divergence or oscillation during the initial tracking phase. The error in the z-direction also gradually converges to zero due to the independent proportional control term, and the desired trajectory is maintained without residual error throughout the entire simulation interval. These results indicate that the proposed algorithm guarantees stable error convergence even in a mobile manipulator system characterized by nonlinear kinematic coupling. In addition to the convergence performance shown in Figure 5, the two algorithms also differ in computational time. Figure 6 compares the computational time required by the two methods.

Figure 6 presents a comparison between the proposed algorithm and the QP-based control method in terms of computational time. From the perspective of computation, the difference between the two approaches is more pronounced. For the proposed algorithm, the average computation time per control loop is more than 13 times faster than that of the optimization-based method, and the variation in computation time throughout the entire simulation interval is very small. This improvement is attributed to the reduced control complexity achieved by the proposed framework. In contrast, the QP-based controller must solve an optimization problem at every control cycle, resulting in a larger average computation time and the potential for occasional computation-time spikes in worst-case scenarios. When the computation times shown in Figure 6 are analyzed along the time axis, the statistical results are summarized in Table 4.

Table 4. The computation time comparison.

	Mean Time	**Max Time**	**Min Time**
Proposed algorithm	0.044 (ms)	0.47 (ms)	0.039 (ms)
QP-based algorithm	0.576 (ms)	2.85 (ms)	0.23 (ms)

From the results in Figure 6 and Table 4, it can be concluded that the proposed algorithm is more suitable for mobile manipulator systems requiring real-time control. In particular, for applications demanding high-frequency control (e.g., above 1 kHz), the proposed method demonstrates clear computational efficiency advantages over the QP-based approach.

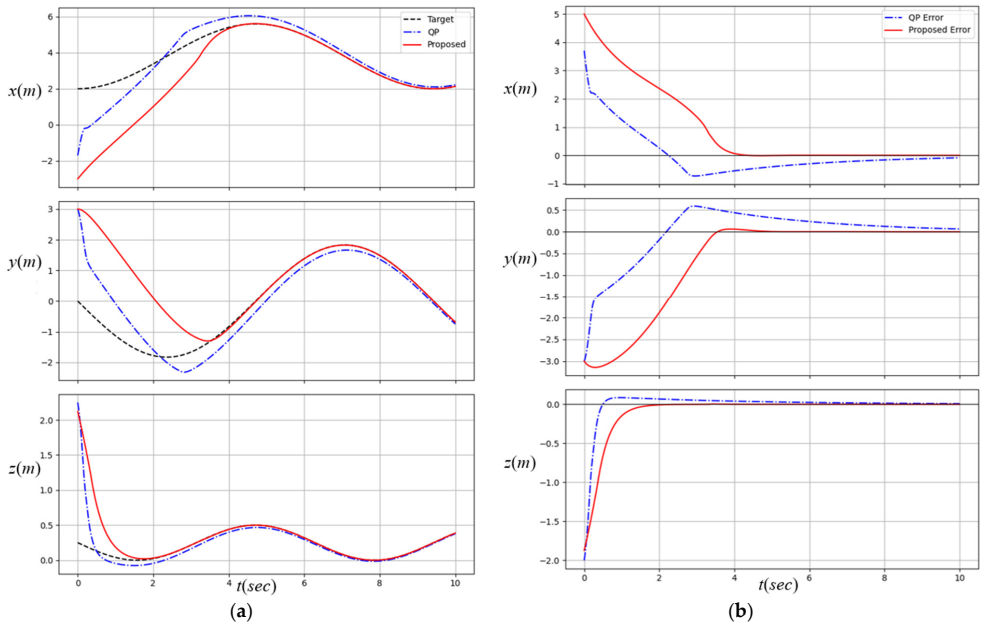


Figure 5. Three-dimensional trajectory tracking performance and error comparison between the proposed EE-driven WBTC and QP-based control methods (dash: target, dash-dot: QP, line: proposed): (**a**) Trajectory tracking routes in each axis; (**b**) Position errors in each axis.

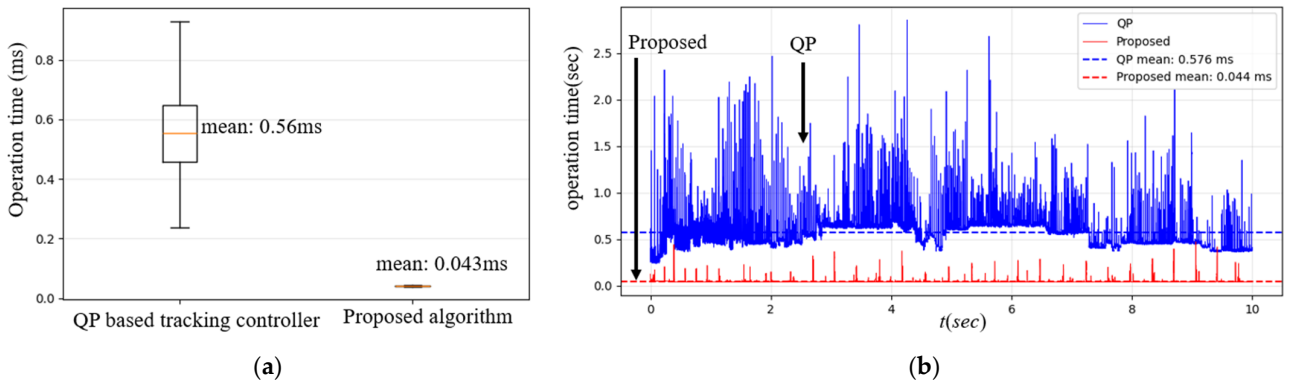


Figure 6. Comparison of computational time per control loop between the QP-based tracking controller and the proposed EE-driven algorithm: (**a**) Box plot of operation time distribution; (**b**) Time series of operation time over a 10-s interval.

6. Conclusions

This paper proposes the EE-driven WBTC algorithm based on linear and angular motion decomposition. The objective is to reduce the high computational complexity inherent in the WBC of high DOF WMM while ensuring stable and accurate 3D trajectory tracking. Conventional optimization-based approaches require solving constrained optimization problems at every control cycle, leading to high computational costs and time variability. In contrast, the proposed method formulates the control problem directly in the EE space, decomposing the required motion into planar linear, vertical, and angular components. This allows for the construction of control inputs through explicit kinematic relationships without relying on numerical optimization. Theoretical analysis confirms that this framework guarantees exponential convergence of both position and orientation errors. Redundancy between the mobile base and the manipulator is resolved through a control authority switching algorithm combined with a blending function, rather than weighted optimization. By defining the switching condition relative to a representative

point of the workspace—rather than its boundary—the overall structure is simplified, and abrupt control transitions are effectively prevented.

Numerical simulations validate the performance and contributions of the proposed algorithm. In 3D trajectory tracking scenarios, errors in all axes decreased rapidly without oscillation, demonstrating superior stability. Compared to QP-based controllers, the proposed method showed faster error reduction during the transient phase and maintained precise tracking without residual steady state errors. The distinction in computational efficiency is even more pronounced. The proposed algorithm reduced average computation time per control loop by more than 13 times compared to the QP-based approach, while maintaining minimal time variation. While QP-based controllers suffer from high computational burdens and potential time spikes, the proposed method's efficiency is particularly critical for practical applications requiring high-frequency control (above 1 kHz). By decomposing the total DOF of the WMM, this paper offers a new control paradigm that lowers control difficulty and confirms the fundamental feasibility of WBTC.

However, for real-world deployment where disturbances and uncertainties exist, further research is needed in the following areas. Regarding disturbance compensation, the proposed kinematic-based framework does not explicitly account for dynamic uncertainties such as payload variations, friction, and inertia changes; the integration of adaptive or online data-driven compensation techniques is required to maintain tracking performance under such conditions. Regarding fault detection, actuator faults including wheel slip and joint performance degradation may occur during prolonged operation, necessitating fault-tolerant control strategies that can detect and accommodate such faults without compromising stability. Regarding collision avoidance, the current framework does not incorporate obstacle avoidance capabilities; in cluttered or dynamic environments, the integration of real-time collision avoidance with the proposed control authority switching mechanism is essential to ensure safe operation.

In conclusion, the proposed EE-driven WBTC method significantly reduces computational complexity compared to previous approaches while maintaining stable and accurate tracking. These results demonstrate that the proposed method is a practical and viable framework with strong real time applicability for mobile manipulator systems.

Author Contributions: Conceptualization, J.-W.K.; methodology, J.-W.K.; software, T.U. and J.-H.P.; validation, T.U. and J.H.H.; formal analysis, T.U.; investigation, T.U.; resources, T.U.; data curation, J.L.; writing—original draft preparation, J.-W.K.; writing—review and editing, J.-W.K.; visualization, J.L.; supervision, J.-W.K.; project administration, T.U.; funding acquisition, T.U. All authors have read and agreed to the published version of the manuscript. Finally, all authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (2022-0-01010, Development of an intelligent manufacturing logistics system based on autonomous transport mobile manipulation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ghodsian, N.; Benfriha, K.; Olabi, A.; Gopinath, V.; Arnou, A. Mobile manipulators in industry 4.0: A review of developments for industrial applications. *Sensors* **2023**, *23*, 8026. [CrossRef] [PubMed]
2. Seraji, H. A unified approach to motion control of mobile manipulators. *Int. J. Robot. Res.* **1998**, *17*, 107–118. [CrossRef]
3. Outón, J.L.; Villaverde, I.; Herrero, H.; Esnaola, U.; Sierra, B. Innovative mobile manipulator solution for modern flexible manufacturing processes. *Sensors* **2019**, *19*, 5414. [CrossRef] [PubMed]
4. Zhang, J.; Shan, L.; Ma, E.; Chen, P.; Wang, W. Manipulability-guided MPC with repulsive potential fields for mobile manipulator whole-body control. In Proceedings of the 4th Conference on Fully Actuated System Theory and Applications, Nanjing, China, 4–6 July 2025.
5. Kindle, J.; Furrer, F.; Novkovic, T.; Chung, J.J.; Siegwart, R.; Nieto, J. Whole-body control of a mobile manipulator using end-to-end reinforcement learning. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020.
6. Yuan, W.; Liu, Y.-H.; Su, C.-Y.; Zhao, F. Whole-body control of an autonomous mobile manipulator using model predictive control and adaptive fuzzy technique. *IEEE Trans. Fuzzy Syst.* **2023**, *31*, 799–809. [CrossRef]
7. Wang, Y.; Chen, R.; Zhao, M. Whole-body model predictive control for mobile manipulation with task priority transition. In Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA), Atlanta, GA, USA, 19–23 May 2025.
8. Minniti, M.V.; Grandia, R.; Fäh, K.; Farshidian, F.; Hutter, M. Model Predictive Robot-Environment Interaction Control for Mobile Manipulation Tasks. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 31 May–4 June 2021.
9. Pankert, J.; Hutter, M. Perceptive model predictive control for continuous mobile manipulation. *IEEE Robot. Autom. Lett.* **2020**, *5*, 6177–6184. [CrossRef]
10. Misawa, K.; Xu, F.; Sekiguchi, K.; Nonaka, K. Model predictive control for mobile manipulators considering the mobility range and accuracy of each mechanism. *Artif. Life Robot.* **2022**, *27*, 855–866. [CrossRef]
11. Sleiman, J.-P.; Farshidian, F.; Minniti, M.V.; Hutter, M. A Unified MPC framework for whole-body dynamic locomotion and manipulation. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4688–4695. [CrossRef]
12. Hu, J.; Wang, F.; Li, X.; Qin, Y.; Guo, F.; Jiang, M. Trajectory tracking control for robotic manipulator based on soft actor–critic and generative adversarial imitation learning. *Biomimetics* **2024**, *9*, 779. [CrossRef] [PubMed]
13. Wang, C.; Zhang, Q.; Tian, Q.; Sun, Q.; Huang, X. Learning mobile manipulation through deep reinforcement learning. *Sensors* **2020**, *20*, 939. [CrossRef] [PubMed]
14. Breyer, M.; Furrer, F.; Novkovic, T.; Siegwart, R.; Nieto, J. Comparing task simplifications to learn closed-loop object picking using deep reinforcement learning. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1549–1556. [CrossRef]
15. Fu, Z.; Cheng, X.; Pathak, D. Deep Whole-Body Control: Learning a unified policy for manipulation and locomotion. In Proceedings of the Conference on Robot Learning (CoRL), Auckland, New Zealand, 14–18 December 2022.
16. Zhai, Z.M.; Moradi, M.; Kong, L.W.; Glaz, B.; Haile, M.; Lai, Y.-C. Model-Free Tracking Control of Complex Dynamical Trajectories with Machine Learning. *Nat. Commun.* **2023**, *14*, 5698. [CrossRef] [PubMed]
17. Li, M.; Yang, Z.; Zha, F.; Wang, X.; Chen, F. Design and analysis of a whole-body controller for a velocity controlled robot mobile manipulator. *Sci. China Inf. Sci.* **2020**, *63*, 170204. [CrossRef]
18. Kim, S.; Jang, K.; Park, S.; Lee, Y.; Lee, S.Y.; Park, J. Whole-body control of non-holonomic mobile manipulator based on hierarchical quadratic programming and continuous task transition. In *Proceedings of the IEEE International Conference on Advanced Robotics and Mechatronics (ARM), La Mirada, CA, USA, 3–5 July 2019*; IEEE Press: New York, NY, USA, 2019; pp. 414–419.
19. Töpfer, A.T.; Schmidt, M.J.; Müller, J.B.W.; Staudinger, J.; Maier, A.K. Improving data-based trajectory generation by quadratic programming for redundant mobile manipulators. In Proceedings of the 2024 10th International Conference on Automation, Robotics and Applications (ICARA), Dubai, United Arab Emirates, 15–19 July 2024.
20. Guevara, B.S.; Lopez-Lopez, V.E.; Lopez-Flores, R.E.; Flores-Godoy, M.A. Optimal control and dynamic compensation of a mobile manipulator robot. In Proceedings of the 2nd International Conference on Distributed Sensing and Intelligent Systems (ICDSIS), Puebla, Mexico, 30–31 July 2021.
21. Haviland, J.; Sünderhauf, N.; Corke, P. A Holistic approach to reactive mobile manipulation. *IEEE Robot. Autom. Lett.* **2022**, *7*, 3122–3129. [CrossRef]
22. Kwon, J.-W.; Chwa, D. Hierarchical formation control based on a vector field method for wheeled mobile robots. *IEEE Trans. Robot.* **2012**, *28*, 1335–1345. [CrossRef]

23. Chwa, D. Tracking control of differential-drive wheeled mobile robots using backstepping-like feedback linearization. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2010**, *40*, 1285–1295. [CrossRef]
24. Kwon, J.-W.; Park, J.-H.; Uhm, T.; Lee, J.; Lee, J.; Choi, Y.-H. Whole-body tele-operation for mobile manipulator based on linear and angular motion decomposition. *Appl. Sci.* **2026**, *16*, 712. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.