Humanoid Robot Teleoperation for Nonprehensile Transportation: A Multiple-Constraint Safety-Critical Control Framework

Xinyang Fan and Fenglei Ni *

Posted Date: 21 April 2026

doi: 10.20944/preprints202604.1405.v1

Keywords: humanoid robots; nonprehensile transportation; teleoperation; multiple constraints; safety-critical; hierarchical control

 doi:10.20944/preprints202604.1405.v1

Article

Humanoid Robot Teleoperation for Nonprehensile Transportation: A Multiple-Constraint Safety-Critical Control Framework

Xinyang Fan and Fenglei Ni *

State Key Laboratory of Robotics and Systems, Harbin Institute of Technology

* Correspondence: flni@hit.edu.cn; Tel.: +86 186 0451 3569

Abstract

This paper investigates the conflicting multiple constraints and safety challenges in humanoid robot teleoperation for nonprehensile transportation tasks. The robot's complex workspace and high degrees of freedom frequently conflict with highly dynamic task requirements, imposing stringent demands on coordinated motion. To address these issues, this paper proposes a Multiple-Constraint Safety-Critical Control Framework (MC-SCCF) featuring a hierarchical three-layer architecture. The top layer guarantees intrinsic safety against workspace boundaries using a continuously differentiable reachability surrogate model and an improved control barrier function (CBF)-based safe velocity filter for smooth deceleration. The middle layer maps user commands into pose-coupled reference trajectories to ensure task-level object safety, satisfying strict non-slip and non-toppling constraints. The bottom layer utilizes a quadratic programming (QP)-based inverse kinematics solver to achieve self-collision avoidance, coordinated motion, and optimal configuration while strictly enforcing joint and manipulability limits. Simulations and hardware experiments demonstrate that the MC-SCCF achieves real-time, high-precision reachability evaluation and successfully coordinates task dynamics with physical constraints, enhancing operational safety and the human-robot interaction experience.

Keywords: humanoid robots; nonprehensile transportation; teleoperation; multiple constraints; safety-critical; hierarchical control

1. Introduction

Humanoid robots, characterized by their anthropomorphic morphology and versatile motion capabilities, exhibit significant application potential in unstructured environments such as disaster rescue, domestic service, and industrial collaboration. Within these complex scenarios, object transportation is a fundamental and critical task. As an essential mode of operation, nonprehensile transportation demonstrates unique advantages in practical applications [1], making it particularly suitable for objects that are difficult to grasp directly. For instance, during the transfer of hazardous, fragile, or contaminated items, or when carrying food on a tray, nonprehensile transportation achieves object manipulation by strategically leveraging friction, gravity, and inertial forces. This approach not only reduces the internal forces exerted on the objects, thereby lowering the risk of damage, but also enables the simultaneous transportation of multiple objects, significantly improving operational efficiency.

However, such tasks impose stringent requirements on the dynamic coordination capabilities of the system. Given the current limitations in the decision-making capabilities of fully autonomous robots in complex dynamic environments, teleoperation, as a human-in-the-loop solution, integrates the cognitive adaptability of human operators with the physical manipulation capabilities of robots. Consequently, it has emerged as the prominent approach for executing such complex transportation tasks.

 doi:10.20944/preprints202604.1405.v1

Although teleoperation imparts human intelligence to the robot, executing nonprehensile transportation introduces challenges to the design of the robot's control system. On the one hand, it is imperative to ensure that the object does not slip or topple during motion, which imposes strict dynamic constraints on the pose coupling of the end-effector trajectory. On the other hand, to maintain the object's balance, the robot is frequently forced to operate near its workspace boundaries.

The evaluation of workspace boundaries for dual-arm coordinated tasks differs fundamentally from that of single-arm scenarios. Dual-arm systems entail coordinated motion requirements, resulting in more restrictive workspace limitations. In addition, because the two arms share a movable waist joint, their respective kinematic chains are highly coupled. Even if a specific pose is reachable within the independent workspace of each arm, there is no guarantee that a common waist configuration exists to allow both end-effectors to simultaneously reach their target poses.

When approaching these complex workspace boundaries, nonprehensile transportation tasks require the robot to both halt both safely and stably. While, during complex motions, the intrinsic kinematic constraints of the robot frequently conflict with the dynamic constraints of the task. The abrupt enforcement of traditional hard constraints results in severe acceleration discontinuities at the end-effectors, which directly disrupts the object's balance and causes task failure. While conventional CBF can initiate deceleration in advance, this approach tends to be overly conservative and fails to control the object's orientation during the deceleration phase.

To address the aforementioned challenges, this paper proposes a Multiple-Constraint Safety-Critical Control Framework (MC-SCCF). This research aims to construct a multilayered safety system based on a hierarchical design paradigm, integrating workspace boundaries, task-level object dynamic balance, and the robot's intrinsic kinematic constraints, as illustrated in Figure 1.

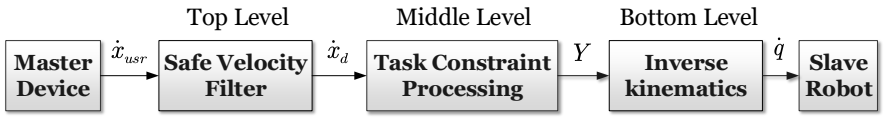


Figure 1. Block diagram of the proposed MC-SCCF.

The primary contributions of this paper are summarized as follows:

1. A MC-SCCF is proposed for humanoid robot nonprehensile transportation. This architecture decouples complex whole-body motion and task dynamics into three hierarchical levels. The top layer is responsible for the robot's intrinsic safety subject to workspace constraints. The middle layer guarantees the object's dynamic constraints to ensure task safety. The bottom layer enforces the remaining kinematic constraints and executes the task. It achieves the objectives of self-collision avoidance, coordinated motion, and configuration optimization while strictly satisfying joint and manipulability constraints. This framework systematically resolves the coordination challenges between task dynamic constraints and the kinematic constraints of the humanoid robot.
2. A top-level safe velocity modulation strategy is proposed to handle the workspace boundaries of the humanoid robot. To address the conflict between conservative deceleration and orientation jitter encountered when applying traditional CBFs, an improved CBF based on acceleration constraints is designed. This ensures that the robot can decelerate smoothly in an acceleration-controlled manner when approaching boundaries. Additionally, to tackle the difficulty of analytically describing the boundaries of the cooperative workspace, a reachability evaluation method based on a differentiable reachability surrogate model is designed.
3. The effectiveness of the proposed MC-SCCF is validated through experiments. Teleoperation experiments of multi-object nonprehensile transportation are designed and conducted on a physical humanoid robot platform. The experimental results demonstrate that this multilayered control framework can simultaneously guarantee the task dynamic constraints and the robot's intrinsic kinematic constraints, thereby enhancing operational safety and smoothness while improving the human-robot interaction experience.

2. Related Work

 doi:10.20944/preprints202604.1405.v1

2.1. Determination of Workspace Boundaries and Target Reachability Analysis

When humanoid robots execute nonprehensile transportation tasks, accurately evaluating the reachability of target poses and determining workspace boundaries are prerequisites for ensuring system safety and task success [2,3]. Traditional methods for determining workspace boundaries primarily include the Monte Carlo method [4], geometric envelope scanning [5], and numerical tracking methods based on Jacobian matrix singularity analysis [6,7]. These methods can macroscopically delineate the furthest theoretical physical boundaries reachable by the robot's end-effectors [8]. For dual-arm nonprehensile transportation tasks, merely obtaining a rough outer envelope surface is insufficient [9,10]. During practical teleoperation, the robot frequently needs to perform precise pose adjustments near the workspace boundaries. In such cases, accurately evaluating the reachability of specific target poses in real time becomes critical.

Reachability evaluation for dual-arm coordinated tasks is subject to relative pose constraints, which restrict the effective workspace. Furthermore, because the dual arms of a humanoid robot share a movable waist joint, the two kinematic chains exhibit highly coupled characteristics. Every movement of the waist simultaneously affects the workspaces of both arms [11,12]. Currently, there are three primary strategies for evaluating target pose reachability. The first category is optimization-based numerical inverse kinematics, such as whole-body control (WBC) methods [13–15]. Although this approach can handle complex dual-arm coordination and waist coupling constraints through nonlinear optimization, its solution process depends on the selection of initial values [16]. If used solely for preliminary reachability queries, this method is not only time-consuming but also prone to falling into local minima, leading to misjudgments [17–20]. The second category comprises algebraic or geometric analytical methods [21]. These methods exhibit fast computational speeds but typically require the robot to possess specific structures, such as a spherical wrist, to achieve kinematic decoupling. Therefore, they are difficult to generalize directly to the highly redundant upper-body systems of humanoid robots [22]. The third category is the currently widely applied data-driven reachability map method [23–27]. While this method achieves fast online querying by calculating and recording the reachability probability within Cartesian space voxels offline, extending it to dual-arm coupled configurations with a waist joint requires traversing every possible waist configuration for querying, which is computationally inefficient in practical applications [28,29].

In summary, when dealing with the highly coupled dual-arm systems of humanoid robots with a waist, existing reachability analysis methods frequently fall into a conflict between computational efficiency and evaluation accuracy, making it difficult to meet the demands of teleoperation tasks. Designing a real-time and accurate method for evaluating the reachability of cooperative dual-arm workspace boundaries remains a challenge in current teleoperation control frameworks.

2.2. Robot Intrinsic Constraint Processing Methods in Teleoperation

In humanoid robot teleoperation, ensuring the intrinsic safety of the robot (e.g., joint limits, self-collision avoidance, and singularity avoidance) is the foundation for stable system operation. Current solutions are primarily divided into two categories: feedback-based assistive guidance and optimization-based local autonomous control.

Feedback-based assistive guidance aims to enhance the operator's perception of the environment and constraints through multimodal information, thereby guiding the operator to make correct decisions. Regarding force feedback, *virtual fixtures* are widely applied to prevent the manipulator from colliding with obstacles and to avoid self-collisions by establishing repulsive force fields around obstacles and between robot links [30,31]. Metrics related to robot reachability maps and manipulability are utilized to establish attractive, repulsive, or damping forces. These forces guide the operator to actively avoid robot singular configurations and stay clear of dangerous areas such as workspace boundaries [32–35]. In addition, tactile feedback conveys the contact state with the interacting object through vibration or electrical stimulation, or it alerts the operator to dangerous situations using stimuli of varying intensities and frequencies as an alternative to force feedback [36,37]. Visual feedback provides the operator with a global perspective through the integration of digital model simulations, predicted trajectories, AR/VR systems, and multimodal information, intuitively presenting potential collision risks to the operator in a feedforward manner [38,39].

 doi:10.20944/preprints202604.1405.v1

Force and tactile feedback impose stringent requirements on hardware equipment, communication latency, and force algorithms. Inappropriate force guidance may interfere with the operator's normal tasks. More importantly, feedback-based assistance merely serves as a cue. Ultimate safety still depends on the operator's experience and reaction speed, making it difficult to provide deterministic safety guarantees during highly dynamic interactions.

In contrast, optimization-based local autonomous control has become the current mainstream trend. In this approach, algorithms directly manage constraint processing. Its core idea is to model the control problem as a constrained optimization problem, typically QP. Within this framework, joint limits, velocity boundaries, and obstacle avoidance requirements are transformed into mathematical inequality constraints, ensuring that the generated control commands lie strictly within the feasible region [40–42]. Recently, control barrier functions (CBFs) and control Lyapunov functions (CLFs), serving as effective tools for constructing safe sets and ensuring stability [43–45], are frequently integrated into optimization frameworks to enhance performance.

In teleoperation research focusing on nonprehensile transportation, some researchers have attempted to combine these two approaches. Selvaggio et al. [46] proposed a shared control architecture that introduces friction cone constraints into the QP to handle object dynamics while simultaneously generating virtual forces based on position deviations to guide the user. Muchacho et al. [47,48] processed task constraints via model predictive control (MPC) and indicated the robot's intrinsic motion limitations through force feedback. In our previous work [49,50], we proposed a shared teleoperation framework based on virtual object model predictive control (VO-MPC). This method resolves the algorithm's dependence on precise object parameters and enhances the system's robustness to unknown objects by transforming complex multi-object constraints into the boundary constraints of a single virtual offset object.

Nevertheless, the aforementioned studies are primarily based on single-manipulator systems, whose kinematic constraints are relatively simple. Directly migrating these methods to humanoid robot nonprehensile transportation tasks presents difficulties. The upper-body system of a humanoid robot possesses high kinematic redundancy, resulting in complex intrinsic kinematic constraints. Concurrently, nonprehensile transportation tasks involve strict dynamic constraints. A coupling relationship exists between the object's position and orientation trajectories, and the object's balance is highly sensitive to acceleration and pose. These two types of constraints frequently conflict during complex motions. Achieving compatibility among multiple hard constraints through weight allocation or soft constraint processing is difficult. Therefore, existing research has yet to resolve this challenge: how to construct a hierarchically decoupled control framework that achieves the unified integration of task execution and system safety while satisfying multiple physical constraints.

3. Safe Velocity Modulation Mechanism

Within the proposed MC-SCCF, the primary objective of the top layer is to transform the desired velocities input by the operator into safe velocity commands. To address workspace boundary constraints, this section formulates a safe velocity filter that is minimally invasive to the operator's intent. This filter utilizes an improved CBF as the optimization constraint. The improved CBF employs a piecewise function, enabling smooth, acceleration-controlled deceleration as the manipulator's end-effector approaches the workspace boundaries. Furthermore, to resolve the challenges of target reachability evaluation and workspace boundary distance calculation, a boundary evaluation method based on a differentiable reachability surrogate model is developed.

3.1. Safe Velocity Filter Design

Let $\dot{x}_{usr}$ denote the desired velocity command input by the operator. x_d and $\dot{x}_d$ represent the desired position and velocity of the transported object in the task space, respectively, and x denotes the actual position of the object. We formulate the following real-time QP problem to achieve safety filtering by solving for the optimal control input u^*, which represents the desired acceleration $\ddot{x}_d$:

 doi:10.20944/preprints202604.1405.v1

$$u^* = \arg\min_u \|\dot{x}_d - \dot{x}_{usr}\|^2,$$
$$\text{s.t.} \quad \dot{h}(x) \geq -\alpha(h(x)),$$
$$\dot{x}_d = f(x_d, u).$$

In this formulation, $h(x)$ represents the signed distance from the object to the workspace boundary along the direction of the current desired velocity. When the velocity direction is constant, $\dot{h}(x) = -\|\dot{x}\|$, where $\dot{x}$ is the actual velocity of the object. Within the proposed multilayer optimization framework, the middle and bottom layers ensure that x tracks x_d in real time. The term $f(x_d, u)$ denotes the model relating the end-effector velocity to the input. The core of this framework is the inequality constraint $\dot{h}(x) \geq -\alpha(h(x))$, which dictates the braking characteristics of the system as it approaches the boundaries.

3.2. Improved CBF Based on Acceleration Constraints

In handling workspace boundary constraints, CBFs provide an effective mechanism for ensuring forward invariance by modulating the system state derivative. For a given safe set $\mathcal{C} = \{x \in \mathbb{R}^n | h(x) \geq 0\}$, the function $\alpha(h(x))$ in conventional CBF design is typically chosen as a linear class $\mathcal{K}$ function, $\gamma h(x)$, where $\gamma > 0$ is a constant. When the barrier constraint is activated, the system velocity toward the boundary decays exponentially and asymptotically reaches zero at the boundary.

However, this linear class $\mathcal{K}$ function possesses limitations in nonprehensile transportation tasks. In such tasks, the object's orientation is coupled with its acceleration. When an object approaches the boundary at a certain velocity, the instantaneous activation of the CBF constraint can generate a large acceleration peak, with an acceleration norm of $\gamma|\dot{h}|$, leading to the object tilting and orientation jitter. Reducing γ to mitigate orientation jitter would cause the robot to restrict its velocity at a significant distance from the boundary, thereby compressing the effective workspace and reducing operational efficiency.

To address this, we design an improved CBF based on acceleration constraints, aimed at ensuring the controllability and boundedness of the object's orientation during deceleration. The improved class $\mathcal{K}$ function is defined as follows:

$$\dot{h}(x) \geq \begin{cases} -\sqrt{2a_{\max}(h(x) - 0.5h_t)}, & \text{if } h(x) \geq h_t, \\ -\gamma h(x), & \text{if } h(x) < h_t, \end{cases}$$

where $a_{\max}$ denotes the maximum allowable acceleration during the deceleration phase, and h_t is the switching threshold used to smoothly connect the piecewise function. To ensure constraint continuity and avoid step signals at the switching point h_t, the gain must satisfy the matching condition $\gamma = \sqrt{a_{\max}/h_t}$. This constraint maintains the fundamental structure of the CBF. As long as the initial state satisfies $h(x(0)) \geq 0$ and this constraint is strictly enforced during control solving, the state will remain within the safe set.

The improved class $\mathcal{K}$ function partitions the boundary approach process into a constant deceleration phase and an exponential convergence phase:

Constant Deceleration Phase ($h(x) \geq h_t$): When $\dot{h} \geq \sqrt{a_{\max} h_t}$, the system undergoes constant deceleration subject to the constraint of $a_{\max}$. The deceleration distance is $\frac{\dot{h}^2}{2a_{\max}}$.

Exponential Convergence Phase ($h(x) < h_t$): When $\dot{h} < \sqrt{a_{\max} h_t}$, the constraint switches to the linear CBF form, and the system velocity decays exponentially. The deceleration distance is $\frac{\dot{h}^2}{a_{\max}}$.

Assuming a constant velocity direction for the object, when $\|\dot{x}\| \geq \sqrt{a_{\max} h_t}$, the object undergoes two stages of deceleration, and the total deceleration distance is:

 doi:10.20944/preprints202604.1405.v1

$$D_{dec} = \frac{\|\dot{x}\|^2}{2a_{\max}} + \frac{1}{2}h_t. \quad (1)$$

When $\|\dot{x}\| < \sqrt{a_{\max} h_t}$, the deceleration process remains in the exponential convergence phase, and the deceleration distance is:

$$D_{dec} = \frac{\|\dot{x}\|^2}{a_{\max}}. \quad (2)$$

By introducing a maximum acceleration constraint, the improved CBF ensures stable forces on the object, with controlled acceleration and pose during deceleration. As the robot nears the boundary, the switch to the linear CBF ensures smoother deceleration. This leverages the asymptotic convergence of the exponential function to eliminate chattering phenomena as the velocity approaches zero, ensuring the robot stops smoothly at the boundary.

3.3. Workspace Boundary Determination

To achieve velocity modulation, the system must calculate the distance $h(x)$ from the current position to the workspace boundary in real time. Under the constraint of the CBF, the object will stop at the workspace boundary. Therefore, $h(x)$ is defined as the Euclidean distance between the current position and the furthest reachable position in the current velocity direction with a static orientation.

During dual-arm cooperative tasks, workspace reachability can be defined as whether there exists a waist configuration q^*_{waist} that makes the target poses of both end-effectors simultaneously reachable. Querying by traversing all waist configurations cannot satisfy real-time requirements. Hence, we design a target reachability evaluation method utilizing a differentiable reachability surrogate model. The reachability of target points is iteratively determined using the bisection method to achieve a fast and accurate approximation of the workspace boundary.

3.3.1. Reachability Determination

In dual-arm cooperative tasks, the reachability of target poses is restricted not only by kinematic coupling but also by the robot's self-collision constraints. Therefore, prior to conducting the kinematic reachability search, the shortest distance between the bounding volumes of the dual-arm end-effectors and the bounding volume of the robot base at the target pose is calculated. If a collision occurs, the target is directly determined as unreachable. When no collision exists, reachability determination at the kinematic level is performed.

Considering the kinematic independence of the two arms relative to their respective shoulder bases, we reduce the dimensionality of the problem. First, the true discrete reachability maps of both arms relative to the shoulder coordinate systems are established, and a continuously differentiable reachability surrogate model is trained accordingly. Subsequently, utilizing the gradient information of the surrogate model, the optimal waist configuration is determined to maximize the reachability of the dual-arm target positions. Finally, according to the optimal configuration, the reachability of the target pose is determined within the true reachability map. The reachability determination process is summarized in Algorithm 1.

The discrete reachability map divides the space into 3D voxels with an edge length of d_r, where the centroid of each voxel represents its position. The axis-angle representation is employed to define orientations. Specifically, n_a unit rotational axes uniformly distributed in space and n_c rotational angles uniformly distributed within $[0, \pi]$ are specified. These axes and angles jointly constitute $n_a n_c$ orientations. The inverse kinematics for all orientations are calculated sequentially at the position of each voxel. All poses are encoded, and the existence of their inverse kinematic solutions is recorded.

The reachability of position P is defined as:

$$R(P) = \frac{r(P)}{n_a n_c} \times 100\% \in [0, 1],$$

where $r(P)$ denotes the number of existing inverse kinematic solutions at this position. Let P_l and P_r denote the positions of the left arm end-effector relative to the left shoulder base and the right arm end-effector relative to the right shoulder base, respectively. $R_l(P_l)$ and $R_r(P_r)$ represent the

 doi:10.20944/preprints202604.1405.v1

reachability maps of the left and right arms, respectively. The reachability maps of the dual arms are illustrated in Figure 2.

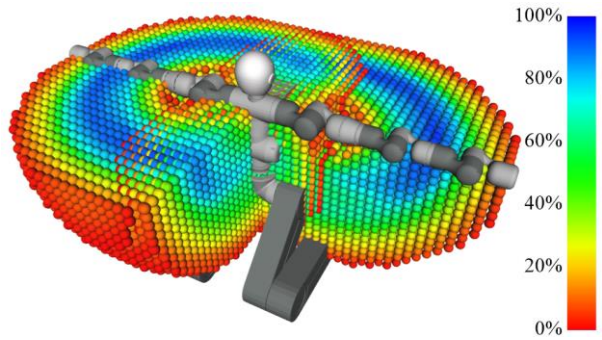
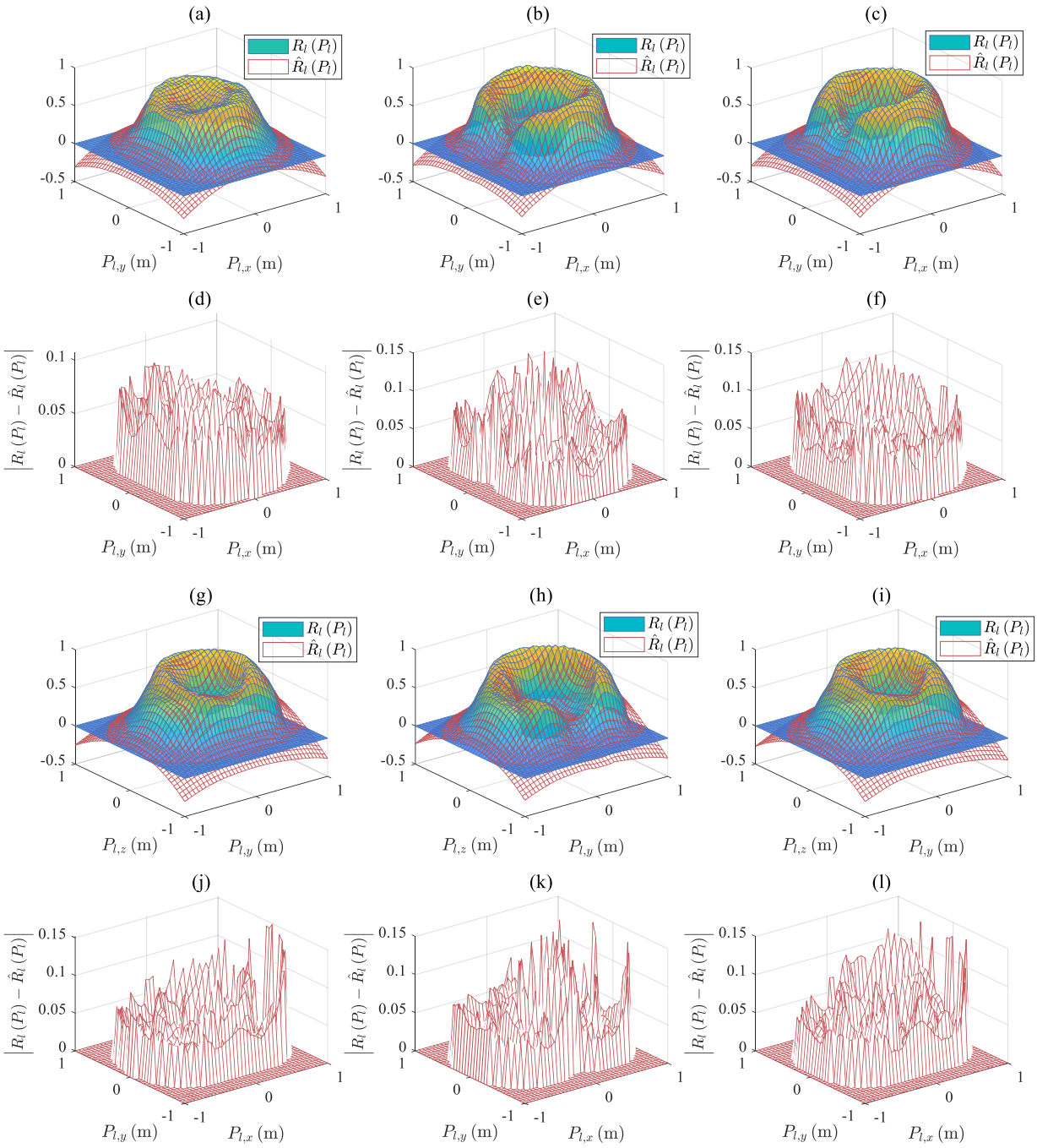


Figure 2. Dual-arm reachability maps of the custom-built humanoid robot platform.



 doi:10.20944/preprints202604.1405.v1

Figure 3. Fitting results of $\hat{R}_l(P_l)$ to $R_l(P_l)$. (a)-(c) depict the fitting results for the cross-sections at $P_{l,z}=-0.3,\ 0,\ 0.3$, respectively. (d)-(f) represent the fitting errors for the cross-sections at $P_{l,z}=-0.3,\ 0,\ 0.3$, respectively. (g)-(i) illustrate the fitting results for the cross-sections at $P_{l,x}=-0.3,\ 0,\ 0.3$, respectively. (j)-(l) denote the fitting errors for the cross-sections at $P_{l,x}=-0.3,\ 0,\ 0.3$, respectively.

A three-layer fully connected neural network (FCNN) is employed to train the reachability surrogate model, learning the nonlinear mapping $\hat{R}_i(P_i)$ from the end-effector position P_i to the reachability index $R_i(P_i)$, where $i\in\{l,r\}$. The network achieves a high-fidelity fit to the discrete reachability data (test set $R^2=97.1\%$) and constructs a continuously differentiable reachability manifold. The fitting results are presented in Figure 3. In the illustrated cross-sections, the fitting error is less than 0.15.

Utilizing this continuously differentiable reachability surrogate model, the gradient of reachability with respect to position, denoted as $\nabla_{p_i}\hat{R}_i(P_i)$, can be calculated. This gradient indicates the direction of the steepest increase in reachability. Consequently, the search for the optimal waist configuration q^*_{waist} is modeled as a gradient-driven iterative optimization process.

For a given waist configuration, the reachability gradient $\nabla\hat{R}_i(P_i)$ at the target position P_i can be regarded as a virtual attractive force pulling the position toward the direction of increasing reachability. The reaction force of this virtual attractive force, $-\nabla\hat{R}_i(P_i)$, acts on the shoulder base. If the waist joint is set to damping mode, it will move under the influence of $-\nabla\hat{R}_i(P_i)$ toward a configuration that increases the reachability of the target position. The iterative formula for the waist joint angle is defined as:

$$\Delta q_{waist}=-\frac{1}{\beta}\left(J^{\mathrm{T}}_{waist,l}\nabla\hat{R}_l(P_l)+J^{\mathrm{T}}_{waist,r}\nabla\hat{R}_r(P_r)\right).$$

Where β is the damping coefficient, and Δq_{waist} represents the update step size. $J_{waist,i}$ denotes the Jacobian matrix of the waist joint, with $i\in\{l,r\}$.

The iteration terminates when $\|\Delta q_{waist}\|<\varepsilon$ or the maximum number of iterations, $\mathrm{max\_iters}$, is reached. Utilizing q^*_{waist}, the system calculates the exact poses of the dual-arm target poses within their respective shoulder coordinate systems and performs a nearest neighbor query within the true discrete reachability maps. If inverse kinematic solutions exist for the nearest neighbors of both target poses, the target is determined as reachable; otherwise, it is determined as unreachable.

Algorithm 1: Reachability determination of target pose

Input: ${}^sP_l, {}^sP_r$: Target positions of left and right end-effectors in world frame;

${}^s\boldsymbol{\Phi}_l, {}^s\boldsymbol{\Phi}_r$: Target orientations of left and right end-effectors in world frame;

ε: Convergence threshold.

Output: Reachability of target poses.

1: **If** $\mathrm{ComputeMinDistance}({}^sP_l,\ {}^sP_r,\ \mathcal{V}_{base})<0$ **Then**

2: **Return** false

3: **End If**

4: $q_{waist}\leftarrow\mathbf{0}$

5: **for** i=1: max_iters **do**

6: // Step 1: Compute EE positions relative to shoulder frames

7: $P_l\leftarrow\mathrm{Kinematics}(q_{waist},\ {}^sP_l)\ \ P_r\leftarrow\mathrm{Kinematics}(q_{waist},\ {}^sP_r)$

8: // Step 2: Compute Virtual Forces and Map to Waist Joints

9: $\tau_{waist}\leftarrow -J^{\mathrm{T}}_{waist,l}\nabla\hat{R}_l(P_l)-J^{\mathrm{T}}_{waist,r}\nabla\hat{R}_r(P_r)$

10: // Step 3: Enforce joint limits and Update Waist Angle

 doi:10.20944/preprints202604.1405.v1

11: $\Delta q_{waist} \leftarrow \max\left(\min\left(\tau_{waist}/\beta,\ \bar{q}_{waist} - q_{waist}\right),\ \underline{q}_{waist} - q_{waist}\right)$
12: $q_{waist} \leftarrow q_{waist} + \Delta q_{waist}$
13: // Step 4: Check Convergence
14: **If** $\|\Delta q_{waist}\| < \varepsilon$ **Then**
15: $q^*_{waist} \leftarrow q_{waist}$
16: **Break for**
17: **End If**
18: **End for**
19: // Step 5: Nearest Neighbor Query
20: $(P_l,\ \Phi_l) \leftarrow \text{Kinematics}(q^*_{waist}, {}^sP_l, {}^s\Phi_l)$, $(P_r,\ \Phi_r) \leftarrow \text{Kinematics}(q^*_{waist}, {}^sP_r, {}^s\Phi_r)$
21: $(\hat{P}_l,\ \hat{\Phi}_l) \leftarrow \text{NearestNeighbor}(\text{Map}_l, P_l, \Phi_l)$, $(\hat{P}_r, \hat{\Phi}_r) \leftarrow \text{NearestNeighbor}(\text{Map}_r, P_r, \Phi_r)$
22: $\text{isReachable}_l \leftarrow \text{CheckIK}(\text{Map}_l,\ \hat{P}_l,\ \hat{\Phi}_l)$, $\text{isReachable}_r \leftarrow \text{CheckIK}(\text{Map}_r, \hat{P}_r,\ \hat{\Phi}_r)$
23: // Step 6: Reachability Decision
24: **If** isReachable_l **AND** isReachable_r **Then**
25: **Return** true
26: **Else**
27: **Return** false
28: **End If**

3.3.2. Boundary Distance Calculation

The purpose of calculating the distance $h(x)$ from the current position to the workspace boundary is to determine whether the CBF constraint needs to be applied to the current motion. Therefore, precise boundary searching is not required at all times. We improve computational efficiency through predictive pruning.

1. Predictive Stopping Pose Detection (Pruning Step): First, the system calculates the Predicted Stopping Pose (PSP) based on the current motion state. The PSP is defined as the pose where the object would come to a complete stop along the current velocity direction if the CBF constraint were triggered at the current moment. According to (1) and (2), the position of the PSP is calculated as $x_{\text{PSP}} = x + D_{dec}\frac{\dot{x}}{\|\dot{x}\|}$. The orientation of the PSP is defined as the static orientation of the object. Then, the reachability of the PSP is verified using the method described in Section 3.3.1. If the PSP is determined to be reachable, it indicates that the robot can safely stop within the workspace given its current velocity and braking capability. In this case, the CBF constraint will not be activated, and the calculation of the precise boundary distance $h(x)$ is unnecessary.
2. Bisection-Based Boundary Approximation (Bisection Search): If the PSP is determined to be unreachable, it implies a risk of boundary violation, and the system initiates a precise search. A bisection search is performed along the ray connecting the current position and the PSP to iteratively sample and determine the reachability of each point. The iteration terminates when the length of the sampling interval is smaller than the spatial resolution d_r of the offline reachability map.

4. Dynamic Constraint Processing Method Based on a Virtual Object

Within the middle layer of the MC-SCCF, the primary objective is to ensure that the transported object (or multiple objects on a tray) does not slip or topple when the robot executes nonprehensile transportation tasks. Nonprehensile transportation relies on contact forces and inertial forces to maintain stability. We adopt the dynamic constraint processing algorithm based on a virtual object

 doi:10.20944/preprints202604.1405.v1

proposed in our previous work [49,50]. The core idea of this algorithm is to transform complex multi-object constraints into the constraints of a single virtual offset object, combined with MPC to achieve online trajectory smoothing.

4.1. Construction of virtual objects

In nonprehensile transportation systems, the risk of object instability (slipping or toppling) is highly correlated with its relative position on the tray, geometric dimensions, and the friction coefficient of the contact surface. To achieve constraint dimensionality reduction, we extract the limit parameters from multiple real objects on the tray: the minimum base edge length $d=\min\{d_i\}$, the maximum height $h=\max\{h_i\}$, and the minimum friction coefficient $\mu=\min\{\mu_i\}$. Two special virtual objects are constructed:

- Virtual reference object O (for trajectory planning): This object is placed at the geometric center of the tray. The controller refers to object O for trajectory planning and ensures C^4 continuity of the position. Simultaneously, the center of mass (CoM) coordinate system of O is utilized as the center of rotation of the tray to calculate the orientation. Reference object O and offset object A share the same shape and mass.
- Virtual offset object A (for dynamic constraint analysis): The shape of offset object A is a rectangular prism with a square base, featuring a base edge length of d, a height of h, and a friction coefficient of μ. Its mass is concentrated at its eight vertices, with a total mass of 1 kg. The maximum moment of inertia of the offset object is defined as $J_{\max}=\frac{1}{4}(d^2+h^2)$. The specific position of offset object A is arbitrary, provided that the distance between A and O is maximized. The distance between the offset object and the reference object is expressed as $\|P_A\|=\sqrt{R_{tray}^2+(0.5h)^2}$, where R_{tray} denotes the maximum radius of the actual tray. Although the offset object is not necessarily located on the physical tray, it can be assumed that the tray is sufficiently large, allowing the offset object to be utilized for constraint calculation.

This configuration renders the offset object more susceptible to slipping and toppling than any real object on the tray. Under any motion state, as long as the offset object remains stable, all real objects on the tray are guaranteed to maintain stability. Therefore, the dynamic constraints of multiple objects are transformed into the constraints of a single offset object.

4.2. Dynamic Constraint Analysis Based on a Virtual Object

In nonprehensile transportation, the system counteracts tangential contact forces by adjusting the tray's orientation. Thus, the object's orientation and linear acceleration are highly coupled. During motion, the offset object and the reference object share the identical orientation, angular velocity, and angular acceleration (i.e., $\Omega_A=\Omega_o$ and $\dot{\Omega}_A=\dot{\Omega}_o$); nevertheless, their linear accelerations differ. To ensure safety, offset object A must simultaneously satisfy the friction cone constraint and the zero moment point (ZMP) constraint. By neglecting high-order minor coupling terms, these two physical constraints can be directly transformed into a restriction on the angular acceleration $\|\dot{\Omega}_o\|$ of reference object O:

$$\|\dot{\Omega}_o\|\le\min\left\{\frac{\|\ddot{x}_o-g\|}{\|P_A\|}\sin\theta_0,\ \frac{\|\ddot{x}_o-g\|d}{2J_{\max}+\|P_A\|\sqrt{d^2+h^2}}\right\},\tag{3}$$

where, $\ddot{x}_o$ indicates the linear acceleration of the reference object, g represents the gravitational acceleration, and $\theta_0=\arctan\mu$.

4.3. Trajectory Smoothing Based on MPC

Input trajectories generated by human operators during teleoperation typically lack high-order continuity. To generate smooth trajectories that strictly satisfy the aforementioned angular acceleration boundaries, we formulated an MPC position controller structured with a cascade of five integrators. The state vector is defined as $\boldsymbol{X_o}=[x_o^{\mathrm{T}},\ \dot{x}_o^{\mathrm{T}},\ \ddot{x}_o^{\mathrm{T}},\ \dddot{x}_o^{\mathrm{T}},\ \ddddot{x}_o^{\mathrm{T}}]^{\mathrm{T}}$. Here, $x_o\in\mathbb{R}^3$ denotes the desired position of the reference object, and $\boldsymbol{u}\in\mathbb{R}^3$ represents the fifth derivative of x_o. The control

 doi:10.20944/preprints202604.1405.v1

objective is to calculate the optimal system input $\boldsymbol{u}$ such that $\mathcal{T}_0$ tracks $x_{d'}$ which is output by the top-layer controller detailed in Section 3.

To eliminate the tangential forces exerted on the object, the desired orientation of the tray, denoted by $\Phi = \alpha\hat{n}$, is entirely determined by the acceleration $\ddot{x}_o$ of the reference object. Its rotational axis $\hat{n}$ and rotational angle α are defined as follows:

$$\hat{n} = \frac{\ddot{x}_o \times g}{\|\ddot{x}_o \times g\|},\ \alpha = \arccos\left(\frac{g^{\mathrm{T}}(g - \ddot{x}_o)}{\|g\|\,\|g - \ddot{x}_o\|}\right),$$

Successive differentiation of the orientation reveals that the angular acceleration $\dot{\Omega}_o$ of the reference object is dictated by $\dddot{x}_o$. The nonlinear angular acceleration limit $\left\|\dot{\Omega}_o\right\|_{\max}$ in (3) can be mapped into real-time upper and lower bounds for the fourth derivative of the position within the MPC state vector:

$$\|\dddot{x}_o\| \leq \begin{cases} \left\|\dot{\Omega}_o\right\|_{\max} \dfrac{\sqrt{\ddot{x}_{o,1}^2 + \ddot{x}_{o,2}^2}}{\alpha}, & \alpha \neq 0, \\ \left\|\dot{\Omega}_o\right\|_{\max} |g_3 - \ddot{x}_{o,3}|, & \alpha = 0, \end{cases} \tag{4}$$

where $\ddot{x}_o = [\ddot{x}_{o,1} \quad \ddot{x}_{o,2} \quad \ddot{x}_{o,3}]^{\mathrm{T}}$ and $g = [0 \quad 0 \quad g_3]^{\mathrm{T}}$.

The MPC optimization problem is formulated as follows:

$$\begin{gathered} \min_{\boldsymbol{u}} \sum_{k=0}^{N-1} \left(\|\boldsymbol{X_d}(k+1) - \boldsymbol{X_o}(k+1)\|_{W_x}^2 + \|\boldsymbol{u}(k)\|_{W_u}^2 \right), \\ \text{s.t. :}\ \ \boldsymbol{X_o}(k+1) = \boldsymbol{A_d}\boldsymbol{X_o}(k) + \boldsymbol{B_d}\boldsymbol{u}(k), \\ (4),\ \bar{u} \leq \boldsymbol{u}(k) \leq \underline{u}. \end{gathered}$$

In this formulation, N denotes the prediction horizon of the controller; $\boldsymbol{X_d}$ represents the desired state output by the top-layer controller, defined as $\boldsymbol{X_d} = [x_d^{\mathrm{T}}, \dot{x}_d^{\mathrm{T}}, \ddot{x}_d^{\mathrm{T}}]^{\mathrm{T}}$. The equation $\boldsymbol{X_o}(k+1) = \boldsymbol{A_d}\boldsymbol{X_o}(k) + \boldsymbol{B_d}\boldsymbol{u}(k)$ represents the discretized state-space model. The notation $\|\cdot\|_W^2 = (\cdot)^{\mathrm{T}} W(\cdot)$ is employed, where W_x and W_u indicate the weight coefficient matrices for the output and input, respectively.

5. Constraints and Optimization Objectives of Inverse Kinematics

This section maps task-space trajectories to the robot's joint space utilizing a QP-based low-level controller. This approach ensures that the robot strictly satisfies joint limits and manipulability constraints while achieving self-collision avoidance, configuration optimization, and coordinated motion.

Let $q \in \mathbb{R}^n$ represent the generalized coordinates of the humanoid robot's upper-body system, where n indicates the degree of freedom (DoF) of the robot, encompassing the waist joint and the joints of the left and right arms. For each end-effector $i \in \{l, r\}$, the relationship between its task-space velocity $\dot{\chi}_i$ and the joint velocity $\dot{q}$ is expressed as follows:

$$\dot{\chi}_i = J_i(q)\dot{q}.$$

Where $\dot{\chi}_i \in \mathbb{R}^6$, and $J_i(q) \in \mathbb{R}^{6\times n}$ denotes the Jacobian matrix.

5.1. Task Constraints

5.1.1. Trajectory Tracking

In nonprehensile transportation tasks, stability is independent of the virtual reference object's orientation about its Z-axis, which indicates the normal direction of the contact surface. Therefore, we relax the strict tracking requirement for the Z-axis orientation, constraining only the remaining five DoFs. The objective of orientation tracking shifts to aligning the actual Z-axis (z_O) of the reference object's coordinate frame with the desired Z-axis (z_{O_d}).

As demonstrated in Figure 4, $\{S\}$ denotes the world frame, $\{O\}$ represents the coordinate frame of the virtual reference object, and z_O indicates the Z-axis of the object's coordinate frame.

 doi:10.20944/preprints202604.1405.v1

Similarly, z_{O_d} denotes the Z-axis of the desired coordinate frame. A coordinate frame $\{O'\}$ is introduced, which is rigidly attached to $\{O\}$, such that $z_{O'} = z_O$, $y_{O'} = \frac{z_O \times z_{O_d}}{\| z_O \times z_{O_d} \|}$, and $x_{O'} = y_{O'} \times z_{O'}$.

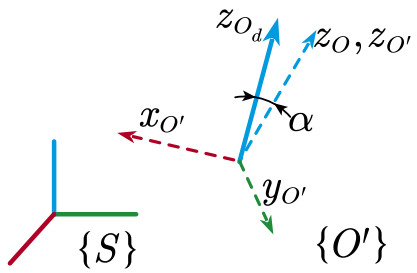


Figure 4. Orientation tracking of the virtual reference object.

The orientation of $\{O'\}$ is represented by the rotation matrix $R_{O'} = [x_{O'}, y_{O'}, z_{O'}]$. Using XYZ Euler angles, it is expressed as $\Phi_{O',XYZ} = [\Phi_x, \Phi_y, \Phi_z]^{\mathrm{T}}$. The orientation deviation Euler angles within $\{O'\}$ are calculated as $\Delta\Phi_{XYZ} = [0, \alpha, C]^{\mathrm{T}}$, where α indicates the angle between z_O and z_{O_d}, and C represents an arbitrary value. The tracking constraint transformed into joint velocities is formulated as:

$$J_{i,O}(q)\dot{q} = \lambda_O R_{O'} E_{XYZ}(\Phi_{O',XYZ}) \Delta\Phi_{XYZ}, \tag{5}$$

where $J_{i,O}(q) \in \mathbb{R}^{3\times n}$ indicates the orientation component of $J_i(q)$. $E_{XYZ} \in \mathbb{R}^{3\times 3}$ represents the mapping between the XYZ Euler angle rates and the angular velocity vector, defined as:

$$E_{XYZ} = \begin{bmatrix} 1 & 0 & \sin\Phi_y \\ 0 & \cos\Phi_x & -\sin\Phi_x \cos\Phi_y \\ 0 & \sin\Phi_x & \cos\Phi_x \cos\Phi_y \end{bmatrix}.$$

The tracking constraint for the object position is defined as:

$$J_{i,P}(q)\dot{q} = \lambda_P \Delta P, \tag{6}$$

where $J_{i,P}(q) \in \mathbb{R}^{3\times n}$ denotes the position component of $J_i(q)$, and ΔP indicates the position deviation of the object.

5.1.2. Constant Relative Pose of Dual-Arm End-Effectors

Nonprehensile transportation tasks require that the dual-arm end-effectors maintain no relative motion; otherwise, the tray might fall, or the object might slip and lose contact. As a result, the relative pose between the end-effectors must remain constant. Transforming this relative pose constraint into joint velocity constraints yields:

$$\frac{\partial \chi_{rel}(q)}{\partial q}\dot{q} = J_{rel}(q)\dot{q} = -\lambda_{rel}(\chi_{rel}(q) - \chi_{rel}(q_0)). \tag{7}$$

In this formulation, $\chi_{rel}(q)$ denotes the relative pose of the end-effectors at joint angle q, while q_0 indicates the initial joint configuration. $J_{rel}(q)$ represents the relative Jacobian matrix, defined as $J_{rel}(q) = \Psi(q)J(q)$, where $J(q)$ indicates the composite Jacobian matrix $J(q) = [J_l^{\mathrm{T}}(q) \quad J_r^{\mathrm{T}}(q)]^{\mathrm{T}}$. Following the methodology in [51], $\Psi(q)$ is expressed as:

$$\Psi(q) = \begin{cases} \begin{bmatrix} -R_l^{\mathrm{T}} & R_l^{\mathrm{T}} S({}^{l}P_r) & R_l^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & -R_l^{\mathrm{T}} & \mathbf{0} & R_l^{\mathrm{T}} \end{bmatrix}, & L \text{ relative to } R, \\ \begin{bmatrix} R_r^{\mathrm{T}} & \mathbf{0} & -R_r^{\mathrm{T}} & R_r^{\mathrm{T}} S({}^{r}P_l) \\ \mathbf{0} & R_r^{\mathrm{T}} & \mathbf{0} & -R_r^{\mathrm{T}} \end{bmatrix}, & R \text{ relative to } L, \end{cases}$$

where R_i $(i \in \{l, r\})$ indicates the absolute orientation of the end-effectors, ${}^{l}P_r$ represents the position of the right end-effector relative to the left, and $S({}^{l}P_r)$ denotes the skew-symmetric matrix of ${}^{l}P_r$.

5.1.3. Singularity Avoidance

 doi:10.20944/preprints202604.1405.v1

To maintain adequate kinematic dexterity, the robot must continuously avoid singular configurations. Considering that formulating manipulability directly as an optimization objective might conflict with other objectives, we opt to constrain its lower bound. This approach guarantees system flexibility while releasing substantial optimization space. Utilizing a CBF, this requirement is modeled as an inequality constraint to ensure that the manipulability measure $M(q)$ consistently remains above a safety threshold $M_{\min}$:

$$\dot{M}(q) \geq -\lambda_M (M(q) - M_{\min}). \tag{8}$$

Where, $\dot{M}(q) = M(q) \sum_{i=1}^{n} \mathrm{tr}\left(\frac{\partial J(q)}{\partial q_i} J^{\dagger}(q)\right) \dot{q}_i$, $M(q) = \sqrt{\det(J(q) J^{\mathrm{T}}(q))}$.

5.1.4. Joint Physical Constraints

Let the joint position limits of the robot be denoted as $[\underline{q}, \overline{q}]$. Considering the control period T_s, the joint position constraints for the subsequent time step can be transformed into velocity constraints at the current moment:

$$\underline{q} - q \leq \dot{q} T_s \leq \overline{q} - q \tag{9}$$

5.2. Optimization Objectives

5.2.1. Self-Collision Avoidance

Employing an artificial potential field (APF) strategy for self-collision avoidance ensures safety during the dual-arm collaboration process [52,53]. Spherical bounding volumes are utilized to approximate the geometric shape of the robot [54,55]. When the minimum Euclidean distance between bounding volumes is less than a safety threshold, a repulsive force F_{ij} is generated along the line connecting their centers:

$$F_{ij} = \begin{cases} \eta\left(\frac{1}{d_{ij}} - \frac{1}{d_{safe}}\right)\frac{1}{d_{ij}^2} n_{ij}, & \text{if } d_{ij} < d_{safe}, \\ 0, & \text{if } d_{ij} \geq d_{safe}, \end{cases}$$

where d_{ij} indicates the minimum distance between bounding volumes i and j, d_{safe} represents the safety threshold, η is the repulsion gain coefficient, and n_{ij} denotes the unit vector indicating the direction of repulsion.

Assuming the joints possess viscous damping characteristics with a damping coefficient B, the desired avoidance velocity induced by the repulsive force is formulated as:

$$\dot{q}_{avoid} = B^{-1} \sum J_{ij}^{\mathrm{T}} F_{ij}.$$

Subsequently, $\dot{q}_{avoid}$ is incorporated into the cost function as the desired velocity of a secondary task, guiding the robot configuration away from self-collision regions.

5.2.2. Waist Joint Regulation

To ensure more anthropomorphic waist motion and prevent the waist from remaining in extreme pitch states for extended periods, an optimization objective is introduced to return the waist pitch joint to its nominal upright position q_{nom}. A negative feedback control law featuring saturation characteristics is designed to calculate the desired velocity $\dot{q}_{waist}$ of the waist joint:

$$\dot{q}_{waist} = \begin{cases} \overline{\dot{q}}_{waist}, & \text{if } -\lambda_{waist}(q_{waist} - q_{nom}) \geq \overline{\dot{q}}_{waist}, \\ \underline{\dot{q}}_{waist}, & \text{if } -\lambda_{waist}(q_{waist} - q_{nom}) \leq \underline{\dot{q}}_{waist}, \\ -\lambda_{waist}(q_{waist} - q_{nom}), & \text{otherwise.} \end{cases}$$

In this formulation, λ_{waist} denotes the regulation gain, while $\overline{\dot{q}}_{waist}$ and $\underline{\dot{q}}_{waist}$ represent the upper and lower bounds of the saturation velocity, respectively, with typically $\overline{\dot{q}}_{waist} = -\underline{\dot{q}}_{waist}$. The design of this saturation block aims to prevent excessively large regulation velocities from causing disturbances to the primary tasks.

5.3. Inverse Kinematics Solution

 doi:10.20944/preprints202604.1405.v1

Based on the aforementioned formulations, the optimal joint velocity vector $\dot{q}$ is obtained by solving the following problem:

$$\min_{\dot{q}} J(\dot{q}) = \|\dot{q} - \dot{q}_{avoid}\|_{W_1}^2 + \|\dot{q} - \dot{q}_{waist}\|_{W_2}^2 + \|\dot{q}\|^2$$

$$\text{s.t. } (5),\ (6),\ (7),\ (8),\ (9).$$

Where W_1 and W_2 indicate the weight matrices for collision avoidance and waist regulation, respectively. The penalty term $\|\dot{q}\|^2$ at the end of the cost function is employed to ensure numerical stability and minimize energy consumption. By solving the aforementioned QP problem, the controller can output safe and reliable joint velocities with millisecond-level real-time performance.

6. Simulations

The simulations primarily tested the methods proposed in Section 3. To verify the effectiveness and real-time performance of the proposed MC-SCCF, we conducted systematic evaluation tests in a simulation environment. The humanoid upper-body robot model utilized features 16 DoFs, comprising two 7-DoF spherical-revolute-spherical (SRS) manipulators and a 2-DoF waist. The robot is illustrated in Figure 5. The yellow translucent objects in the figure denote the bounding volumes attached to the robot links for self-collision avoidance. The relevant parameters of the algorithms are listed in Table 1.

6.1. Verification of the Reachability Determination Method

Before evaluating the reachability determination algorithm based on the differentiable surrogate model, it is necessary to obtain the accurate ground truth of the dual-arm target pose reachability as a comparison baseline. Because the employed manipulators feature an SRS configuration, analytical solutions for inverse kinematics exist. For a given dual-arm target pose, after verifying the absence of interference with the mobile base, we traverse all possible configurations within the waist joint limits at a resolution of 0.5° and solve for the analytical inverse kinematic solutions of both arms under the corresponding waist configuration. If analytical solutions exist for both arms under any waist configuration, the target pose is determined as truly reachable. Conversely, if no solution exists after traversing all waist configurations, it is determined as truly unreachable. The selection of bounding volumes is demonstrated in Figure 5.

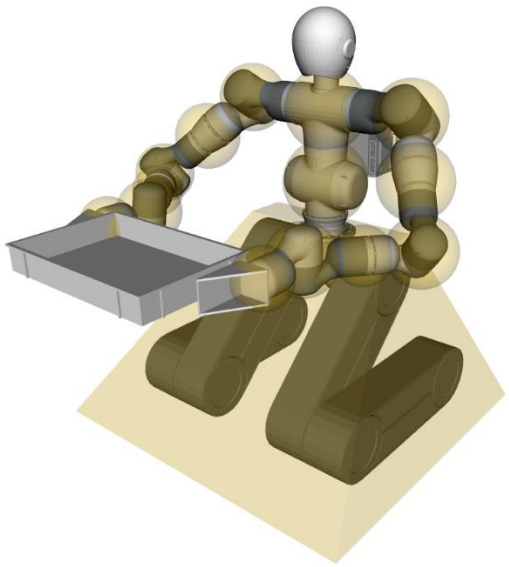


Figure 5. Simulation model of the robot and its bounding volumes.

Table 1. Algorithm Parameters.

Parameter	Value	Parameter	Value

 doi:10.20944/preprints202604.1405.v1

d_r	5cm	d	5cm
n_a	40	h	25cm
n_c	18	μ	0.15
β	0.015rad/Nm	R_{tray}	38cm
ε	1e-3rad	W_x	$\mathrm{diag}(100\mathrm{I}_3,\ 5\mathrm{I}_3)$
max_iters	50	W_u	I_3

In nonprehensile transportation tasks, a deterministic relative pose exists between the robot's dual-arm end-effectors and the tray. Instead of directly evaluating the reachability of arbitrary dual-arm target poses, our evaluation strategy involves randomly generating tray poses in space and thereafter deriving the target poses of the dual-arm end-effectors. Accordingly, this approach aligns more closely with practical task requirements.

Specifically, within a spherical space with a radius of 1.5 m centered at the robot's waist base joint, 10^6 target poses of trays with varying widths (25 cm to 80 cm) and random orientations are generated. The accuracy of the algorithm is evaluated by comparing its determination results against the aforementioned ground truth. The experimental results are presented in Table 2.

Table 2. Confusion Matrix for Reachability Determination Results.

Ground Truth	Detected as Reachable	Detected as Unreachable
Reachable	348023(TP, True Positive)	12869(FN, False Negative)
Unreachable	8558(FP, False Positive)	630550(TN, True Negative)

The precision of the algorithm is $\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \times 100\% \approx 97.60\%$. This indicates that when the algorithm determines a pose as reachable, the probability of the robot reaching it is 97.60%. The overall accuracy of the algorithm is $\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{Total}} \times 100\% \approx 97.86\%$.

The computational efficiency results for determining target pose reachability applying the proposed method are as follows: the average number of iterations to calculate the optimal waist configuration for a single reachability query is 8.5, with a standard deviation of 2.4. The average computation time for a single query is 60 μs, with a standard deviation of 34 μs.

Although the test results cannot guarantee absolute accuracy, an accuracy exceeding 97% is sufficient for practical applications. Minor prediction deviations can be eliminated during actual engineering deployment by introducing a static safety margin. Furthermore, the computational efficiency of the algorithm satisfies the expected objectives, and this computational overhead is sufficiently low to support the real-time control requirements of teleoperation.

6.2. Evaluation of the Workspace Boundary Determination Algorithm

To objectively evaluate the accuracy and computational efficiency of the workspace boundary determination algorithm, the experiment directly employs the bisection search for the workspace boundary (without applying the predictive stopping pose pruning function). The experiment randomly generates 10^4 initial tray positions and corresponding velocity directions within the workspace. A high-precision search for the true workspace boundary distance is conducted combining the true reachability determination method from Section 5.1 with the bisection method, setting the iteration termination condition to a sampling interval length of less than 1 cm. The true distance is denoted as D_0, while the boundary distance detected by the proposed method is defined as $\hat{D}$. The detection deviation of the distance determination is defined as $E = \hat{D} - D_0$. The experimental results are revealed as follows.

 doi:10.20944/preprints202604.1405.v1

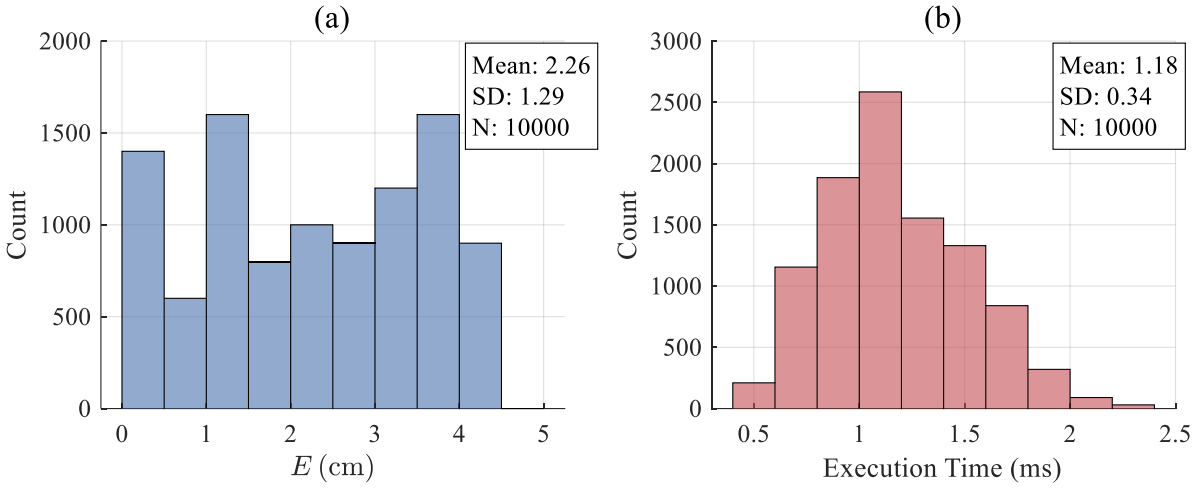


Figure 6. Experimental results of the workspace boundary determination algorithm. (a) Histogram of determination errors. (b) Histogram of algorithm execution times.

The experimental results demonstrate that the proposed method exhibits high precision in estimating the boundary distance. As illustrated in Figure 6(a), the determination deviation distribution is relatively even, with a mean of only 2.26 cm and a standard deviation of 1.29 cm. Among the 10,000 random samples, the maximum detection deviation remains within 4.5 cm. This detection deviation is primarily restricted by the grid resolution (5 cm) of the discrete reachability map; thus, improving this resolution can further enhance the determination accuracy. During actual deployment, to completely eliminate the risk of boundary violation, the safety can be guaranteed at the algorithmic level simply by setting the practical safe boundary distance in the control system to the detected distance minus 4.5 cm.

The time consumption distribution histogram in Figure 6(b) exhibits the high efficiency of this search strategy. The average time for a single workspace boundary detection is 1.18 ms, with a standard deviation of 0.34 ms, and the maximum detection time consistently remains below 2.5 ms. The computational efficiency of this algorithm can fulfill the requirements of real-time control frameworks for most humanoid robots.

6.3. Evaluation of the Safe Velocity Filter

To verify the performance of the safe velocity filter based on the improved CBF proposed in this paper, the experiment introduces two control strategies based on the conventional CBF with different parameters as comparative baselines. The parameter settings for each control scheme are detailed in Table 3.

Table 3. Parameters for Different Safe Velocity Filter Schemes.

Mode	γ	$a_{\max}$	h_t
CBF 1	1.20	——	——
CBF 2	1.85	——	——
Improved CBF	——	0.7m/s²	0.15m

A dynamic deceleration simulation scenario is designed: the robot's dual arms support a tray carrying an object and execute linear motion toward the workspace boundary at various constant initial velocities ranging from 0.5 m/s to 0.9 m/s, applying the three aforementioned control schemes respectively. The velocity and orientation of the object are recorded during the experiment, focusing on statistically analyzing the maximum tilt angle and the deceleration distance of the object during motion. The comparative simulation results are exhibited in Figure 7.

 doi:10.20944/preprints202604.1405.v1

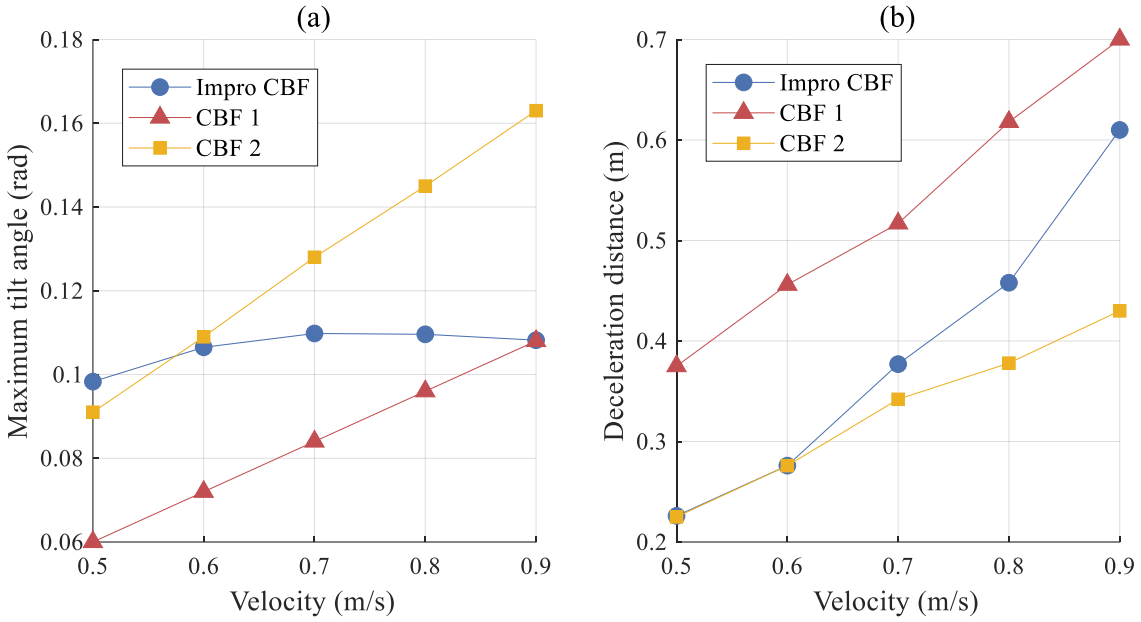


Figure 7. Results of safe velocity filters with three different CBFs. (a) Relationship between the object's maximum tilt angle during deceleration and the object's velocity. (b) Relationship between the object's deceleration distance and the object's velocity.

Under conventional linear CBF control (CBF 1 and CBF 2), both the maximum tilt angle and the deceleration distance of the object are positively correlated with the object's velocity. A horizontal comparison of the two conventional schemes reveals that at the same velocity, a smaller γ value reduces the maximum tilt angle but increases the deceleration distance. This indicates that mitigating orientation jitter during the deceleration phase by decreasing γ comes at the cost of reducing the robot's flexibility and operational workspace. This experimental phenomenon is consistent with the analysis presented in Section 3.

When employing the improved CBF scheme, the maximum tilt angle during the deceleration phase no longer exhibits a notable correlation with the velocity. Theoretically, the maximum tilt angle should not vary with the object's velocity. In practice, due to the trajectory smoothing effect of the middle-layer controller, the maximum tilt angle is influenced by the velocity to a certain extent, which is expected. Nonetheless, compared to the conventional schemes, the improved CBF achieves an effective limitation on the maximum tilt angle of the deceleration orientation, fulfilling the anticipated objective of maintaining a controlled maximum tilt angle during deceleration.

Comparing the results of the improved CBF and CBF 1, at an object velocity of 0.9 m/s, the maximum tilt angles during the deceleration phase are approximately identical. However, regarding the deceleration distance, the improved CBF yields a shorter distance than CBF 1. Comparing the improved CBF with CBF 2, the deceleration distances are roughly equivalent at velocities of 0.5 m/s and 0.6 m/s; in contrast, as the velocity increases, the maximum tilt angle under CBF 2 becomes markedly larger than that under the improved CBF.

As a result, compared to conventional CBFs, the improved method introduces a maximum acceleration constraint and utilizes a piecewise transition, achieving a tradeoff between constrained space and orientation controllability. It is less invasive to user commands when far from the boundary, thereby preserving a larger operational space; meanwhile, when approaching the boundary, the deceleration is smoother with controlled acceleration, effectively mitigating the risk of object toppling.

7. Experiments

7.1. Experimental Platform and Task Setup

To verify the effectiveness of the proposed MC-SCCF in a physical system, this section designs an experiment for a humanoid robot executing a nonprehensile transportation task. The hardware

 doi:10.20944/preprints202604.1405.v1

system for teleoperation is illustrated in Figure 8. The experiment utilizes our custom-built dual-arm humanoid robot [56]. Its upper body comprises two 7-DoF SRS anthropomorphic manipulators and 2-DoF (yaw and pitch) head and waist joints, mounted on a tracked mobile base. The experiments in this study solely control the upper body manipulation of the robot, without involving the locomotion of the mobile base.

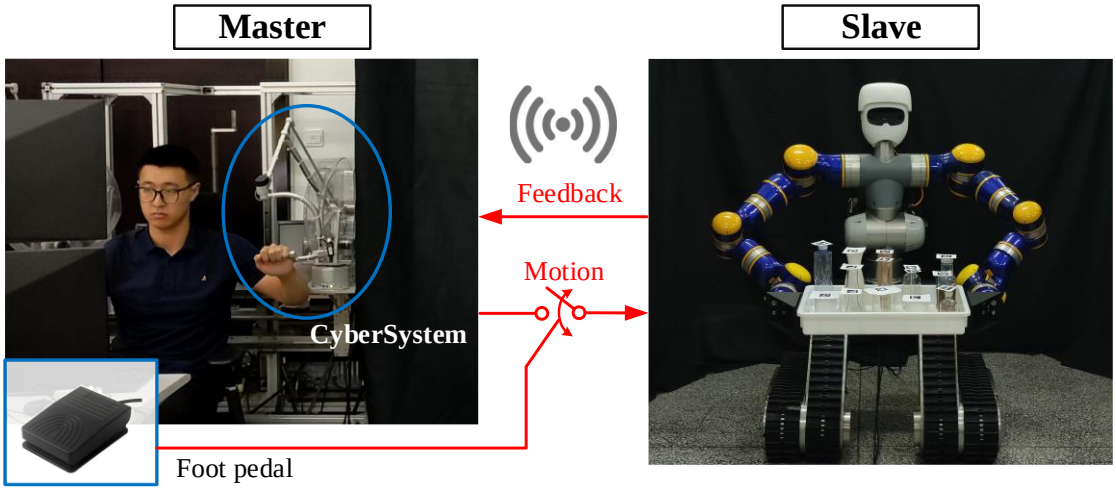
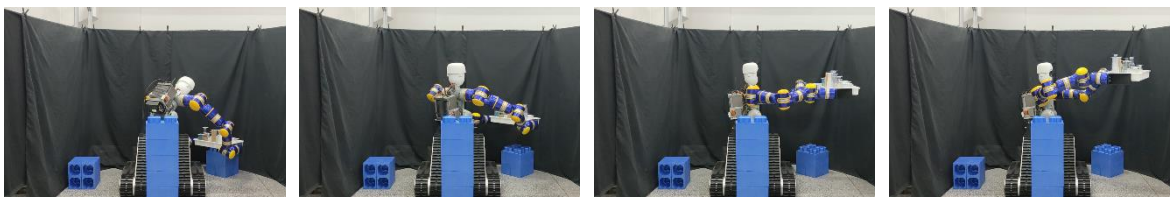


Figure 8. Teleoperation hardware system.

The master device employs a CyberSystem hand controller to track the 3D position data of the human wrist, which is mapped to the humanoid robot's workspace via an incremental Cartesian space mapping. Because the robot features a large workspace, the displacement of the master user is scaled up by a factor of 4 before being mapped to the robot's workspace. Additionally, the master device is equipped with a foot pedal functioning as a clutch switch. Thus, operators can adjust their arm positions at any time during the task to achieve more comfortable operation.

A computer running Ubuntu 20.04, equipped with an Intel Core i9-9900K CPU and 64 GB of RAM, is utilized to execute the teleoperation controller. The controller is implemented in C++, applying multithreading technology to separate hand controller data acquisition, teleoperation control, and state feedback reception. The data acquisition thread collects user input data from the master device at a frequency of 40 Hz, with a communication latency between the master device and the teleoperation controller strictly below 1 ms. The teleoperation control operates within the planning thread at a control frequency of 20 Hz, and the communication latency between the teleoperation controller and the robot is less than 20 ms. The optimization problem is solved using the open-source qpOASES package. The thread dedicated to receiving robot state feedback processes a 24 Hz video stream and 100 Hz robot joint states.

The experiment requires operators to control the robot via the master device to execute an obstacle-avoidance transportation task. Operators must teleoperate the robot's dual arms to lift a tray carrying multiple objects from a starting point at the rear left, cross an obstacle directly in front, and ultimately place it smoothly in a target area on the right. The experimental process is demonstrated in Figure 9.



 doi:10.20944/preprints202604.1405.v1

(e) (f) (g) (h)

Figure 9. Robot transportation process. (a) The robot reaches the initial position to begin transportation. (b) Lifting the tray. (c) Elevating the tray. (d) Further elevating to a height sufficient to clear the obstacle. (e) Crossing the obstacle. (f) Completely clearing the obstacle. (g) Lowering the tray. (h) Reaching the target position.

The dimensions of the tray are 560 mm $\times$ 375 mm $\times$ 80 mm. 14 objects to be transported are randomly positioned on the tray. These objects comprise common items, including vases, water bottles, wooden blocks, and glassware, with heights ranging from 100 mm to 235 mm, as exhibited in Figure 8. Because objects inevitably slide during motion, fiducial markers are attached to their tops, and a camera is employed to detect their sliding distances. The camera is fixed above the robot platform and is exclusively utilized to detect sliding distances for result analysis; the positions of the objects are practically under open-loop control.

Visual feedback for the operator is provided by the stereo camera on the robot's head. No global field of view (FoV) is available during the experiment, which is highly typical in practical teleoperation task execution scenarios. The controller provides kinematic simulations, where both the actual and desired configurations of the robot are simultaneously displayed on the simulation interface. Operators can switch perspectives within the simulation interface to obtain an intuitive understanding of the teleoperated robot's current state.

7.2. Experimental Conditions and Evaluation Metrics

This section defines the following three control modes:

- Mode A (Hard Boundary Mode): The middle-layer MPC (ensuring multi-object dynamic constraints) and low-level kinematic hard limits are enabled, whereas the top-layer safe velocity filter is deactivated.
- Mode B (Conservative Assistive Mode): The middle-layer MPC is enabled, and a conventional linear CBF ($\gamma = 1.5$) is employed at the top layer as the safe velocity filter.
- Mode C (Proposed Multiple-Constraint Framework): The middle-layer MPC is enabled, and the improved CBF proposed in this study is utilized at the top layer (with algorithm parameters identical to those in the Section 6 simulations).

To reduce experimental randomness, a total of 10 subjects were recruited. All participants signed an informed consent form prior to participating in the experiment. During the procedures, exclusively the input trajectories from the subjects operating the hand controller were recorded, strictly excluding any sensitive or privacy-related data. The experimental protocol was approved by the institutional ethics committee (Approval No. HIT-2025013) and conducted in accordance with the Declaration of Helsinki. Among the subjects, one was female and nine were male, with four possessing prior teleoperation experience. All subjects exhibited normal cognitive levels and equipment operation capabilities. Before the experiment commenced, the designers spent 40 minutes explaining the task to each subject and assisting them in familiarizing themselves with the equipment. For each trial, the initial robot configuration, the placement of the tray, and the arrangement of objects on the tray remained strictly identical. Each subject performed the experiment using the three modes separately, conducting four trials per mode. This yielded a total of 120 experimental trials.

The evaluation metrics encompass three dimensions: task metrics, subjective evaluation metrics, and control performance metrics. The task metrics include the following:

- Task Success Rate: A success is defined as the tray reaching the target position without colliding with obstacles, and without any objects toppling or falling.

 doi:10.20944/preprints202604.1405.v1

- Completion Time (T): Timing initiates when the operator lifts the tray and concludes when the tray is placed at the target position.
- Total Object Sliding Distance (S): The sum of the sliding distances of all objects on the tray. A shorter sliding distance indicates smoother motion.
- Activation Time Ratio of the Safe Velocity Filter: Calculated as $\mathrm{T}_{\mathrm{Filter}}/\mathrm{T}\times 100\%$, where $\mathrm{T}_{\mathrm{Filter}}$ denotes the duration for which the filter remains active.
- Number of Pauses During the Task: Operators can adjust the master device to the most optimal operational area during pauses.

Subjective metrics were collected via a 7-point Likert scale, gathering the subjects' scores on operational intuitiveness, task difficulty, cognitive load (concentration), and operational confidence.

Control performance metrics comprise the robot's trajectory tracking accuracy, relative pose error between dual-arm end-effectors, manipulability, minimum distance, and computation time. These metrics are primarily utilized to analyze the effectiveness of the low-level controller.

7.3. Experimental Results

7.3.1. Task Metrics

The statistical experimental results under the three modes are demonstrated in Table 4 and Figure 10. Regarding the task success rate, Mode A achieved 82.5%. The failures were caused by the lack of inverse kinematic solutions leading to emergency stops, as well as the toppling and slipping of objects. This indicates that when facing complex workspace boundaries and multiple dynamic constraints, relying solely on the operator's intuition is insufficient to maintain the system within the safe region. In contrast, Modes B and C, which incorporate the top-layer safe velocity filter, both achieved a 100% success rate. This result intuitively confirms the necessity and effectiveness of the velocity filtering mechanism in evading kinematic boundary violations and preventing task failures.

Table 4. Success Rates of the Three Control Modes.

Control Mode	Success Rate
Mode A	82.5%
Mode B	100%
Mode C	100%

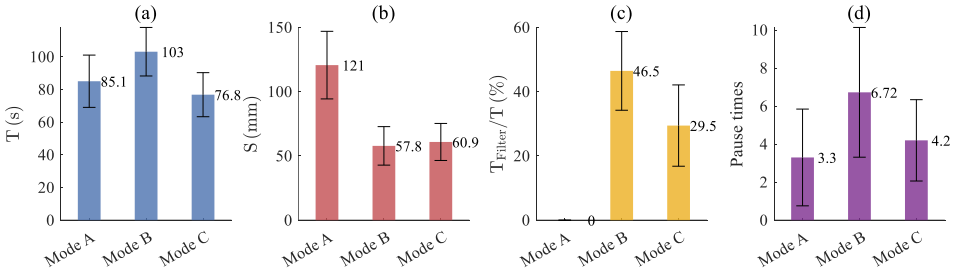


Figure 10. Experimental results under the three control modes.

The completion time T and the number of pauses intuitively reflect the system's efficiency and the fluency of human-robot collaboration. Mode C exhibited the optimal comprehensive performance, requiring the shortest task duration (76.8 s) while maintaining the number of pauses at a reasonable level (4.2). In contrast, although Mode B simultaneously guaranteed safety, it required the longest task duration (103 s) and the highest number of pauses (6.72). The underlying reason is that the conventional linear CBF is highly conservative and frequently modifies user input commands, causing severe positional mapping deviations between the master and slave devices. The operator's subjective displacement fails to elicit a response at the slave end, forcing the operator to frequently depress the clutch switch to readjust the master device's position. This not only increases the operation time but also disrupts the continuity of teleoperation.

 doi:10.20944/preprints202604.1405.v1

The total object sliding distance S measures the smoothness of the transportation process. Because touching hard boundaries in Mode A easily triggers emergency stops and abrupt acceleration changes, the object sliding distance reached 121 mm. In Modes B and C, where the top-layer safe velocity filter is enabled, the sliding distance was substantially suppressed (57.8 mm and 60.9 mm, respectively). The sliding distance in Mode B was slightly smaller than that in Mode C, as its conservative filtering strategy resulted in a slower overall motion velocity. Nonetheless, while maintaining high efficiency, Mode C still controlled the sliding distance to an optimal level nearly equivalent to that of Mode B. This supports the efficacy of the improved CBF in controlling acceleration and smoothness during the deceleration phase.

Regarding the activation time ratio of the safe velocity filter (T_{Filter}/T), Mode C still triggered the safe velocity regulation mechanism for 29.5% of the time. The experimental task is not inherently complex, and the robot possesses a large workspace; nevertheless, operators easily drive the robot near the workspace boundaries. This is primarily attributed to two reasons. First is the configuration mismatch. Human operators have a different kinematic configuration from the robot. Particularly when holding a tray of fixed width with both arms, a closed-loop kinematic chain is formed. Operators cannot maintain a clear perception of the robot's configuration at all times. Thus, trajectories input by the operator at the master device frequently cause the robot to touch the workspace boundaries at the slave end.

The second vital reason is that teleoperation typically relies on screens or cameras with specific viewing angles, leading to inaccurate depth perception and limited telepresence. When facing unstructured environments, highly conservative operations are often required. For instance, during the experiment, operators could not accurately evaluate the safe clearance between the tray bottom and the obstacle. To evade collision risks, subjects generally preferred conservative obstacle avoidance strategies, such as excessive elevation or excessive retraction, which accordingly made it easier to drive the robot to its extension limits. However, precisely because operators inevitably drive the robot close to the boundaries, the top-layer velocity modulation mechanism proves to be particularly crucial.

7.3.2 Subjective Evaluation Metrics

The statistical results of the Likert scale align closely with the objective data, as demonstrated in Figure 11. The figure illustrates evaluation results across four dimensions: the intuitiveness of the control mode (7 indicates extremely intuitive), the difficulty of the task (7 indicates extremely easy), the cognitive load and concentration (7 indicates extremely effortless), and the confidence in completing similar future tasks (7 indicates highly confident).

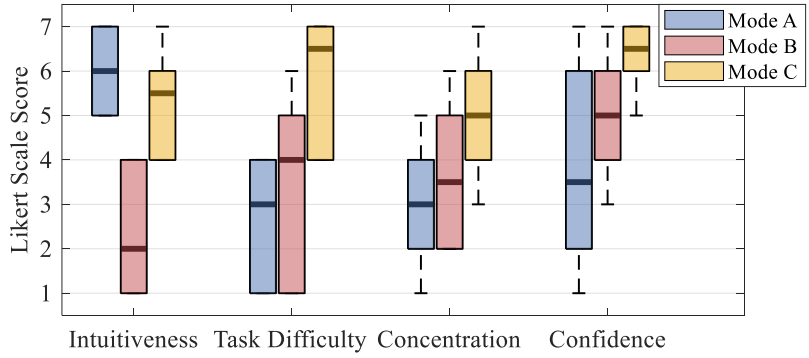


Figure 11. Subjective feedback results of the subjects after using the three control modes. Results were compiled using a 7-point Likert scale, where higher scores represent better subjective experiences.

Mode A: Although the task success rate under this mode was relatively low (only 82.5%) and objective metrics indicated a larger object sliding distance, operators generally awarded it the highest score for operational intuitiveness. This is primarily because Mode A did not introduce the safe velocity filter; accordingly, user input commands could be directly and entirely transformed into the tray's motion, realizing a *what-you-see-is-what-you-get* experience. Nonetheless, this mechanism

 doi:10.20944/preprints202604.1405.v1

lacking safety protection also resulted in Mode A receiving the lowest task difficulty score. In assessing confidence for future tasks, the operators' scores exhibited considerable variance. Operators who assigned low scores believed that without the protection of the top-layer safe velocity filter, the robot was highly prone to touching workspace boundaries during operation, rendering the task more difficult. This group also assigned relatively lower scores for cognitive load (concentration) in Mode A. In contrast, operators who assigned high scores were more inclined to believe that the superior operational intuitiveness of Mode A was more conducive to task completion; these high-scoring individuals simultaneously assigned higher scores for concentration.

Mode B: Benefiting from the introduction of the conventional linear CBF, the overall evaluation of task difficulty for this mode was easier than that of Mode A, though the statistical results similarly presented considerable variance. The core reason subjects found it easy was that the algorithm guaranteed safety; the tray would not cross workspace boundaries, thereby avoiding situations without inverse kinematic solutions. However, the reason this mode was perceived as difficult was its poor operational intuitiveness. Due to the conservative filtering mechanism of the conventional CBF, master inputs were frequently suppressed by the system, triggering a notable sluggishness in operation. This unstable mapping of master-slave positions, coupled with the robot's frequent slow motions, substantially increased the operators' cognitive load.

Mode C: Integrating all subjective metrics, Mode C provided a superior interaction experience among the tested modes. Subjects required the lowest concentration when executing tasks, indicating that this mode possesses a minimal cognitive load, allowing users to complete the nonprehensile transportation experiment with ease. Furthermore, this mode not only overcame the sluggishness noted in Mode B and preserved adequate operational intuitiveness but also provided operators with the highest confidence for completing future tasks. Overall, driven by the MC-SCCF integrated with the improved CBF, Mode C balances system safety and operational intuitiveness, making it the most preferred among the three modes.

7.3.3. Control Performance Metrics

To verify the effectiveness of the low-level controller and more intuitively demonstrate the experimental process, a set of trials using Mode C was selected. Figure 12 illustrates the variations of relevant variables during the experiment, where the red regions indicate periods of operational pauses.

During the experiment, the maximum motion velocity of the object exceeded 0.25 m/s, and its maximum tilt angle exceeded 6°. Throughout the entire motion, the trajectory position error remained below 10^{-4} m, the relative position error between the dual-arm end-effectors was below 6×10^{-5} m, and the relative orientation error was below 8×10^{-5} rad. This achieved the objectives of trajectory tracking and maintaining a constant relative pose between the dual-arm end-effectors.

Although the robot's configuration underwent considerable changes during transportation, safety was consistently guaranteed. The robot's manipulability remained above 0.04, and the minimum distance between its bounding volumes was greater than 0.29 m. The goals of evading singular configurations and avoiding self-collisions were realized. During the lifting of the tray and obstacle crossing, driven by the waist regulation objective, the waist pitch axis automatically returned to its upright state after providing the necessary forward-reaching compensation. This not only prevented the waist joint from remaining in a high-load configuration for extended periods but also rendered the robot's overall operational posture more natural and anthropomorphic.

Within each control period during the task, the inverse kinematics solving time was approximately 3 ms, with a peak below 6 ms. This computational overhead is lower than the control period of the teleoperation system, meets the real-time requirements of teleoperation control.

The data indicate that the low-level controller can effectively achieve the design objectives of multiple-constraint coordinated processing and configuration optimization while fulfilling real-time performance requirements.

 doi:10.20944/preprints202604.1405.v1

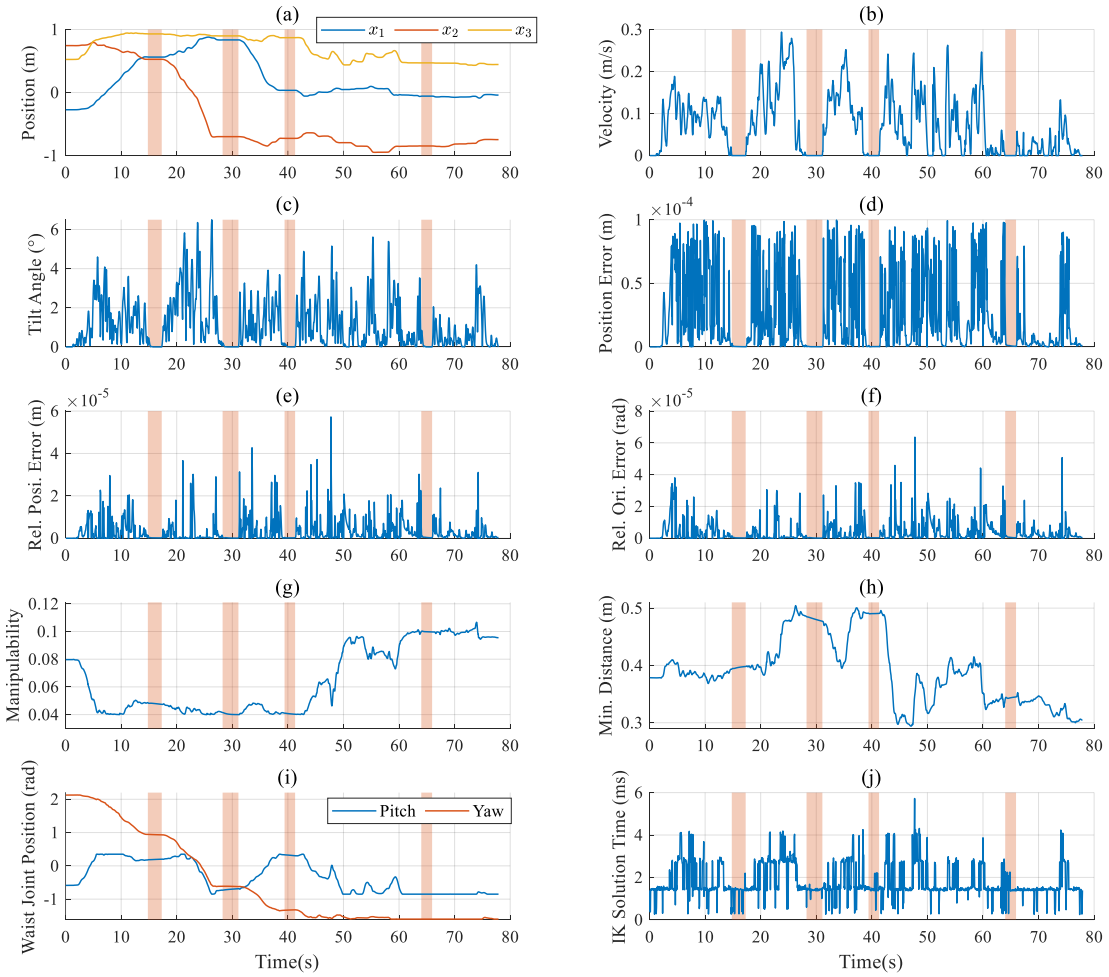


Figure 12. Variation curves of relevant variables during the experiment. (a) Object position. (b) Object velocity. (c) Object tilt angle. (d) Object position error. (e) Relative position error between dual-arm end-effectors. (f) Relative *orientation* error between dual-arm end-effectors. (g) Robot manipulability. (h) Minimum distance between bounding volumes. (i) Robot waist joint angles. (j) Inverse kinematics solving time.

8. Conclusions

To address the challenges of multiple constraint conflicts and safety control in humanoid robot teleoperation for nonprehensile transportation tasks, this article proposes a hierarchically decoupled Multiple-Constraint Safety-Critical Control Framework (MC-SCCF). This framework divides the system into a top layer responsible for intrinsic safety regarding workspace boundaries, a middle layer managing object dynamic equilibrium, and a low level handling various kinematic hard constraints and configuration optimization.

System simulation tests demonstrate that the target pose determination method based on the differentiable reachability surrogate model exhibits high accuracy and real-time performance. The average computation time per query is only 60 μs, with a prediction accuracy reaching 97.86%. The bisection search algorithm for the workspace boundary distance can control the average detection error to 2.26 cm (with a maximum detection deviation below 4.5 cm) at a marginal computational overhead of 1.18 ms on average. In dynamic deceleration scenarios, the proposed improved CBF based on the maximum acceleration constraint overcomes the overly conservative limitations of the conventional linear CBF. It limits the maximum tilt angle of the object as the robot approaches the boundary, achieving smooth deceleration and orientation controllability. The simulation data adequately prove the theoretical effectiveness of the proposed safe velocity filter and reachability determination methods.

We recruited 10 subjects to conduct multi-object obstacle-crossing transportation teleoperation experiments on a physical dual-arm humanoid robot platform. Task metrics reveal that the control mode employing the proposed framework (Mode C) achieved a 100% task success rate, whereas

 doi:10.20944/preprints202604.1405.v1

Mode A, lacking top-layer protection, had a success rate of only 82.5%. Regarding efficiency and smoothness, Mode C shortened the task completion time to 76.8 s, which is superior to the 103 s of the conventional conservative CBF strategy (Mode B). Furthermore, this framework reduced the operators' average number of pauses to 4.2 and markedly suppressed the total object sliding displacement to 60.9 mm (a reduction of nearly 50% compared to the 121 mm in Mode A). Subjective Likert scale feedback indicates that this control scheme considerably reduces the operators' cognitive load and provides users with high operational intuitiveness and task confidence. These statistical results confirm the operators' recognition of this scheme. Additionally, the completion of extensive experiments validates the stability of the control architecture during deployment on a physical robot.

Control performance data throughout the experiments further demonstrate that the low-level system, when handling highly dynamic teleoperation commands, effectively maintains a constant relative pose between the dual-arm end-effectors and high-precision trajectory tracking. The trajectory position error consistently remained below 10^{-4} m, while the relative position and orientation errors between the dual arms were below 6×10^{-5} m and 8×10^{-5} rad, respectively. Moreover, the robot's manipulability consistently exceeded 0.04, the minimum distance between bounding volumes was maintained above 0.29 m, and configuration optimization objectives such as automatic waist regulation were realized. The average solving time of the inverse kinematics algorithm was approximately 3 ms (with a peak below 6 ms), fully satisfying the real-time requirements of teleoperation and directly verifying the effectiveness of the low-level controller.

In summary, the control framework proposed in this article successfully resolves the underlying conflicts between stringent task dynamic constraints and complex intrinsic constraints of humanoid robots. Under the premise of ensuring system safety, this research substantially improves the efficiency and user experience of nonprehensile transportation tasks, providing a practical and feasible control-theoretic solution for humanoid robots to execute complex collaborative tasks in unstructured environments.

Author Contributions: Funding acquisition, methodology, writing—review and editing, Fenglei Ni; software, writing—original draft preparation, Xinyang Fan; All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under the Basic Science Center Program for "Space Robot Intelligent Manipulation" (Grant No. T2388101).

Data Availability Statement: Data is contained within the article. The original contributions presented in this 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.

Abbreviations

The following abbreviations are used in this manuscript:

MC-SCCF	Multiple-Constraint Safety-Critical Control Framework
CBF	Directory of open access journals
QP	Three letter acronym
WBC	Whole-Body Control
CLF	Control Lyapunov Functions
MPC	Model Predictive Control
VO-MPC	Virtual Object Model Predictive Control
PSP	Predicted Stopping Pose
CoM	Center of Mass
ZMP	Zero Moment Point
FCNN	Fully Connected Neural Network
DoF	Degree of Freedom
APF	Artificial Potential Field
SRS	Spherical-Revolute-Spherical
FoV	Field of View

 doi:10.20944/preprints202604.1405.v1

References

1. L. Chen, H. Yu, L. Zhang, A. Naceri, A. Swikir, and S. Haddadin, Trajectory Planning for Non-Prehensile Object Transportation, *in* 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (2024): pp. 9939–9946. https://doi.org/10.1109/IROS58592.2024.10801587.
2. M. Selvaggio, A. Garg, F. Ruggiero, G. Oriolo, and B. Siciliano, Non-Prehensile Object Transportation via Model Predictive Non-Sliding Manipulation Control, *IEEE Transactions on Control Systems Technology* 31 (5) (2023) 2231–2244. https://doi.org/10.1109/TCST.2023.3277224.
3. O. Kroemer, S. Niekum, and G. Konidaris, A Review of Robot Learning for Manipulation: Challenges, Representations, and Algorithms, *Journal of Machine Learning Research* 22 (30) (2021) 1–82.
4. M. Aboelnasr, H.M. Bahaa, and O. Mokhiamar, Novel use of the Monte-Carlo methods to visualize singularity configurations in serial manipulators, *Journal of Mechanical Engineering and Sciences* 15 (2) (2021) 7948–7963. https://doi.org/10.15282/jmes.15.2.2021.02.0627.
5. J. Zhao, Z. Zhao, X. Yang, L. Zhao, G. Yang, and H. Liu, Inverse kinematics and workspace analysis of a novel SSRMS-type reconfigurable space manipulator with two lockable passive telescopic links, *Mechanism and Machine Theory* 180 (2023) 105152. https://doi.org/10.1016/j.mechmachtheory.2022.105152.
6. H. Wang, Z. Zhou, X. Zhong, and Q. Chen, Singular Configuration Analysis and Singularity Avoidance with Application in an Intelligent Robotic Manipulator, *Sensors* 22 (3) (2022) 1239. https://doi.org/10.3390/s22031239.
7. D. Calzolari, R. Lampariello, and A.M. Giordano, Singularity Maps of Space Robots and their Application to Gradient-based Trajectory Planning, *in* 16th Robotics: Science and Systems, RSS 2020, Virtual Conference, (2020).
8. F. Zacharias, C. Borst, and G. Hirzinger, Capturing robot workspace structure: representing robot capabilities, *in* 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, (2007): pp. 3229–3236. https://doi.org/10.1109/IROS.2007.4399105.
9. J. Lee and B.-K. Cho, Design of 9-DOF humanoid arms inspired by the human's inner shoulder to enhance versatility and workspace, *Robotics and Autonomous Systems* 166 (2023) 104447. https://doi.org/10.1016/j.robot.2023.104447.
10. M. Selvaggio, M. Cognetti, S. Nikolaidis, S. Ivaldi, and B. Siciliano, Autonomy in Physical Human-Robot Interaction: A Brief Survey, *IEEE Robotics and Automation Letters* 6 (4) (2021) 7989–7996. https://doi.org/10.1109/LRA.2021.3100603.
11. R.J. Ansari and Y. Karayiannidis, Task-Based Role Adaptation for Human-Robot Cooperative Object Handling, *IEEE Robotics and Automation Letters* 6 (2) (2021) 3592–3598. https://doi.org/10.1109/LRA.2021.3064498.
12. J. Oh, O. Sim, H. Jeong, and J.-H. Oh, Humanoid whole-body remote-control framework with delayed reference generator for imitating human motion, *Mechatronics* 62 (2019) 102253. https://doi.org/10.1016/j.mechatronics.2019.102253.
13. J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, The Pinocchio C++ library : A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives, *in* 2019 IEEE/SICE International Symposium on System Integration (SII), (2019): pp. 614–619. https://doi.org/10.1109/SII.2019.8700380.

 doi:10.20944/preprints202604.1405.v1

14. D. Valencia, J. Jia, R. Li, A. Hayashi, M. Lecchi, R. Terezakis, T. Gee, M. Liarokapis, B.A. MacDonald, and H. Williams, Comparison of Model-Based and Model-Free Reinforcement Learning for Real-World Dexterous Robotic Manipulation Tasks, *in* 2023 IEEE International Conference on Robotics and Automation (ICRA), *IEEE*, London, United Kingdom, (2023): pp. 871–878. https://doi.org/10.1109/ICRA48891.2023.10160983.
15. S. Katayama, M. Murooka, and Y. Tazaki, Model predictive control of legged and humanoid robots: models and algorithms, *Advanced Robotics* 37 (5) (2023) 298–315. https://doi.org/10.1080/01691864.2023.2168134.
16. N. Mansard, O. Stasse, P. Evrard, and A. Kheddar, A Versatile Generalized Inverted Kinematics Implementation for Collaborative Humanoid Robots: The Stack of Tasks, (n.d.).
17. G. Romualdi, N.A. Villa, S. Dafarra, D. Pucci, and O. Stasse, Whole-Body Control and Estimation of Humanoid Robots with Link Flexibility, *in* 2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids), (2022): pp. 104–111. https://doi.org/10.1109/Humanoids53995.2022.10000157.
18. J. Kim, R.T. Fawcett, V.R. Kamidi, A.D. Ames, and K.A. Hamed, Layered Control for Cooperative Locomotion of Two Quadrupedal Robots: Centralized and Distributed Approaches, *IEEE Transactions on Robotics* 39 (6) (2023) 4728–4748. https://doi.org/10.1109/TRO.2023.3319896.
19. C. Dawson, S. Gao, and C. Fan, Safe Control With Learned Certificates: A Survey of Neural Lyapunov, Barrier, and Contraction Methods for Robotics and Control, *IEEE Transactions on Robotics* 39 (3) (2023) 1749–1767. https://doi.org/10.1109/TRO.2022.3232542.
20. M. Kang, Y. Cho, and S.-E. Yoon, RCIK: Real-Time Collision-Free Inverse Kinematics Using a Collision-Cost Prediction Network, *IEEE Robotics and Automation Letters* 7 (1) (2022) 610–617. https://doi.org/10.1109/LRA.2021.3128238.
21. M.A. Ali, H. Andy Park, and C.S.G. Lee, Closed-form inverse kinematic joint solution for humanoid robots, *in* 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, (2010): pp. 704–709. https://doi.org/10.1109/IROS.2010.5649842.
22. T. Sandakalum and M.H. Ang, Motion Planning for Mobile Manipulators—A Systematic Review, *Machines* 10 (2) (2022) 97. https://doi.org/10.3390/machines10020097.
23. O. Porges, T. Stouraitis, C. Borst, and M.A. Roa, Reachability and Capability Analysis for Manipulation Tasks, *in* M.A. Armada, A. Sanfeliu, M. Ferre (Eds.), ROBOT2013: First Iberian Robotics Conference, *Springer International Publishing*, Cham, (2014): pp. 703–718. https://doi.org/10.1007/978-3-319-03653-3_50.
24. T. Birr, C. Pohl, and T. Asfour, Oriented Surface Reachability Maps for Robot Placement, *in* 2022 International Conference on Robotics and Automation (ICRA), (2022): pp. 3357–3363. https://doi.org/10.1109/ICRA46639.2022.9811600.
25. M. Murooka, I. Kumagai, M. Morisawa, F. Kanehiro, and A. Kheddar, Humanoid Loco-Manipulation Planning Based on Graph Search and Reachability Maps, *IEEE Robotics and Automation Letters* 6 (2) (2021) 1840–1847. https://doi.org/10.1109/LRA.2021.3060728.
26. A. Makhal and A.K. Goins, Reuleaux: Robot Base Placement by Reachability Analysis, *in* 2018 Second IEEE International Conference on Robotic Computing (IRC), (2018): pp. 137–142. https://doi.org/10.1109/IRC.2018.00028.

 doi:10.20944/preprints202604.1405.v1

27. J. Dong and J.C. Trinkle, Orientation-based reachability map for robot base placement, *in* 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (2015): pp. 1488–1493. https://doi.org/10.1109/IROS.2015.7353564.
28. S. Jauhri, J. Peters, and G. Chalvatzaki, Robot Learning of Mobile Manipulation With Reachability Behavior Priors, *IEEE Robotics and Automation Letters* 7 (3) (2022) 8399–8406. https://doi.org/10.1109/LRA.2022.3188109.
29. L. Petrović, F. Marić, I. Marković, J. Kelly, and I. Petrović, Trajectory Optimization with Geometry-Aware Singularity Avoidance for Robot Motion Planning, *in* 2021 21st International Conference on Control, Automation and Systems (ICCAS), (2021): pp. 1760–1765. https://doi.org/10.23919/ICCAS52745.2021.9650039.
30. S.A. Bowyer, B.L. Davies, and F. Rodriguez y Baena, Active Constraints/Virtual Fixtures: A Survey, *IEEE Transactions on Robotics* 30 (1) (2014) 138–157. https://doi.org/10.1109/TRO.2013.2283410.
31. M. Mühlbauer, T. Hulin, B. Weber, S. Calinon, F. Stulp, A. Albu-Schäffer, and J. Silvério, A Probabilistic Approach to Multi-Modal Adaptive Virtual Fixtures, *IEEE Robotics and Automation Letters* 9 (6) (2024) 5298–5305. https://doi.org/10.1109/LRA.2024.3384759.
32. M. Selvaggio, A.M.G. E, R. Moccia, F. Ficuciello, and B. Siciliano, Haptic-guided shared control for needle grasping optimization in minimally invasive robotic surgery, *in* 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (2019): pp. 3617–3623. https://doi.org/10.1109/IROS40897.2019.8968109.
33. M. Coffey and A. Pierson, Collaborative Teleoperation with Haptic Feedback for Collision-Free Navigation of Ground Robots, *in* 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (2022): pp. 8141–8148. https://doi.org/10.1109/IROS47612.2022.9981426.
34. A.M. Mohammadi and A. Akbarzadeh, A real-time impedance-based singularity and joint-limits avoidance approach for manual guidance of industrial robots, *Advanced Robotics* (2017).
35. E. Shahriari, P. Svarny, S.A.B. Birjandi, M. Hoffmann, and S. Haddadin, Path-Constrained Haptic Motion Guidance via Adaptive Phase-Based Admittance Control, *IEEE Transactions on Robotics* 41 (2025) 1039–1058. https://doi.org/10.1109/TRO.2024.3521861.
36. J. Bimbo, C. Pacchierotti, M. Aggravi, N. Tsagarakis, and D. Prattichizzo, Teleoperation in cluttered environments using wearable haptic feedback, *in* 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (2017): pp. 3401–3408. https://doi.org/10.1109/IROS.2017.8206180.
37. C. Pacchierotti and D. Prattichizzo, Cutaneous/Tactile Haptic Feedback in Robotic Teleoperation: Motivation, Survey, and Perspectives, *IEEE Transactions on Robotics* 40 (2024) 978–998. https://doi.org/10.1109/TRO.2023.3344027.
38. R. Suzuki, A. Karim, T. Xia, H. Hedayati, and N. Marquardt, Augmented Reality and Robotics: A Survey and Taxonomy for AR-enhanced Human-Robot Interaction and Robotic Interfaces, *in* Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, *Association for Computing Machinery*, New York, NY, USA, (2022): pp. 1–33. https://doi.org/10.1145/3491102.3517719.
39. F. Richter, Y. Zhang, Y. Zhi, R.K. Orosco, and M.C. Yip, Augmented Reality Predictive Displays to Help Mitigate the Effects of Delayed Telesurgery, *in* 2019 International Conference on Robotics and Automation (ICRA), (2019): pp. 444–450. https://doi.org/10.1109/ICRA.2019.8794051.

 doi:10.20944/preprints202604.1405.v1

40. J. Funda, R.H. Taylor, B. Eldridge, S. Gomory, and K.G. Gruben, Constrained Cartesian motion control for teleoperated surgical robots, *IEEE Transactions on Robotics and Automation* 12 (3) (1996) 453–465. https://doi.org/10.1109/70.499826.
41. O. Kanoun, F. Lamiraux, and P.-B. Wieber, Kinematic Control of Redundant Manipulators: Generalizing the Task-Priority Framework to Inequality Task, *IEEE Transactions on Robotics* 27 (4) (2011) 785–792. https://doi.org/10.1109/TRO.2011.2142450.
42. A. Escande, N. Mansard, and P.-B. Wieber, Hierarchical quadratic programming: Fast online humanoid-robot motion generation, *The International Journal of Robotics Research* 33 (7) (2014) 1006–1028. https://doi.org/10.1177/0278364914521306.
43. B. Xu and K. Sreenath, Safe Teleoperation of Dynamic UAVs Through Control Barrier Functions, *in* 2018 IEEE International Conference on Robotics and Automation (ICRA), (2018): pp. 7848–7855. https://doi.org/10.1109/ICRA.2018.8463194.
44. R. Periotto, M. Ferizbegovic, F.S. Barbosa, and R.C. Sundin, MPC-CBF with Adaptive Safety Margins for Safety-Critical Teleoperation over Imperfect Network Connections, *in* 2024 European Control Conference (ECC), (2024): pp. 1609–1615. https://doi.org/10.23919/ECC64448.2024.10590767.
45. D. Zhang and R. Tron, Stable Haptic Teleoperation of UAVs via Small L2 Gain and Control Barrier Functions, *in* 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (2021): pp. 8352–8357. https://doi.org/10.1109/IROS51168.2021.9635952.
46. M. Selvaggio, J. Cacace, C. Pacchierotti, F. Ruggiero, and P.R. Giordano, A Shared-Control Teleoperation Architecture for Nonprehensile Object Transportation, *IEEE Transactions on Robotics* 38 (1) (2022) 569–583. https://doi.org/10.1109/TRO.2021.3086773.
47. R.I.C. Muchacho, S. Bien, R. Laha, A. Naceri, L.F.C. Figueredo, and S. Haddadin, Shared Autonomy Control for Slosh-Free Teleoperation, *in* 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (2023): pp. 10676–10683. https://doi.org/10.1109/IROS55552.2023.10342234.
48. R.I.C. Muchacho, R. Laha, L.F.C. Figueredo, and S. Haddadin, A Solution to Slosh-free Robot Trajectory Optimization, *in* 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (2022): pp. 223–230. https://doi.org/10.1109/IROS47612.2022.9981173.
49. X. Fan, Z. Chen, Z. Jiang, F. Ni, and H. Liu, A novel shared teleoperation control approach for nonprehensile object transportation based on orientation feedforward smoothing, *ENGINEERING Mechanical Engineering* 21 (2) (2026) 100880. https://doi.org/10.1007/s11465-026-0880-8.
50. X. Fan, Z. Chen, S. Xin, Y. Ren, Z. Jiang, F. Ni, and H. Liu, Towards Multi-Object Nonprehensile Transportation via Shared Teleoperation: A Framework Based on Virtual Object Model Predictive Control, (2026). https://doi.org/10.48550/arXiv.2604.06932.
51. R.S. Jamisola and R.G. Roberts, A more compact expression of relative Jacobian based on individual manipulator Jacobians, *Robotics and Autonomous Systems* 63 (2015) 158–164. https://doi.org/10.1016/j.robot.2014.08.011.
52. E. Gilbert and D. Johnson, Distance functions and their application to robot path planning in the presence of obstacles, *IEEE Journal on Robotics and Automation* 1 (1) (1985) 21–30. https://doi.org/10.1109/JRA.1985.1087003.
53. O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, *in* 1985 IEEE International Conference on Robotics and Automation Proceedings, (1985): pp. 500–505. https://doi.org/10.1109/ROBOT.1985.1087247.

 doi:10.20944/preprints202604.1405.v1

54. A. Escande, S. Miossec, M. Benallegue, and A. Kheddar, A Strictly Convex Hull for Computing Proximity Distances With Continuous Gradients, *IEEE Transactions on Robotics* 30 (3) (2014) 666–678. https://doi.org/10.1109/TRO.2013.2296332.
55. T. Zhu, J. Mao, L. Han, C. Zhang, and J. Yang, Real-Time Dynamic Obstacle Avoidance for Robot Manipulators Based on Cascaded Nonlinear MPC With Artificial Potential Field, *IEEE Transactions on Industrial Electronics* 71 (7) (2024) 7424–7434. https://doi.org/10.1109/TIE.2023.3306405.
56. X. Fan, X. Shu, B. Tu, C. Liu, F. Ni, and Z. Jiang, A humanoid robot teleoperation approach based on waist–arm coordination, *IR* 50 (5) (2023) 804–813. https://doi.org/10.1108/ir-12-2022-0306.