biomimetics

MDPI

Article

A Reactive Synchronized Motion Controller for Dual-Arm Cooperation with Closed-Chain Constraints

Fengjia Ju, Zijian Wang, Mingda Ge, Hongzhe Jin * and Jie Zhao

School of Mechatronics Engineering, Harbin Institute of Technology, Harbin 150001, China; 20b908029@stu.hit.edu.cn (F.J.); 24b308009@stu.hit.edu.cn (Z.W.); gemingda@outlook.com (M.G.); jzhao@hit.edu.cn (J.Z.)

* Correspondence: hongzhejin@hit.edu.cn

Abstract

When a rigid object is manipulated by dual arms to form a closed chain, the dual-arm motion must satisfy closed-chain constraints. Although synchronized motion can be achieved by strictly tracking predefined global trajectories, the presence of dynamic obstacles necessitates reactive local planning. However, existing local planning methods designed for single-arm manipulators cannot guarantee synchronization between dual arms. To address this limitation, we propose a dual-arm reactive synchronized motion controller (SMC) by incorporating closed-chain constraints on dual-arm slack velocities based on spherical geometric velocity constraints, and by implementing a flexible master-slave arm switching strategy. As a result, the proposed controller achieves synchronized dual-arm control while preserving excellent motion performance, including manipulability enhancement, obstacle avoidance, and compliance with joint angle and velocity constraints. Simulations and experiments on a humanoid upper-body robot validate the effectiveness of the proposed approach.

Keywords: synchronized motion control; closed-chain constraints; dual arms; quadratic programming

1. Introduction

Dual-arm robots offer advantages over single-arm systems, including higher load capacity and improved operational efficiency. During dual-arm collaboration, dual arms may perform independent tasks, such as pouring water into a cup [1], or may jointly manipulate non-rigid objects, such as folding clothes [2]. In these scenarios, closed-chain constraints between dual arms are not required, and each arm can be controlled independently. However, when dual arms manipulate a rigid object—such as a steering wheel [3] or a box—unsynchronized motions that violate closed-chain constraints can generate internal forces between the arms and object, potentially causing damage. Therefore, closed-chain constraints must be considered when handling rigid objects with dual arms.

The global trajectories of the dual-arm end-effectors satisfying closed-chain constraints can be obtained by interpolating the center of the dual-arm end-effectors and subsequently leveraging the relative poses between the end-effectors and the center. In addition, sampling-based methods [4,5], such as PRM, can incorporate global information to achieve collision-free path planning with guarantees on optimality. By enforcing closed-chain constraints on both position and orientation during the sampling process, a fully

Check for updates

Academic Editor: Yi Zhang

Received: 26 March 2026
Revised: 19 April 2026
Accepted: 22 April 2026
Published: 24 April 2026

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

feasible roadmap [6] can be constructed to achieve coordinated manipulation of dual-arm or multi-arm systems through path search and inverse kinematics [7,8].

However, when a robot operates in environments with dynamic obstacles as shown in Figure 1, trajectories generated by a global planner may become infeasible, necessitating the involvement of a local planner [9,10]. A local planner leverages real-time perception to compute short-horizon, collision-avoidance trajectories. Although global optimality cannot be guaranteed, it enables rapid responsiveness. Reactive approaches [7,8] also fall under local planning methods and offer high computational efficiency. Instead of explicitly generating a planned path, reactive methods directly output control commands as inverse kinematic solutions for the manipulators. Hence, such reactive local planners can also be regarded as local controllers. In order to further improve the tracking ability of the control method in disturbed environments, Refs. [11,12] proposed state filters and multilayer neuroadaptive reinforcement learning methods, respectively. However, reactive adjustments made by local controllers cannot inherently ensure the satisfaction of closed-chain constraints.

Therefore, we aim to develop a local controller capable of maintaining synchronized dual-arm motion. NEO [13] is a velocity-level local controller based on quadratic programming, which considers obstacles, joint limits, and manipulability. By introducing slack velocities, NEO allows the arms to deviate from the desired velocity, and the resulting offsets are exploited to enhance the motion performance. Based on this reactive control framework, we design a motion optimization approach that enforces closed-chain constraints at the velocity level. The slack velocities of dual arms are constrained by the spherical geometric velocity constraint. This constraint is incorporated into the original equality constraints of the velocity-level forward-kinematics in the QP formulation, ensuring that the slack velocities of dual arms satisfy closed-chain constraints.

The main contributions of this work are as follows:

- A reactive synchronized dual-arm motion controller is proposed, enabling point-to-point synchronized inverse kinematics over large motion ranges without trajectory interpolation.
- Based on spherical geometric velocity constraints, equality constraints are established for the synchronized desired velocities and the synchronized slack velocities.
- A master-slave arm selection and switching strategy is designed to enhance manipulability and obstacle avoidance while maintaining closed-chain constraints.

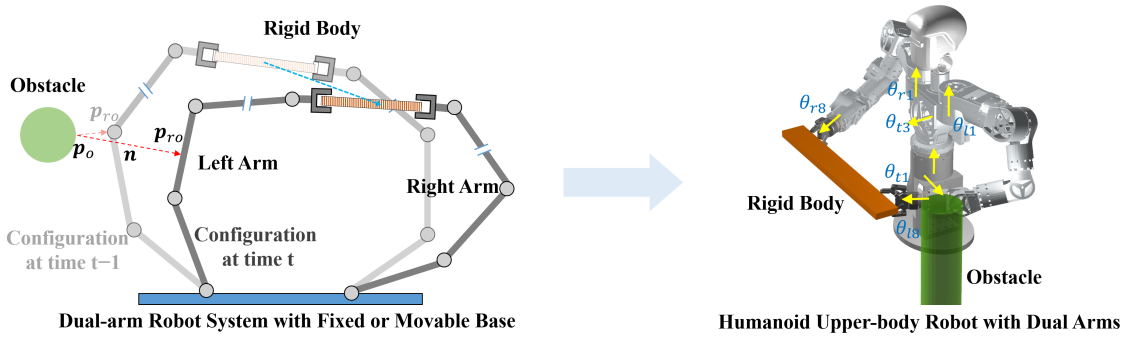


Figure 1. Synchronization motion in the dual-arm system with closed-chain constraints.

2. Related Work

Extensive research has been conducted on synchronized dual-arm motion. Depending on whether force/torque information is incorporated into the control strategy, existing approaches can be broadly classified into dynamic [14–18] or kinematic [6,19–21] motion control methods. Furthermore, based on whether the control commands of dual arms are

computed independently, these methods can be further categorized into distributed and centralized architectures.

Dynamic motion control typically involves interactions between the grasped object and the environment, such as dual-arm peg-in-hole assembly [14] or dual-arm mirror polishing [16]. These methods rely on force/torque sensors to measure internal forces induced by closed-chain constraints, as well as external forces arising from interactions with the environment [17]. In distributed dynamic control architectures [18], each arm independently considers its own tracking error and internal force, whereas centralized dynamic control architectures [15–17] construct synchronization errors based on tracking errors and internal forces of both arms.

In contrast, kinematic motion control aims to avoid the generation of internal forces and is typically designed for transportation tasks [6,19,20] that do not involve interactions between the grasped object and the environment. As a result, force/torque sensors are not required to measure either internal or external forces. Employing kinematic models avoids the complexity of dynamic control or the cost associated with force/torque sensors, while facilitating the integration of additional motion functionalities, such as manipulability enhancement and obstacle avoidance. Similarly, kinematic motion control approaches can also be categorized into distributed and centralized control architectures. Distributed architectures [19,21] perform tracking control based on the individual tracking error of each arm, whereas centralized architectures [20] integrate the tracking errors of dual arms to construct synchronization errors, thereby improving coordination and preventing internal force generation.

However, the aforementioned approaches lack flexibility in local control. The flexibility in local control is challenging because the adjustments of the two arms for motion flexibility and dynamic obstacle avoidance must remain synchronized, while an appropriate master-slave relationship for satisfying synchronization must also be determined. Most existing studies focus on tracking globally synchronized trajectories, often considering disturbances such as visual noise, but lack investigation of synchronized control in environments with dynamic obstacles [22,23]. Furthermore, a master-slave control scheme is introduced for obstacle avoidance in [19]; however, it relies on a fixed assignment of master and slave arms and lacks a dynamic switching strategy. Enhancing the flexibility of local control significantly improves the system's capability to operate in unstructured and complex environments. Without requiring computationally expensive global replanning, the low-level controller can react in real time to resolve local collision risks or singularity-induced deadlocks. Such flexibility also allows the manipulators to deviate from the desired global trajectory for obstacle avoidance, which may temporarily sacrifice the absolute tracking accuracy of the end-effectors. However, this is acceptable for tasks that do not require strict tracking of the global trajectory. Therefore, we adopt a kinematic reactive control perspective and implement slack-velocity control for dual arms under closed-chain constraints within a centralized QP-based solver, and design a master–slave switching strategy. This approach ensures synchronization while simultaneously enhancing motion flexibility and safety through slack velocities.

The rest of this paper is organized as follows: Section 3 introduces our humanoid upper-body robot and extends the single-arm-based NEO method for dual arms. Section 4 details the proposed synchronized motion controller, while Section 5 presents the simulation and experimental results. Finally, conclusions are drawn in Section 6.

3. Problem Description and Modeling Process

3.1. Forward Kinematics of Humanoid Upper-Body Robot

The Denavit–Hartenberg (DH) parameters of the humanoid upper-body robot are shown in Table 1. The humanoid upper-body robot consists of a 3-DOF torso

($\boldsymbol{\theta}_t = [\theta_{t1}, \theta_{t2}, \theta_{t3}]^{\mathrm{T}}$) and two 8-DOF arms (left arm $\boldsymbol{\theta}_l = [\theta_{l1}, \theta_{l2}, \ldots, \theta_{l8}]^{\mathrm{T}}$; right arm $\boldsymbol{\theta}_r = [\theta_{r1}, \theta_{r2}, \ldots, \theta_{r8}]^{\mathrm{T}}$).

Table 1. DH Parameters of humanoid upper-body robot.

Joint	Intial θ_i (°)	d_i (mm)	a_i (mm)	α_i (°)
θ_{t1}	180	−15.5	0	90
θ_{t2}	−90	377	0	90
θ_{t3}	180	34.5	130 | −130	90
$\theta_{l1} \mid \theta_{r1}$	90	144.5	0	90
$\theta_{l2} \mid \theta_{r2}$	90	323.5 | −323.5	0	90
$\theta_{l3} \mid \theta_{r3}$	90	0	0	90
$\theta_{l4} \mid \theta_{r4}$	90	−266.2	0	90
$\theta_{l5} \mid \theta_{r5}$	90	0	0	90
$\theta_{l6} \mid \theta_{r6}$	90	−248.2	0	90
$\theta_{l7} \mid \theta_{r7}$	180	0	0	90
$\theta_{l8} \mid \theta_{r8}$	0	−250	0	0

The forward kinematics of the robot is expressed as

$$\boldsymbol{T}_b^i = \mathrm{F}(\boldsymbol{\theta}) \tag{1}$$

where $\boldsymbol{\theta} = [\boldsymbol{\theta}_t; \boldsymbol{\theta}_l; \boldsymbol{\theta}_r] \in \mathbb{R}^{19}$ denotes the joint angle vector, and $\boldsymbol{T}_b^i \in \mathbb{R}^{4\times4}$ denotes the left or right end-effector pose with respect to the robot base frame, i.e., $\boldsymbol{T}_b^l$ or $\boldsymbol{T}_b^r$.

The corresponding velocity-level relationship of the forward kinematics is given by

$$\dot{\boldsymbol{X}} = \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{2}$$

where $\boldsymbol{J}(\boldsymbol{\theta})$ is formed by stacking the Jacobian matrices [24] of the "Torso–Left Arm" and "Torso–Right Arm" kinematic chains,

$$\boldsymbol{J}(\boldsymbol{\theta}) = \begin{bmatrix} \boldsymbol{J}_t^l & \boldsymbol{J}_l^l & \boldsymbol{0}_{6\times8} \\ \boldsymbol{J}_t^r & \boldsymbol{0}_{6\times8} & \boldsymbol{J}_r^r \end{bmatrix} \in \mathbb{R}^{12\times19} \tag{3}$$

$\boldsymbol{J}_t^l = \partial\mathrm{F}(\boldsymbol{\theta}_t; \boldsymbol{\theta}_l)/\partial\boldsymbol{\theta}_t \in \mathbb{R}^{6\times3}$, $\boldsymbol{J}_t^r = \partial\mathrm{F}(\boldsymbol{\theta}_t; \boldsymbol{\theta}_r)/\partial\boldsymbol{\theta}_t \in \mathbb{R}^{6\times3}$, $\boldsymbol{J}_l^l = \partial\mathrm{F}(\boldsymbol{\theta}_t; \boldsymbol{\theta}_l)/\partial\boldsymbol{\theta}_l \in \mathbb{R}^{6\times8}$, $\boldsymbol{J}_r^r = \partial\mathrm{F}(\boldsymbol{\theta}_t; \boldsymbol{\theta}_r)/\partial\boldsymbol{\theta}_r \in \mathbb{R}^{6\times8}$, $\dot{\boldsymbol{X}} = [\dot{\boldsymbol{X}}_l; \dot{\boldsymbol{X}}_r] \in \mathbb{R}^{12}$ is the actual velocity vector of the left and right end-effectors, and $\dot{\boldsymbol{\theta}} \in \mathbb{R}^{19}$ is the joint angular velocity vector.

The traditional desired velocity combining the velocity forward feedback and the tracking error feedback for a single arm to track a target pose is computed by

$$\boldsymbol{v} = \dot{\boldsymbol{X}}_{tgt} + k_p \cdot \boldsymbol{E} = \dot{\boldsymbol{X}}_{tgt} + k_p \cdot (\boldsymbol{X}_{tgt} - \boldsymbol{X}) \tag{4}$$

where k_p is a gain; $\boldsymbol{E}$ is the tracking error; $\boldsymbol{X}_{tgt}$ and $\dot{\boldsymbol{X}}_{tgt}$ are the pose and velocity of the target, respectively; and $\boldsymbol{X} = [\boldsymbol{X}_l; \boldsymbol{X}_r] \in \mathbb{R}^{12}$ is the current left and right end-effector pose vector in the robot base frame. $\boldsymbol{X}_l = \Phi(\boldsymbol{T}_b^l), \boldsymbol{X}_r = \Phi(\boldsymbol{T}_b^r)$, where $\Phi(\cdot) : \mathbb{R}^{4\times4} \mapsto \mathbb{R}^6$ is a function which converts a homogeneous transformation matrix to a pose vector.

3.2. Rebuilding NEO Method to Adapt to Dual Arms

To apply NEO [13] to the humanoid upper-body robot, the stacked Jacobian matrix $\boldsymbol{J}(\boldsymbol{\theta})$ in (3) is incorporated, and the task-space dimension of the QP problem is expanded from 6 for a single arm to 12 for dual arms. We rebuild NEO to adapt to dual arms as follows:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \mathcal{Q}\boldsymbol{x} + \mathcal{C}^T\boldsymbol{x} \\ \text{s.t.} \quad & \mathcal{J}\boldsymbol{x} = \boldsymbol{v}, \\ & \mathcal{A}\boldsymbol{x} \leq \mathcal{B}, \\ & \boldsymbol{x}^- \leq \boldsymbol{x} \leq \boldsymbol{x}^+ \end{aligned} \tag{5}$$

where

$$\boldsymbol{x} = \begin{bmatrix} \dot{\boldsymbol{\theta}} \\ \boldsymbol{\delta} \end{bmatrix} \in \mathbb{R}^{n+12} \tag{6}$$

$$\mathcal{Q} = \begin{bmatrix} \lambda_\theta \boldsymbol{I}_{n\times n} & \boldsymbol{0}_{6\times 6} & \boldsymbol{0}_{6\times 6} \\ \boldsymbol{0}_{6\times n} & \lambda_{\delta_l}\boldsymbol{I}_{6\times 6} & \boldsymbol{0}_{6\times 6} \\ \boldsymbol{0}_{6\times n} & \boldsymbol{0}_{6\times 6} & \lambda_{\delta_r}\boldsymbol{I}_{6\times 6} \end{bmatrix} \in \mathbb{R}^{(n+12)\times(n+12)} \tag{7}$$

$$\mathcal{J} = \begin{bmatrix} \boldsymbol{J}(\boldsymbol{\theta}) & \boldsymbol{I}_{12\times 12} \end{bmatrix} \in \mathbb{R}^{12\times(n+12)} \tag{8}$$

$$\mathcal{C} = \begin{bmatrix} -\boldsymbol{J}_m \\ \boldsymbol{0}_{12\times 1} \end{bmatrix} \in \mathbb{R}^{n+12} \tag{9}$$

$$\boldsymbol{x}^{-,+} = \begin{bmatrix} \dot{\boldsymbol{\theta}}^{-,+} \\ \boldsymbol{\delta}^{-,+} \end{bmatrix} \in \mathbb{R}^{n+12} \tag{10}$$

Here, n is the number of robot joints, $\boldsymbol{\delta} = [\boldsymbol{\delta}_l; \boldsymbol{\delta}_r] \in \mathbb{R}^{12}$ is the slack velocities of the left and right end-effectors, and $\boldsymbol{J}_m \in \mathbb{R}^n$ is the manipulability gradient calculated by the Jacobian and Hessian matrices [13]. The objective function $f(\boldsymbol{x})$ is derived from $\|\dot{\boldsymbol{\theta}} - \boldsymbol{J}_m)\|_2^2$ [25]. By minimizing $f(\boldsymbol{x})$, the joint velocities calculated via inverse kinematics are driven toward a direction that enhances manipulability.

The manipulability mm is defined by the sum of the manipulabilities [26] of the left and right kinematic chains,

$$mm = mm_l + mm_r = \sqrt{\det([\boldsymbol{J}_t^l \boldsymbol{J}_l^l \boldsymbol{0}_{6\times 8}][\boldsymbol{J}_t^l \boldsymbol{J}_l^l \boldsymbol{0}_{6\times 8}]^T)} + \sqrt{\det([\boldsymbol{J}_t^r \boldsymbol{0}_{6\times 8} \boldsymbol{J}_r^r][\boldsymbol{J}_t^r \boldsymbol{0}_{6\times 8} \boldsymbol{J}_r^r]^T)} \tag{11}$$

where $\det(\cdot)$ denotes the determinant of a matrix. Manipulability reflects the motion flexibility of a robot. A higher manipulability indicates greater flexibility, a larger distance from singular configurations, and reduced variations in joint velocities during motion, thereby enhancing the safety of the system.

In the previous version of NEO, namely the Manipulability Motion Controller (MMC) [27], a joint velocity damper is introduced to avoid angle limits. Accordingly, the matrices $\mathcal{A}$ and $\mathcal{B}$ are designed as

$$\mathcal{A} = \begin{bmatrix} \boldsymbol{I}_{n\times n} & \boldsymbol{0}_{n\times 12} \end{bmatrix} \in \mathbb{R}^{n\times(n+12)} \tag{12}$$

$$\mathcal{B} = \eta \begin{bmatrix} \frac{\rho_1 - \rho_s}{\rho_i - \rho_s} & \cdots & \frac{\rho_n - \rho_s}{\rho_i - \rho_s} \end{bmatrix}^T \in \mathbb{R}^n \tag{13}$$

where η is a gain, ρ is the distance to the nearest joint limit, ρ_s is the minimum allowable distance at which a joint is permitted to approach its limit, and ρ_i is the influence distance at which the velocity damper becomes active.

In NEO [13], to further avoid obstacles, the matrices of the obstacle collision damper, $\mathcal{A}_c$ and $\mathcal{B}_c$, are formulated based on the closest distance d between the robot and obstacles,

$$\mathcal{A}_c = \begin{bmatrix} \boldsymbol{J}_{d1} & \boldsymbol{0}_{1\times(n-k_1)} \\ \vdots & \vdots \\ \boldsymbol{J}_{dw} & \boldsymbol{0}_{1\times(n-k_w)} \end{bmatrix} \in \mathbb{R}^{w\times n} \tag{14}$$

$$\mathcal{B}_c = \begin{bmatrix} \zeta\frac{d_1-d_s}{d_i-d_s} - \boldsymbol{n}_1^T\dot{\boldsymbol{p}}_1 \\ \vdots \\ \zeta\frac{d_w-d_s}{d_i-d_s} - \boldsymbol{n}_w^T\dot{\boldsymbol{p}}_w \end{bmatrix} \in \mathbb{R}^{w} \tag{15}$$

where ζ is a gain, w is the number of obstacles, k_w denotes the index of the robot link corresponding to the closest point to the w-th obstacle, $\dot{\boldsymbol{p}}_w$ is the w-th obstacle's velocity, $\boldsymbol{n}_w$ is the unit vector of closest distance from the w-th obstacle to the robot, $\boldsymbol{J}_{dw} = -\boldsymbol{n}_w{}^T\boldsymbol{J}_{p_{ro}} \in \mathbb{R}^{1\times k_w}$, $\boldsymbol{J}_{p_{ro}}$ is the Jacobian matrix corresponding to the closest distance point on the robot p_{ro}, and d_s and d_i correspond to ρ_s and ρ_i, respectively. To combine the obstacle collision damper with the joint velocity damper, $\mathcal{A}$ and $\mathcal{B}$ are reformulated as

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}_c & \boldsymbol{0}_{w\times 12} \\ \boldsymbol{I}_{n\times n} & \boldsymbol{0}_{n\times 12} \end{bmatrix} \in \mathbb{R}^{(w+n)\times(n+12)} \tag{16}$$

$$\mathcal{B} = \begin{bmatrix} \mathcal{B}_c{}^T & \eta\frac{\rho_1-\rho_s}{\rho_i-\rho_s} & \cdots & \eta\frac{\rho_n-\rho_s}{\rho_i-\rho_s} \end{bmatrix}^T \in \mathbb{R}^{w+n} \tag{17}$$

Substituting the optimization variable vector $\boldsymbol{x} = [\dot{\boldsymbol{\theta}}; \boldsymbol{\delta}]$ into the equality constraint $\mathcal{J}\boldsymbol{x} = \boldsymbol{v}$, the equality constraint (8) in NEO can be equivalently written as

$$\boldsymbol{v} - \boldsymbol{\delta} = \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{18}$$

where $\boldsymbol{\delta}$ is introduced into QP as a part of the variables, allowing the end-effectors to deviate from the desired velocity in order to improve overall motion performance. This is reasonable and efficient for transportation scenarios in which the master requirement is accurate and safe arrival at the target pose rather than strictly tracking a global trajectory. However, if no constraints are imposed on $\boldsymbol{\delta}_l$ and $\boldsymbol{\delta}_r$, the closed-chain constraint may be violated. Therefore, a slack-velocity computation method that explicitly enforces the closed-chain constraint is required, leading to the proposed synchronized motion controller (SMC).

4. Synchronized Motion Controller

To ensure flexible and synchronized dual-arm motion, the SMC is designed from two perspectives, namely synchronized desired velocity and synchronized slack velocity, as shown in Figure 2. The detailed methodology is described as follows.

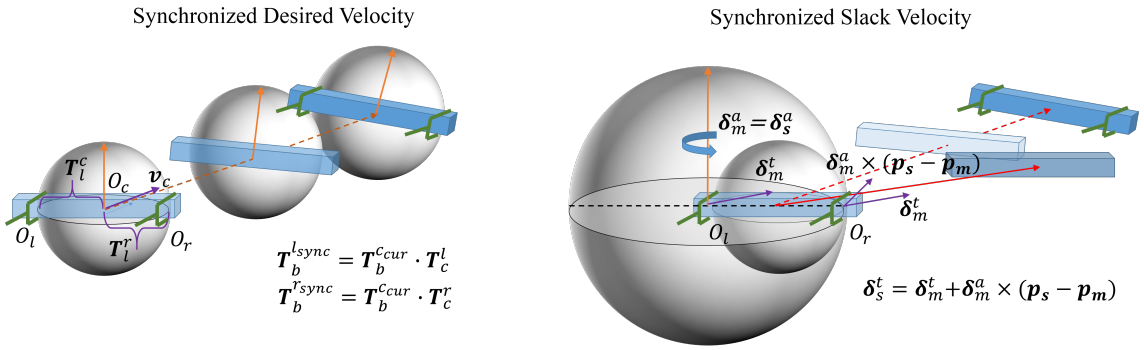


Figure 2. Synchronized desired velocity and synchronized slack velocity based on spherical geometric velocity constraints.

4.1. Synchronized Desired Velocity

For dual-arm cooperative transportation tasks that form a closed chain, the desired end-effector velocities cannot be computed independently using (4), because the resulting translational velocity is constrained to the straight line connecting the current and target object poses. Pure translational motion satisfies the closed-chain constraint; however, motions involving rotation generally violate this constraint.

When the closed-chain constraint is satisfied, the relative pose between two end-effectors (O_l and O_r) ideally remains equal to the initial relative pose. Accordingly, the line segment connecting the two end-effectors can be regarded as a diameter, and the instantaneous motion space of this segment can be represented by a sphere, as illustrated on the left side of Figure 2.

The proper approach is to first compute the desired velocity of the center point O_c of the two end-effectors using (4), and then derive the desired velocities of O_l and O_r based on their fixed relative poses with respect to the center point.

Specifically, first, the current poses of the left and right end-effectors ($\boldsymbol{T}_b^{l_{cur}}$, $\boldsymbol{T}_b^{r_{cur}}$) are obtained by (1). Then, exploiting the symmetry between the two end-effector poses, the current pose of the center point $\boldsymbol{T}_b^{c_{cur}}$ is computed. Next, the desired velocity of the center point $\boldsymbol{v}_c = [v_c^x, v_c^y, v_c^z, v_c^{ox}, v_c^{oy}, v_c^{oz}]^T$ is calculated using (4) based on $\boldsymbol{T}_b^{c_{cur}}$ and the target pose $\boldsymbol{T}_b^{c_{tgt}}$. Finally, the desired velocities of the left and right end-effectors ($\boldsymbol{v}_l = [v_l^x, v_l^y, v_l^z, v_l^{ox}, v_l^{oy}, v_l^{oz}]^T$, $\boldsymbol{v}_r = [v_r^x, v_r^y, v_r^z, v_r^{ox}, v_r^{oy}, v_r^{oz}]^T$) are then derived based on the spherical geometric velocity constraints and the respective initial relative pose relationships between each end-effector and the center point, as follows:

$$\begin{bmatrix} v_l^{ox} \\ v_l^{oy} \\ v_l^{oz} \end{bmatrix} = \begin{bmatrix} v_r^{ox} \\ v_r^{oy} \\ v_r^{oz} \end{bmatrix} = \begin{bmatrix} v_c^{ox} \\ v_c^{oy} \\ v_c^{oz} \end{bmatrix} \tag{19}$$

$$\begin{bmatrix} v_l^{x} \\ v_l^{y} \\ v_l^{z} \end{bmatrix} = \begin{bmatrix} v_c^{x} \\ v_c^{y} \\ v_c^{z} \end{bmatrix} + \begin{bmatrix} v_c^{ox} \\ v_c^{oy} \\ v_c^{oz} \end{bmatrix} \times (\boldsymbol{p}_l - \boldsymbol{p}_c) \tag{20}$$

$$\begin{bmatrix} v_r^{x} \\ v_r^{y} \\ v_r^{z} \end{bmatrix} = \begin{bmatrix} v_c^{x} \\ v_c^{y} \\ v_c^{z} \end{bmatrix} + \begin{bmatrix} v_c^{ox} \\ v_c^{oy} \\ v_c^{oz} \end{bmatrix} \times (\boldsymbol{p}_r - \boldsymbol{p}_c) \tag{21}$$

where $\boldsymbol{p}_l, \boldsymbol{p}_r, \boldsymbol{p}_c \in \mathbb{R}^3$ represents the position components in $\boldsymbol{T}_b^{l_{sync}}$, $\boldsymbol{T}_b^{r_{sync}}$, and $\boldsymbol{T}_b^{c_{cur}}$, respectively. $\boldsymbol{T}_b^{l_{sync}} = \boldsymbol{T}_b^{c_{cur}} \cdot \boldsymbol{T}_c^{l}$ and $\boldsymbol{T}_b^{r_{sync}} = \boldsymbol{T}_b^{c_{cur}} \cdot \boldsymbol{T}_c^{r}$ denote the synchronized poses of the left and right end-effectors, respectively, which satisfy the closed-chain constraints with respect to the current center point $\boldsymbol{T}_b^{c_{cur}}$. $\boldsymbol{T}_c^{l}$ is the initial relative pose between O_c and O_l, while $\boldsymbol{T}_c^{r}$ is the initial relative pose between O_c and O_r. $\times$ denotes the cross product.

Remark 1. *It should be noted that, if the current actual poses of the left and right end-effectors are directly used in (19)–(21), the resulting synchronized desired velocities may violate the closed-chain constraint. This is because both end-effectors exhibit deviations between their current actual poses and the desired synchronized poses, thereby compromising synchronization. Therefore, when computing the synchronized desired velocities in (19)–(21), the poses of the left and right end-effectors are defined relative to the center point in a manner that preserves the synchronization relationship. In this way, the resulting velocities satisfy the closed-chain constraint. However, due to the discrepancy between the desired and actual poses, an additional compensation term must be introduced into the velocity control commands of the left and right end-effectors to reduce this error and improve synchronization.*

To solve this issue, the velocity compensation terms for the left and right end-effectors ($\boldsymbol{v}_{comp}^{l}$, $\boldsymbol{v}_{comp}^{r}$), respectively, based on their tracking errors are introduced:

$$\boldsymbol{v}_{comp}^{l} = k_v \cdot (\Phi(\boldsymbol{T}_b^{l_{sync}}) - \Phi(\boldsymbol{T}_b^{l_{cur}})) \tag{22}$$

$$\boldsymbol{v}_{comp}^{r} = k_v \cdot (\Phi(\boldsymbol{T}_b^{r_{sync}}) - \Phi(\boldsymbol{T}_b^{r_{cur}})) \tag{23}$$

where k_v is a gain.

The synchronized desired velocity $\boldsymbol{v}$ used in (5) is composed of $\boldsymbol{v}_l$, $\boldsymbol{v}_r$, $\boldsymbol{v}_{comp}^{l}$ and $\boldsymbol{v}_{comp}^{r}$,

$$\boldsymbol{v} = [\boldsymbol{v}_l + \boldsymbol{v}_{comp}^{l}; \boldsymbol{v}_r + \boldsymbol{v}_{comp}^{r}] \tag{24}$$

A calculation flowchart of the synchronized desired velocity is shown in Figure 3.

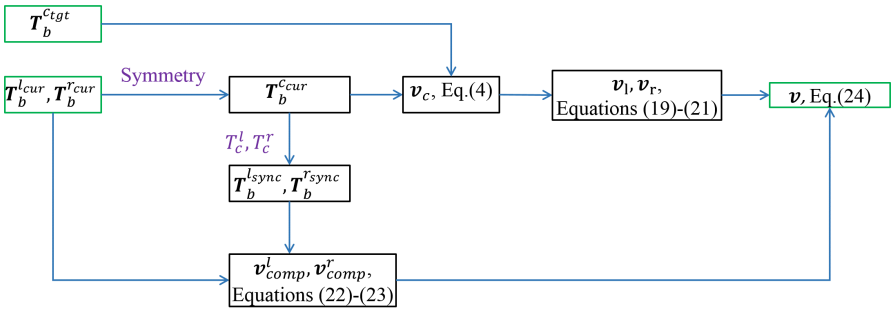


Figure 3. A calculation flowchart of the synchronized desired velocity.

However, an unsynchronized slack velocity δ without constraining the relationship between δ_l and δ_r in (18) can disrupt dual-arm synchronization. Therefore, additional constraints on the slack velocities must be introduced.

4.2. Synchronized Slack Velocity

To address the desynchronization arising from independently computed slack velocities, a master-slave strategy is introduced. The slack velocity of the master arm is unconstrained, while the slave arm adapts its slack velocity to that of the master arm to maintain the closed-chain constraint. The purpose of introducing slack velocity is to allow the end-effector to deviate from the desired velocity, facilitating motion optimizations such as manipulability enhancement and obstacle avoidance. However, the constrained slack velocity of the slave arm may reduce its ability to exploit the flexibility provided by slack velocities, forcing it to passively follow the slack motion of the master arm. Therefore, the selection criterion for the master and slave arms should ensure that the arm with weaker capability for flexible adjustment is assigned as the master arm, while the arm with stronger capability serves as the slave arm.

This capability of flexible adjustment is reflected in the robot's ability to accomplish the master task under different scenarios. The master task is the synchronized target tracking. In obstacle-free environments, the challenge of completing the master task arises from singularity. Higher manipulability indicates that the robot is further from singular configurations. Therefore, the capability of flexible adjustment can be evaluated using manipulability. When manipulability is high, there is less need for slack velocity to adjust the end-effector motion. Even if the slack velocity is constrained by the closed-chain constraint, sufficient flexibility remains to accomplish the master task. Conversely, when manipulability is low, the role of slack velocity becomes more critical.

In environments with obstacles, the challenge of completing the master task is collision. In this case, the capability of flexible adjustment should be evaluated based on the distance

to obstacles. Similarly to the manipulability analysis, a larger distance indicates greater flexibility to accomplish the master task. Accordingly, depending on the scenario, the arm with lower manipulability or a closer distance to obstacles is selected as the master arm, while the other serves as the slave arm. In addition, other criteria can be defined for different task scenarios.

The slack motion of the slave arm is modeled as a rotation about the master arm's end-effector coordinate origin, as shown in the right side of Figure 2. Since the motion is a rotation about a common axis, the angular velocity of any point on the object is identical. Consequently, the slack angular velocities of the master and slave arms are identical and are determined by the slack angular velocity of the master arm. The slack translational velocity of the master arm is computed without enforcing the closed-chain constraint, whereas that of the slave arm is constrained to incorporate both the master arm's translational component and the velocity induced by rotational motion. The detailed closed-chain constraints at the velocity level are shown as follows:

$$\begin{bmatrix} \delta_m^{ox} \\ \delta_m^{oy} \\ \delta_m^{oz} \end{bmatrix} = \begin{bmatrix} \delta_s^{ox} \\ \delta_s^{oy} \\ \delta_s^{oz} \end{bmatrix} \tag{25}$$

$$\begin{bmatrix} \delta_s^{x} \\ \delta_s^{y} \\ \delta_s^{z} \end{bmatrix} = \begin{bmatrix} \delta_m^{x} \\ \delta_m^{y} \\ \delta_m^{z} \end{bmatrix} + \begin{bmatrix} \delta_m^{ox} \\ \delta_m^{oy} \\ \delta_m^{oz} \end{bmatrix} \times (\boldsymbol{p}_s - \boldsymbol{p}_m) \tag{26}$$

where $\boldsymbol{\delta}_m = [\delta_m^t; \delta_m^a] = [\delta_m^x, \delta_m^y, \delta_m^z, \delta_m^{ox}, \delta_m^{oy}, \delta_m^{oz}]^T \in \mathbb{R}^6$ is the slack velocity of the master arm, $\boldsymbol{\delta}_s = [\delta_s^t; \delta_s^a] = [\delta_s^x, \delta_s^y, \delta_s^z, \delta_s^{ox}, \delta_s^{oy}, \delta_s^{oz}]^T \in \mathbb{R}^6$ is the slack velocity of the slave arm, and $\boldsymbol{p}_m, \boldsymbol{p}_s \in \mathbb{R}^3$ represents the position vectors of the master and slave arms' end-effectors. The correspondence between the end-effector positions of the master and slave arms ($\boldsymbol{p}_m, \boldsymbol{p}_s$) and those of the left and right arms ($\boldsymbol{p}_l, \boldsymbol{p}_r$) depends on the designed master-slave switching strategy.

Furthermore, the relationship between the slack velocities of the dual arms in (25) and (26) can be described using a closed-chain constraint matrix $\boldsymbol{M} \in \mathbb{R}^{6\times6}$,

$$\boldsymbol{\delta}_s = \boldsymbol{M} \cdot \boldsymbol{\delta}_m \tag{27}$$

where

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & -\boldsymbol{S}(\boldsymbol{p}_s - \boldsymbol{p}_m) \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \tag{28}$$

$\boldsymbol{S}(\cdot)$ is the skew-symmetric matrix of a vector. By exploiting the relationship between skew-symmetric matrices and the cross-product operation, the cross-product term in (26) is reformulated using a skew-symmetric matrix in (28) for a more compact representation.

Then, by incorporating the closed-chain constraint matrix $\boldsymbol{M}$, $\mathcal{J}$ in (5) and (8) can be rewritten as

$$\mathcal{J} = \begin{cases} \begin{bmatrix} \boldsymbol{J}(\boldsymbol{\theta}) & \boldsymbol{I}_{12\times6} & \boldsymbol{I}_{12\times6} \\ \boldsymbol{0}_{6\times19} & -\boldsymbol{M} & \boldsymbol{I}_{6\times6} \end{bmatrix}, \text{if the left arm is the master arm} \\ \begin{bmatrix} \boldsymbol{J}(\boldsymbol{\theta}) & \boldsymbol{I}_{12\times6} & \boldsymbol{I}_{12\times6} \\ \boldsymbol{0}_{6\times19} & \boldsymbol{I}_{6\times6} & -\boldsymbol{M} \end{bmatrix}, \text{if the right arm is the master arm} \end{cases} \tag{29}$$

The dimension of $\mathcal{J}$ is extended from $12 \times (n+12)$ to $18 \times (n+12)$. Correspondingly, $\boldsymbol{v}$ is extended from a 12-dimensional column vector to an 18-dimensional one by adding $\boldsymbol{0}_{6\times1}$.

The extended equality constraint ($\mathcal{J}\boldsymbol{x} = \boldsymbol{v}$) in (5) simultaneously incorporates the constraints in (18) and (27), enabling the dual-arm end-effectors to possess slack velocities while ensuring that these slack velocities satisfy the synchronization constraints.

Theorem 1. *Under bounded perturbations, the tracking error* $\boldsymbol{E}$ *of the proposed synchronized motion controller (SMC), subject to the constraints in (29), is Uniformly Ultimately Bounded (UUB).*

The proof of Theorem 1 can be found in Appendix A.

4.3. Master–Slave Arm Switching Strategy

Building upon the previously introduced master-slave arm switching strategy driven by manipulability and obstacle avoidance, this subsection presents a precise mathematical formulation of the proposed approach.

We define Δ as the difference between the manipulabilities of dual arms or between their minimum distances to obstacles, depending on the operating scenario, and use it as the decision criterion for master-slave arm switching. In obstacle-free environments, $\Delta = mm_l - mm_r$, and in environments with obstacles, $\Delta = d^l_{min} - d^r_{min}$. mm_l, mm_r, d^l_{min} and d^r_{min} are respectively the manipulabilities and the minimum distances to the obstacles of the left and right arms.

Without the smoothing process, directly applying a hard switching strategy may lead to abrupt changes in the slack velocities due to the variation in closed-chain constraints in the QP problem before and after switching, thereby causing oscillations in the robot motion. To address this issue, we design a dynamically adjustable gain ϵ based on Δ to smoothly regulate the slack-velocity vector. This parameter is incorporated as a weighted factor in the quadratic term of the QP problem, enabling effective control of the slack-velocity vector.

A straightforward approach to ensure smoothness is to increase ϵ during master-slave switching to suppress the slack velocities, driving them close to zero, and then decrease ϵ after the switching is completed. However, this strategy may introduce excessive constraints and reduce the flexibility provided by the slack velocities. Therefore, a more gradual switching scheme is adopted. As indicated by (25) and (26), under the imposed constraints, the slack angular velocities of the master and slave end-effectors remain identical, whereas differences in their slack translational velocities are induced by the angular slack motion.

Since the slack angular velocity influences the slack linear velocity through (26), we indirectly smooth the slack linear velocity by smoothing the slack angular velocity. Consequently, ϵ is applied exclusively to the slack angular velocity to suppress it toward zero. This approach ensures a smooth transition while preserving the flexible adjustment capabilities provided by the slack velocity. Specifically, it enables the slack velocity to manifest as pure translation during the switching phase, and as a combination of translation and rotation during non-switching periods.

To achieve this effect, ϵ should be designed as a smooth function such that it takes a relatively large value when approaching a switching condition (i.e., $\Delta \to 0$) and a smaller value when far from switching (i.e., Δ is away from zero).

Based on the above considerations, various forms of ϵ can be adopted. In this work, we select an exponential function due to its simplicity and ease of tuning, defined as

$$\epsilon = a \cdot \exp\left(b \cdot \mathrm{abs}(\Delta)\right) + 1 \tag{30}$$

where $a \in \mathbb{R}^+, b \in \mathbb{R}^-$. The parameter a controls the degree of suppression on the slack angular velocity: a larger a results in a larger quadratic weight, leading to stronger suppression and thus smaller slack angular velocity, and vice versa. The parameter b

determines the range of influence of ϵ with respect to Δ: a larger magnitude of b leads to a faster variation of the exponential function, concentrating the effect of ϵ around the switching region; conversely, a smaller magnitude extends its influence, which may undesirably affect the flexibility of slack velocity away from switching. To prevent ϵ from becoming too small—thereby causing excessively large slack velocities and degrading tracking performance—a constant term of 1 is added to ensure that $\epsilon > 1$ at all times.

Incorporating ϵ, the weighted factor of the quadratic term in the QP formulation is given as follows:

$$\mathcal{Q} = \begin{bmatrix} \lambda_\theta \boldsymbol{I}_{n\times n} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times n} & \lambda^t_{\delta_l} \boldsymbol{I}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times n} & \mathbf{0}_{3\times 3} & \lambda^a_{\delta_l} \boldsymbol{I}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times n} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \lambda^t_{\delta_r} \boldsymbol{I}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times n} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \lambda^a_{\delta_r} \boldsymbol{I}_{3\times 3} \end{bmatrix} \tag{31}$$

where

$$\begin{aligned} \lambda^t_{\delta_l} &= k_\lambda \cdot \frac{1}{\sum \mathrm{abs}(\boldsymbol{E}_l)}, \ \lambda^t_{\delta_r} = k_\lambda \cdot \frac{1}{\sum \mathrm{abs}(\boldsymbol{E}_r)}, \\ \lambda^a_{\delta_l} &= \epsilon \cdot k_\lambda \cdot \frac{1}{\sum \mathrm{abs}(\boldsymbol{E}_l)}, \ \lambda^a_{\delta_r} = \epsilon \cdot k_\lambda \cdot \frac{1}{\sum \mathrm{abs}(\boldsymbol{E}_r)} \end{aligned} \tag{32}$$

are the weights associated with the translational and angular slack velocities of dual arms, $\boldsymbol{E}_l, \boldsymbol{E}_r \in \mathbb{R}^6$ represents the tracking errors of dual arms, k_λ is the gain, $\mathrm{abs}(\cdot)$ is the element-wise absolute value operator, and $\sum(\cdot)$ is the sum of vector elements.

As stated in [13], a higher λ_θ reduces the ability to maximize manipulability, whereas an excessively low value may lead to extreme joint velocities. Higher $\lambda^t_{\delta_l}, \lambda^t_{\delta_r}, \lambda^a_{\delta_l}, \lambda^a_{\delta_r}$ limits the possible additional manipulability achievable, while a lower value leads to the possibility that the slack will cancel out the desired velocity, leaving large tracking errors. Therefore, the weights are combined with tracking errors to dynamically regulate the magnitude of the slack velocities at different motion stages. In the initial phase, when the tracking error is relatively large, motion performance can be improved by increasing the slack velocity. In the final phase, as the error becomes small, the slack velocity is reduced to ensure tracking accuracy. Moreover, to smooth slack velocity during master-slave switching, ϵ is introduced into $\lambda^a_{\delta_l}, \lambda^a_{\delta_r}$ to suppress the slack angular velocity.

Remark 2. *While this work primarily maximizes manipulability and ensures obstacle avoidance, another crucial aspect in trajectory generation is energy optimization (e.g., minimizing joint movement). In highly constrained dual-arm closed-chain tasks, guaranteeing motion flexibility and collision safety is typically the highest priority, as reaching a singularity or obstacle leads to immediate task failure. Nevertheless, the energy aspect is implicitly partially addressed in our QP formulation. By penalizing the magnitude of the joint velocities (via the weight λ_θ), the solver naturally suppresses excessive joint excursions, acting as a kinematic proxy for minimizing instantaneous kinetic energy.*

The proposed method is summarized in the following pseudocode in Algorithm 1. For the dual-arm synchronized transportation task considered in this work, the initial and target poses of the two end-effectors are determined by the initial and target poses of the grasped object, as well as a task-specific grasping strategy based on the object's geometry. In this work, these quantities are assumed to be given, and their determination is not further discussed.

Algorithm 1: Synchronized Motion Controller

Input: Initial and target poses of the dual-arm end-effectors, $\boldsymbol{T}_b^{l_{init}}$, $\boldsymbol{T}_b^{r_{init}}$, $\boldsymbol{T}_b^{l_{tgt}}$ and $\boldsymbol{T}_b^{r_{tgt}}$; current obstacle pose $\boldsymbol{T}_b^{obs}$.

Output: Joint angle command $\boldsymbol{\theta}_{t_j}$ at time t_j.

Parameter setting: $k_p, k_v, \lambda_\theta, a, b, k_\lambda, \eta, \rho_i, \rho_s, \zeta, d_i, d_s$, final time t_f, time of each control cycle d_t.
Determine initial and target poses of the center point of dual-arm end-effectors, $\boldsymbol{T}_b^{c_{init}}$ and $\boldsymbol{T}_b^{c_{tgt}}$, based on the symmetry between the two end-effector poses.
Calculate the initial relative poses between $\boldsymbol{T}_b^{l_{init}}$ and $\boldsymbol{T}_b^{c_{init}}$, and between $\boldsymbol{T}_b^{r_{init}}$ and $\boldsymbol{T}_b^{c_{init}}$ to obtain $\boldsymbol{T}_c^l$ and $\boldsymbol{T}_c^r$
for $t_j = 0 : t_f$ **do**
    Measure current joint angles $\boldsymbol{\theta}$ of the robot.
    Calculate $\boldsymbol{T}_b^{l_{cur}}$ and $\boldsymbol{T}_b^{r_{cur}}$ based on (1), and $\boldsymbol{T}_b^{c_{cur}}$ based on the symmetry between the two end-effector poses.
    Determine $\boldsymbol{T}_b^{l_{sync}}$ and $\boldsymbol{T}_b^{r_{sync}}$ based on $\boldsymbol{T}_b^{c_{cur}}$, $\boldsymbol{T}_c^l$ and $\boldsymbol{T}_c^r$.
    Calculate $\boldsymbol{J}(\boldsymbol{\theta})$ based on (3), $\boldsymbol{J}_m$.
    Obtain the desired velocity of the center $\boldsymbol{v}_c$ based on $\boldsymbol{T}_b^{c_{cur}}$ and $\boldsymbol{T}_b^{c_{tgt}}$ by (4).
    Calculate synchronized desired velocity (24) based on (19)–(23)
    if *without obstacle* **then**
        Calculate mm_l, mm_r, $\Delta = mm_l - mm_r$.
        if $mm_l < mm_r$ **then**
            Master arm : Left arm. ($\boldsymbol{\delta}_m = \boldsymbol{\delta}_l$, $\boldsymbol{\delta}_s = \boldsymbol{\delta}_r$)
        else
            Master arm : Right arm. ($\boldsymbol{\delta}_m = \boldsymbol{\delta}_r$, $\boldsymbol{\delta}_s = \boldsymbol{\delta}_l$)
        end
        Choose (12), (13) to build SMC.
    else
        Calculate d_{min}^l, d_{min}^r, $\Delta = d_{min}^l - d_{min}^r$.
        if $d_{min}^l < d_{min}^r$ **then**
            Master arm : Left arm. ($\boldsymbol{\delta}_m = \boldsymbol{\delta}_l$, $\boldsymbol{\delta}_s = \boldsymbol{\delta}_r$)
        else
            Master arm : Right arm. ($\boldsymbol{\delta}_m = \boldsymbol{\delta}_r$, $\boldsymbol{\delta}_s = \boldsymbol{\delta}_l$)
        end
        Choose (16), (17) to build SMC.
    end
    Determine tracking errors $\boldsymbol{E}_l$, $\boldsymbol{E}_r$ between $\boldsymbol{T}_b^{l_{cur}}$, $\boldsymbol{T}_b^{r_{cur}}$ and $\boldsymbol{T}_b^{l_{tgt}}$, $\boldsymbol{T}_b^{r_{tgt}}$ based on (4).
    Determine $\mathcal{Q}$ based on (31)–(32).
    Formulate SMC by (5), (6), (9), (10), (12)/(16), (13)/(17), (24), (29), (31).
    Solve the QP problem and obtain $\dot{\boldsymbol{\theta}}$.
    $\boldsymbol{\theta}_{t_j} = \boldsymbol{\theta}_{t_j - 1} + \dot{\boldsymbol{\theta}} \cdot d_t$
end

Remark 3. *Our reactive control approach is local in nature and does not guarantee global optimality. When the robot deviates from the global path to avoid obstacles, it may encounter local minima. In practical applications, if a local minimum is reached, the intervention of a global planner is required for re-planning. Specifically, the global planner is triggered to re-calculate a path whenever the SMC tracking error exceeds a predefined threshold. The newly generated global path then provides updated local waypoints at a specific frequency to refresh the target being tracked by the SMC.*

5. Simulations and Experiments

5.1. Comparison Indexes

The synchronization performance is evaluated using the synchronization position error E_s^p and synchronization orientation error E_s^o. These errors are computed based on the relative pose vector $\boldsymbol{E}_{rel} = [\boldsymbol{E}_{rel}^p, \boldsymbol{E}_{rel}^o]^T \in \mathbb{R}^6$, which is derived from the current poses of the left and right end-effectors, $\boldsymbol{T}_b^{l_{cur}}$ and $\boldsymbol{T}_b^{r_{cur}}$.

$$\boldsymbol{E}_{rel} = \Phi((\boldsymbol{T}_b^{l_{cur}})^{-1} \cdot (\boldsymbol{T}_b^{r_{cur}})) \tag{33}$$

$$E_s^p = \mathrm{N}(\boldsymbol{E}_{rel}^p) - d_0 \tag{34}$$

$$E_s^o = \mathrm{N}(\boldsymbol{E}_{rel}^o) - \theta_0 \tag{35}$$

where $\mathrm{N}(\cdot)$ denotes the 2-norm of a vector, d_0 is the initial distance between the two end-effectors, and θ_0 is the angle required to rotate from one end-effector's orientation to the other around an axis in space at the initial moment.

Additionally, the total of the dual-arm tracking position error E_t^p and the total of the dual-arm tracking orientation error E_t^o are also considered.

$$E_t^p = \mathrm{N}(\boldsymbol{E}_l^p) + \mathrm{N}(\boldsymbol{E}_r^p) \tag{36}$$

$$E_t^o = \mathrm{N}(\boldsymbol{E}_l^o) + \mathrm{N}(\boldsymbol{E}_r^o) \tag{37}$$

where $\boldsymbol{E}_l^p$ and $\boldsymbol{E}_r^p$ are the tracking position error vectors of the left and right arms, and $\boldsymbol{E}_l^o$ and $\boldsymbol{E}_r^o$ are the tracking orientation error vectors of the left and right arms.

The parameters of the proposed SMC used in both simulations and experiments are set as follows: $k_p = \min(1.5 \times (1 - e^{-0.5t_j}), 1.5)$ in (4) to smooth the startup according to the time t_j, $\eta = 1, \rho_i = 15°, \rho_s = 5°$ in (13), $\zeta = 1, d_i = 0.35$ m, $d_s = 0.1$ m in (17), $k_v = 1$ in (22), and (23), $\lambda_\theta = 0.01, k_\lambda = 10, a = 10, b = -100$ in (31), $d_t = 0.01$ s.

5.2. Simulations

In this section, two sets of comparative simulations are conducted: one without obstacles and one with obstacles. To validate the effectiveness of the proposed method in synchronized motion control, we compare the proposed SMC with two baseline methods: the original QP method [25], which strictly tracks the desired velocity without slack variables; the NEO method [13], which introduces slack velocities but lacks closed-chain constraints; and the proposed SMC. All methods use the synchronized desired velocity in (24) for the dual-arm cooperation tasks. In the task setup, only the initial and target poses are specified, without performing trajectory interpolation.

In addition, to demonstrate the capability of SMC to achieve synchronized tracking without trajectory interpolation, we further compare it with the SyncTC method [20]. The SyncTC method is a centralized synchronized tracking control approach that minimizes joint velocities using quadratic programming. It constructs a coupled error by combining tracking error and synchronization error, and further formulates a zeroing neurodynamic model based on the coupled error, which is incorporated into the QP as a closed-chain constraint. The SyncTC method tracks synchronized trajectories that are precomputed using quintic polynomial interpolation.

5.2.1. Simulation A

This simulation is conducted in an obstacle-free environment, where manipulability is used as the criterion for master-slave arm switching. The robot motion trajectories generated by the three methods are shown in Figure 4a.

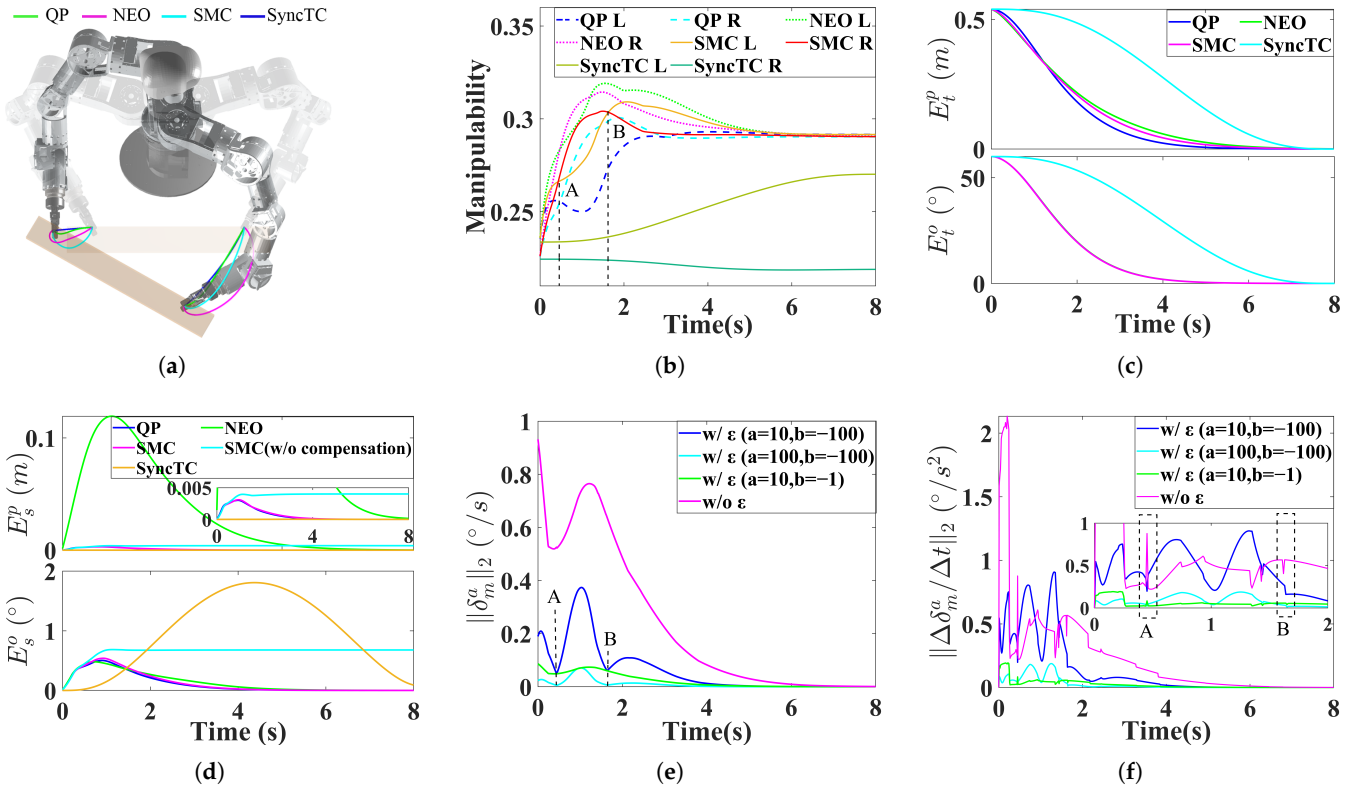


Figure 4. Simulation A. (**a**) Trajectories from a top-down perspective. (**b**) Manipulability. (**c**) Tracking error. (**d**) Synchronization error. (**e**) Slack velocity in SMC. (**f**) Slack acceleration in SMC.

From Figure 4b, the average sums of manipulability of the two arms during motion for the QP, NEO, SMC, and SyncTC methods are 0.571, 0.594, 0.583, and 0.472, respectively. Compared with QP, the NEO and SMC methods, which incorporate slack velocities, improve manipulability by 4.0% and 2.1%, respectively. However, due to the closed-chain constraints imposed in SMC, its manipulability is slightly lower than that of NEO. Compared with SyncTC, which does not include manipulability optimization, SMC improves manipulability by 23.5%.

Figure 4c shows that SMC also achieves high tracking accuracy. At the final time, the sum of the tracking position and orientation errors of the left and right end-effectors under SMC are 0.47 mm and 0.0065°, respectively. However, due to the introduction of slack velocities, the tracking position errors of SMC are larger than those of the QP method (0.15 mm and 0.0066°), and also larger than those of the SyncTC method (0.13 mm and 0.0071°). However, in terms of tracking orientation error, the SMC method outperforms the QP and SyncTC methods.

The synchronization errors shown in Figure 4d indicate that the proposed SMC achieves good synchronization performance, with maximum synchronization position and orientation errors of 3.1 mm and 0.54°, respectively. In contrast, the NEO method without closed-chain constraints results in excessive synchronization position error due to independent slack motions of dual arms, with maximum errors reaching 119.3 mm and 0.48°, respectively. Both the QP method, which strictly tracks synchronized desired velocities, and the SyncTC method, which tracks synchronized trajectories, have smaller synchronization position error, but lack the manipulability benefits provided by slack velocities. Their maximum synchronization position and orientation errors are 3.0 mm and 0.50°, and 0.02 mm and 1.81°, respectively. In terms of synchronization orientation error, the SMC method outperforms the QP and SyncTC methods.

Furthermore, to demonstrate the effect of the compensation velocities v^l_{comp} and v^r_{comp} introduced in the SMC method, we remove these terms for comparison. Without compensation, the synchronization error increases, reaching up to 4.0 mm and 0.69°.

Figure 4e,f present the slack angular velocity and slack angular acceleration in the SMC method. To illustrate the role of the dynamic gain ϵ during master-slave switching, we consider cases without ϵ and with different parameter settings. As shown in Figure 4b, the SMC method (with $a = 10, b = -100$) performs master-slave switching at points A and B. At these moments, the slack angular velocity exhibits a noticeable decrease, while the slack angular acceleration remains relatively smooth. At point A, the acceleration decreases to 0.20 $^\circ/s^2$, with a jerk of 19.6 $^\circ/s^3$. Without ϵ, the slack angular velocity maintains a high magnitude throughout the motion, resulting in a sharp acceleration peak of up to 2.0 $^\circ/s^2$ at the initial stage. At the switching point, the slack angular acceleration reaches 0.88 $^\circ/s^2$, with a jerk of 64.7 $^\circ/s^3$. These results indicate that introducing the dynamic gain ϵ effectively smooths the slack motion. However, the parameter selection of a and b also affects performance. Increasing a from 10 to 100 further smooths the slack motion but excessively suppresses its magnitude, thereby reducing the effectiveness of slack-based adjustment. Changing b from -100 to -1 enlarges the influence range of ϵ, causing the slack velocity to remain suppressed throughout the motion rather than only near switching points.

5.2.2. Simulation B

Static and dynamic obstacles are introduced into the scene, as shown in Figure 5a,b. The spherical obstacle moves at a constant velocity along the Y-direction. The proposed SMC is compared with the QP and NEO methods, both equipped with obstacle avoidance capability. The minimum distance to obstacles is used as the criterion for master-slave arm switching.

As shown in Figure 5c,d, the minimum distances to obstacles achieved by the QP, NEO, and SMC methods are 0.068 m, 0.097 m, and 0.100 m, respectively, while the average sum of manipulability of the two arms during motion is 0.58, 0.61, and 0.61, respectively. Compared with QP, SMC with slack velocities increases the performance in both minimum distance and manipulability by 47% and 5.2%, respectively. Moreover, compared with NEO, SMC does not degrade these metrics, indicating that the introduction of slack velocities compensates for the limitations imposed by closed-chain constraints on motion performance.

From Figure 5e,f, similarly to Simulation A, SMC maintains good tracking accuracy and synchronization, whereas NEO fails to satisfy the closed-chain constraint. By introducing the compensation velocities, the maximum synchronization position and orientation errors are significantly reduced from 4.5 mm and 0.69° to 3.0 mm and 0.51°, corresponding to reductions of 33.3% and 26.1%, respectively.

Figure 5g,h illustrate the slack angular velocity and acceleration in the SMC method. It can be observed that, with dynamic gain regulation, the slack angular velocity becomes smoother. At the switching point, the angular acceleration is 0.55 $^\circ/s^2$ with a jerk of 39.1 $^\circ/s^3$, whereas without ϵ, the angular acceleration reaches 5.6 $^\circ/s^2$ with a jerk of 280.0 $^\circ/s^3$.

It should be noted that, in SMC, the joint velocities of the slave arm are still driven toward the gradient direction of manipulability and constrained by obstacle avoidance inequalities. Therefore, the motion of the slave arm continues to improve its manipulability and increase its distance from obstacles. It can be seen from Simulation B that the synchronization constraints do not hinder the improvement of manipulability or the execution of obstacle avoidance tasks for the slave arm.

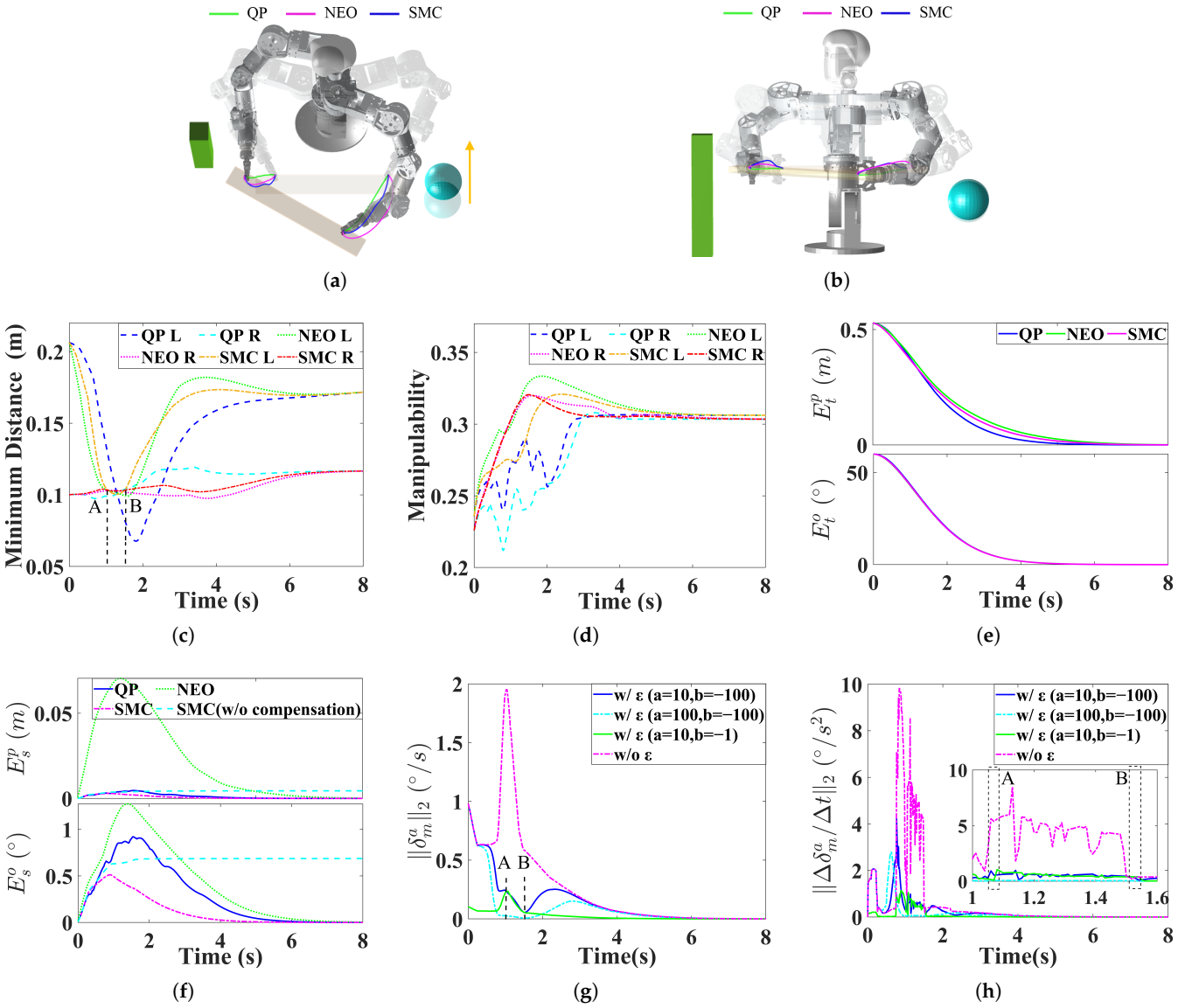


Figure 5. Simulation B. (**a**) Trajectories from a top-down perspective. (**b**) Trajectories from a positive perspective. (**c**) Minimum distance. (**d**) Manipulability. (**e**) Tracking error. (**f**) Synchronization error. (**g**) Slack velocity in SMC. (**h**) Slack acceleration in SMC.

From the above comparisons, it can be observed that SMC enables synchronized dual-arm motion without trajectory interpolation, while improving manipulability and enabling dynamic obstacle avoidance, all while maintaining high synchronization and tracking accuracy. At the same time, each method exhibits its own advantages and limitations. The SyncTC method achieves highly accurate synchronized tracking of a global trajectory and performs well in terms of tracking and synchronization position errors, but suffers from relatively low manipulability. The QP method can also realize synchronized dual-arm motion without trajectory interpolation; however, due to the absence of slack velocities, it cannot generate end-effector deviations, which limits its capability for dynamic obstacle avoidance of end-effectors. The NEO method improves dynamic obstacle avoidance and manipulability by introducing slack velocities, but lacks closed-chain constraints and thus can only achieve coordinated motion without enforcing closed-chain synchronization.

5.3. Experiments

Based on the above simulation results, the NEO method cannot guarantee synchronization between dual arms and is therefore unsuitable for transportation tasks involving closed-chain constraints. Consequently, in the experimental validation, we compare the QP-based method with the proposed SMC. We further designed two experimental scenarios, as shown in Figures 6a and 7a. Experiment A involves a dual-arm transportation task in which a wooden board is carried in an obstacle-free environment, while Experiment B involves transporting a stool in the presence of obstacles, requiring avoidance of the side pillars of a shelf. In both experiments, the tasks are specified solely by a target pose, without requiring trajectory interpolation.

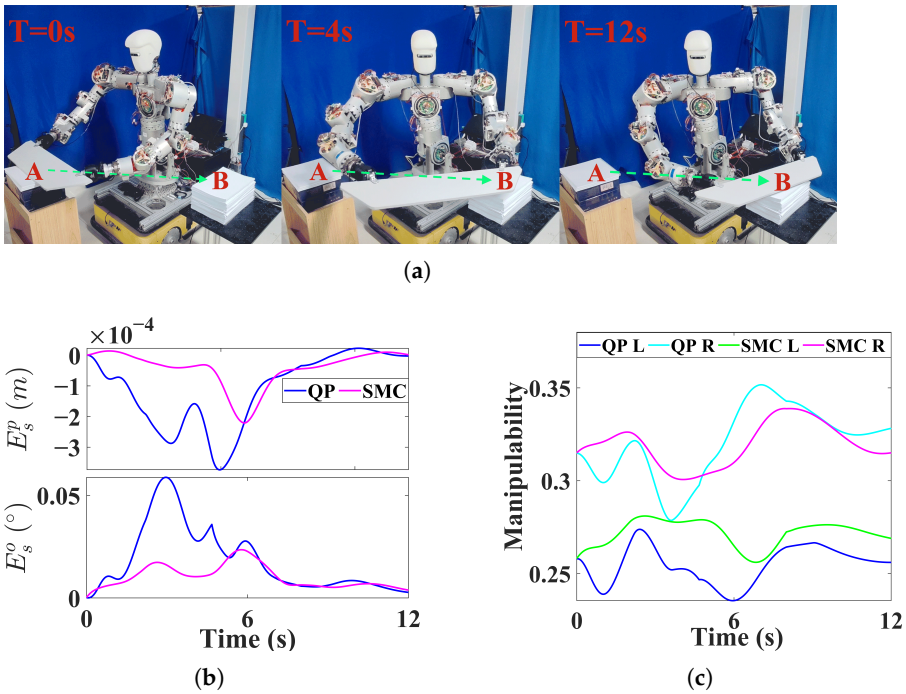


Figure 6. Experiment A. (**a**) The experiment of moving the wooden board. (**b**) Synchronized error. (**c**) Manipulability.

As shown in Figures 6b and 7b, SMC achieves high synchronization performance in both experiments. In Experiment A, as shown in Figure 6c, SMC enables the minimum manipulability of dual arms to remain higher than that achieved by the QP method. This is because, in the SMC method, the arm with lower manipulability is selected as the master arm, and its slack velocity is not constrained. As a result, a more balanced manipulability between the two arms can be maintained, preventing either arm from operating at very low manipulability. In Experiment B, as shown in Figure 7c, since the obstacle lies along the straight line between the initial and target poses, the QP method, lacking slack velocity, cannot generate deviation in the end-effector motion and thus fails to bypass the obstacle. In contrast, the SMC method, with slack velocities incorporated, can generate deviation in motion of the end-effectors and successfully avoids the obstacle.

From the experimental comparisons, it can be observed that the slack velocity with synchronization constraints improves both manipulability and dynamic obstacle avoidance in dual-arm synchronized motion. In addition, the selection criterion in the proposed master-slave switching strategy, where the arm with weaker capability for flexible adjustment is assigned as the master arm, maintains balanced performance between the two arms.

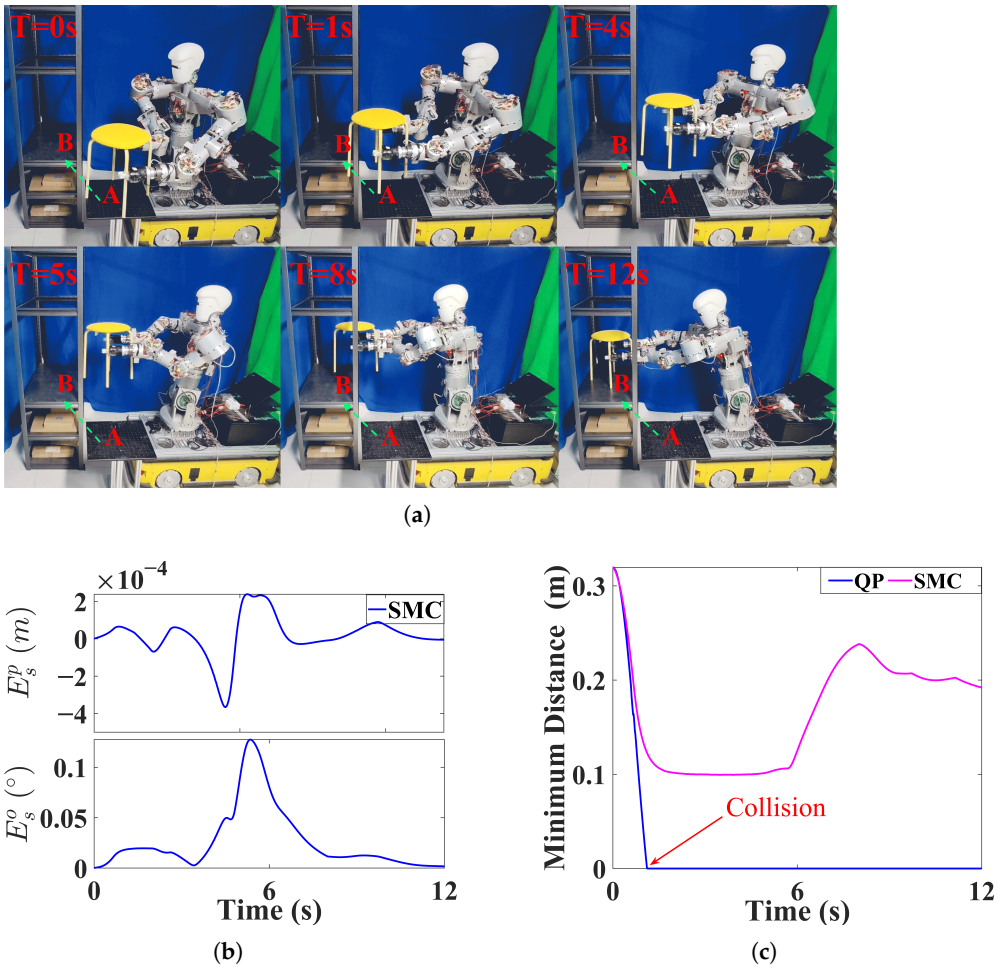


Figure 7. Experiment B. (**a**) The experiment of moving the stool. (**b**) Synchronized error. (**c**) Minimum distance from the obstacle.

6. Conclusions

In this paper, we propose a dual-arm reactive synchronized motion controller that enables synchronized motion without trajectory interpolation. By leveraging the spherical geometric velocity constraints, we derive a synchronized desired velocity and introduce a compensation velocity to further reduce the synchronization error. The compensation velocity can significantly reduce the maximum synchronization position and orientation errors by 33.3% and 26.1%, respectively. Furthermore, we refine the slack-velocity constraints in the NEO framework by incorporating velocity-level closed-chain constraints under a master-slave arm strategy. To ensure smooth master-slave switching, dynamic gains are introduced into the weighting matrix. Extensive simulations and experiments demonstrate that SMC maintains synchronized motion while simultaneously improving manipulability and enabling obstacle avoidance. Compared with QP, SMC with slack velocities can increase the performance in both minimum distance and manipulability by 47% and 5.2%, respectively.

In future work, we plan to apply SMC to mobile humanoid robots for dual-arm mobile manipulation and coordination and further consider practical issues such as object compliance, gripper slippage, and state estimation errors. We will also try to explicitly integrate energy consumption into the unified cost function, achieving multi-objective optimization to balance manipulability and energy consumption.

Author Contributions: Conceptualization, F.J.; methodology, F.J.; software, F.J.; validation, F.J., Z.W. and M.G.; resources, H.J. and J.Z.; data curation, F.J.; writing—original draft preparation, F.J.; writing—review and editing, F.J. and H.J.; supervision, H.J.; project administration, H.J.; funding acquisition, H.J. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China under Grant 62373121, and in part by STI 2030—Major Projects under Grant 2021ZD0201403.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: 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.

Appendix A

Proof. Considering real-world operations in dynamic environments, the system is inevitably subjected to sensor noise, modeling inaccuracies, and external perturbations. Let $\boldsymbol{d}(t) \in \mathbb{R}^{12}$ represent these lumped bounded disturbances, such that $\|\boldsymbol{d}(t)\| \leq \bar{\boldsymbol{d}}$.

Combing the desired velocity defined by the closed-loop law $\boldsymbol{v} = \dot{\boldsymbol{X}}_{tgt} + k_p \boldsymbol{E} = \dot{\boldsymbol{X}}_{tgt} + k_p \cdot (\boldsymbol{X}_{tgt} - \boldsymbol{X})$ and the unfolding form $\boldsymbol{v} - \boldsymbol{\delta} = \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$ of the equality constraint $\mathcal{J}\boldsymbol{x} = \boldsymbol{v}$, we can obtain

$$\dot{\boldsymbol{X}}_{tgt} + k_p \boldsymbol{E} - \boldsymbol{\delta} = \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} + \boldsymbol{d}(t) \tag{A1}$$

According to the actual end-effector velocity $\dot{\boldsymbol{X}} = \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$, the error dynamics of the system is formulated:

$$\dot{\boldsymbol{X}}_{tgt} - \dot{\boldsymbol{X}} + k_p \boldsymbol{E} - \boldsymbol{\delta} = \boldsymbol{d}(t) \implies \dot{\boldsymbol{E}} = -k_p \boldsymbol{E} + \boldsymbol{\delta} + \boldsymbol{d}(t) \tag{A2}$$

Furthermore, incorporating the closed-chain constraint $\boldsymbol{\delta}_s = \boldsymbol{M} \cdot \boldsymbol{\delta}_m$ for the slack velocities of the master and slave end-effectors yields the following error dynamics:

$$\begin{bmatrix} \dot{\boldsymbol{E}}_m \\ \dot{\boldsymbol{E}}_s \end{bmatrix} = -k_p \begin{bmatrix} \boldsymbol{E}_m \\ \boldsymbol{E}_s \end{bmatrix} + \begin{bmatrix} \boldsymbol{I}_{6\times 6} \\ \boldsymbol{M} \end{bmatrix} \boldsymbol{\delta}_m + \begin{bmatrix} \boldsymbol{d}_m(t) \\ \boldsymbol{d}_s(t) \end{bmatrix} \tag{A3}$$

where $\boldsymbol{E}_m$ and $\boldsymbol{E}_s$ are the tracking error vectors of the master and slave end-effectors, respectively. $\boldsymbol{d}_m(t)$ and $\boldsymbol{d}_s(t)$ are the lumped bounded disturbances of the master and slave end-effectors, respectively.

Define a following Lyapunov candidate function:

$$V = \frac{1}{2}\boldsymbol{E}^T \boldsymbol{E} \tag{A4}$$

Taking its time derivative and substituting (A2) yields:

$$\dot{V} = \boldsymbol{E}^T(-k_p \boldsymbol{E} + \boldsymbol{h} + \boldsymbol{d}(\boldsymbol{t})) \tag{A5}$$

where $\boldsymbol{h} = \begin{bmatrix} \boldsymbol{I}_{6\times 6} \\ \boldsymbol{M} \end{bmatrix} \boldsymbol{\delta}_m$.

Applying the Cauchy–Schwarz inequality, we obtain:

$$\dot{V} \leq -k_p \|\boldsymbol{E}\|^2 + \|\boldsymbol{E}\|(\|\boldsymbol{h}\| + \bar{\boldsymbol{d}}) \tag{A6}$$

To ensure a strict decrease in the Lyapunov function ($\dot{V} < 0$), the tracking error must satisfy the following condition:

$$\|\boldsymbol{E}\| > \frac{\|\boldsymbol{h}\| + \bar{\boldsymbol{d}}}{k_p} \tag{A7}$$

Since the slack variable $\boldsymbol{\delta}_m$ is an optimization variable tightly constrained by the physical limits of the robot joints and the feasible region of the task space, its mapped term $\boldsymbol{h}$ is inherently bounded, implying the existence of a maximum norm $\|\boldsymbol{h}\|_{\max}$. Consequently, whenever the error norm exceeds the threshold $(\|\boldsymbol{h}\|_{\max} + \bar{\boldsymbol{d}})/k_p$, the Lyapunov derivative is strictly negative. This mathematically guarantees that the tracking error $\boldsymbol{E}$ will consistently converge toward, and ultimately remain confined within, the bounded compact set:

$$\Omega = \left\{ \boldsymbol{E} \mid \|\boldsymbol{E}\| \leq \frac{\|\boldsymbol{h}\|_{\max} + \bar{\boldsymbol{d}}}{k_p} \right\} \tag{A8}$$

Therefore, our proposed SMC guarantees Uniformly Ultimately Bounded (UUB) stability, ensuring reliable tracking performance even under the enforced closed-chain constraints and external perturbations. □

Appendix B

Table A1. A summary table of main symbols.

Symbol	Description
$\boldsymbol{\theta}_t, \boldsymbol{\theta}_l, \boldsymbol{\theta}_r$	Joint angle vector of the torso, left and right arms.
$\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}$	Joint angle and velocity vectors of the robot.
O_c, O_l, O_c	Left and right end-effectors, and their center point.
$\boldsymbol{T}_b^{l_{sync}}, \boldsymbol{T}_b^{r_{sync}}$	Homogeneous transformation matrices of the synchronized poses of the left and right end-effectors with respect to $\boldsymbol{T}_b^{c_{cur}}$.
$\boldsymbol{T}_c^l, \boldsymbol{T}_c^r$	Initial relative poses between O_c and O_l, between O_c and O_r.
$\boldsymbol{T}_b^{c_{cur}}, \boldsymbol{T}_b^{l_{cur}}, \boldsymbol{T}_b^{r_{cur}}$	Homogeneous transformation matrices of the current poses of the left and right end-effectors, and their center point.
$\boldsymbol{p}_l, \boldsymbol{p}_r, \boldsymbol{p}_c$	Position components in $\boldsymbol{T}_b^{l_{sync}}$, $\boldsymbol{T}_b^{r_{sync}}$, $\boldsymbol{T}_b^{c_{cur}}$.
$\boldsymbol{p}_m, \boldsymbol{p}_s$	Position vectors of the master and slave arms' end-effectors.
$\boldsymbol{T}_b^{l_{init}}, \boldsymbol{T}_b^{r_{init}}$	Homogeneous transformation matrices of the initial poses of the left and right end-effectors.
$\boldsymbol{T}_b^{l_{tgt}}, \boldsymbol{T}_b^{r_{tgt}}, \boldsymbol{T}_b^{c_{tgt}}$	Homogeneous transformation matrices of the target poses of the left and right end-effectors and their center point.
$\boldsymbol{J}(\boldsymbol{\theta})$	Jacobian matrix by stacking kinematic chains.
$\boldsymbol{v}_l, \boldsymbol{v}_r$	Velocity vectors of the left and right end-effectors.
$\boldsymbol{v}$	Velocity vector of the two end-effectors.
$\boldsymbol{v}_{comp}^l, \boldsymbol{v}_{comp}^r$	Compensation velocity terms for the left and right end-effectors.
$\boldsymbol{\delta}_l, \boldsymbol{\delta}_r$	Slack velocity vectors of the left and right end-effectors.
$\boldsymbol{\delta}_m, \boldsymbol{\delta}_s$	Slack velocity vectors of the master and slave end-effectors.
$\boldsymbol{\delta}$	Slack velocity vector of the two end-effectors.
$\boldsymbol{x}$	Optimization variable vector.
$\boldsymbol{J}_m$	Manipulability gradient.
k_p, k_v, ζ, η	Gains.
$\boldsymbol{M}$	Closed-chain constraint matrix.
mm_l, mm_r	Manipulabilities of the left and right arms.
d_{min}^l, d_{min}^r	Minimum distances between the obstacles and the left and right arms.
$\boldsymbol{E}_l, \boldsymbol{E}_r$	Tracking error vectors of left and right end-effetors between the current and target poses.
$\boldsymbol{E}_l^p, \boldsymbol{E}_r^p, \boldsymbol{E}_l^o, \boldsymbol{E}_r^o$	Tracking position and orientation error vectors of left and right end-effetors.
$\boldsymbol{E}$	Tracking error vector of left and right end-effetors.
E_t^p, E_t^o	Total of dual-arm tracking position, total of dual-arm tracking orientation errors.
$\boldsymbol{E}_{rel}$	Relative pose vector between $\boldsymbol{T}_b^{l_{cur}}$ and $\boldsymbol{T}_b^{r_{cur}}$.
E_s^p, E_s^o	Synchronization position and orientation errors.
$\mathcal{Q}, \mathcal{C}$	Coefficient matrices of quadratic and linear terms in the optimization objective function.
$\mathcal{J}$	Equality constraint matrix.
$\mathcal{A}, \mathcal{B}$	Inequality constraint matrix.
$\mathcal{A}_c, \mathcal{B}_c$	Inequality constraint matrix for the obstacle collision damper.
λ_θ	Weighted factor corresponding to the joint angular velocity in $\mathcal{Q}$.
$\lambda_{\delta_l}^t, \lambda_{\delta_l}^a, \lambda_{\delta_r}^t, \lambda_{\delta_r}^a$	Weighted factor corresponding to the slack velocity in $\mathcal{Q}$.
ϵ	Dynamically adjustable gain for slack angular velocity.
a, b	Parameters in ϵ.
Δ	Difference between the manipulabilities of dual arms or between their minimum distances to obstacles.

References

1. Yin, H.; Jin, H.; Ju, F.; Liu, J.; Zhao, M.; Zhao, J. Kinematic Control of Humanoid Upper Body Robot Using Virtual Flexible Joint Dynamics Primitive and Quasi-Sliding Mode Observer. *IEEE Trans. Ind. Electron.* **2025**, *72*, 11464–11477. [CrossRef]
2. Zhou, C.; Luan, F.; Hu, J.; Meng, S.; Wang, Z.; Dong, Y.; Zhou, Y.; He, B. Learning Efficient Robotic Garment Manipulation with Standardization. In Proceedings of the Forty-Second International Conference on Machine Learning, Vancouver, BC, Canada, 13–19 July 2025.
3. Dong, Y.; Yu, Z.; Chen, X.; Zhu, X.; Wang, C.; Gergondet, P.; Huang, Q. Bimanual Continuous Steering Wheel Turning by a Dual-Arm Robot. *IEEE/ASME Trans. Mechatron.* **2024**, *29*, 1773–1784. [CrossRef]
4. Kingston, Z.; Moll, M.; Kavraki, L.E. Sampling-Based Methods for Motion Planning with Constraints. *Annu. Rev. Control. Robot. Auton. Syst.* **2018**, *1*, 159–185. [CrossRef]
5. Cortés, J.; Siméon, T. Sampling-Based Motion Planning under Kinematic Loop-Closure Constraints. In *Algorithmic Foundations of Robotics VI*; Erdmann, M., Overmars, M., Hsu, D., van der Stappen, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 75–90. [CrossRef]
6. Jang, K.; Baek, J.; Park, S.; Park, J. Motion Planning for Closed-Chain Constraints Based on Probabilistic Roadmap With Improved Connectivity. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 2035–2043. [CrossRef]
7. Deo, A.S.; Walker, I.D. Overview of damped least-squares methods for inverse kinematics of robot manipulators. *J. Intell. Robot. Syst.* **1995**, *14*, 43–68. [CrossRef]
8. Beeson, P.; Ames, B. TRAC-IK: An open-source library for improved solving of generic inverse kinematics. In *Proceedings of the 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), Seoul, Republic of Korea, 3–5 November 2015*; IEEE: New York, NY, USA, 2015; pp. 928–935. [CrossRef]
9. Kalakrishnan, M.; Chitta, S.; Theodorou, E.; Pastor, P.; Schaal, S. STOMP: Stochastic trajectory optimization for motion planning. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011*; IEEE: New York, NY, USA, 2011; pp. 4569–4574. [CrossRef]
10. Schulman, J.; Duan, Y.; Ho, J.; Lee, A.; Awwal, I.; Bradlow, H.; Pan, J.; Patil, S.; Goldberg, K.; Abbeel, P. Motion planning with sequential convex optimization and convex collision checking. *Int. J. Robot. Res.* **2014**, *33*, 1251–1270. [CrossRef]
11. Yang, G. State filtered disturbance rejection control. *Nonlinear Dyn.* **2025**, *113*, 6739–6755. [CrossRef]
12. Yang, G. Multilayer Neuroadaptive Reinforcement Learning of Disturbed Nonlinear Systems via Actor-Critic Mechanism. *IEEE Trans. Autom. Sci. Eng.* **2026**, *23*, 1372–1381. [CrossRef]
13. Haviland, J.; Corke, P. NEO: A Novel Expeditious Optimisation Algorithm for Reactive Motion Control of Manipulators. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1043–1050. [CrossRef]
14. Krüger, J.; Schreck, G.; Surdilovic, D. Dual arm robot for flexible and cooperative assembly. *CIRP Ann.* **2011**, *60*, 5–8. [CrossRef]
15. Sun, D.; Mills, J. Adaptive synchronized control for coordination of multirobot assembly tasks. *IEEE Trans. Robot. Autom.* **2002**, *18*, 498–510. [CrossRef]
16. Zhang, Y.; Zhao, X.; Tao, B.; Ding, H. Multi-Objective Synchronization Control for Dual-Robot Interactive Cooperation Using Nonlinear Model Predictive Policy. *IEEE Trans. Ind. Electron.* **2023**, *70*, 582–593. [CrossRef]
17. Bjerkeng, M.; Schrimpf, J.; Myhre, T.; Pettersen, K.Y. Fast dual-arm manipulation using variable admittance control: Implementation and experimental results. In *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014*; IEEE: New York, NY, USA, 2014; pp. 4728–4734. [CrossRef]
18. Jean, J.H.; Fu, L.C. An adaptive control scheme for coordinated multimanipulator systems. *IEEE Trans. Robot. Autom.* **1993**, *9*, 226–231. [CrossRef]
19. Cui, Y.; Xu, Z.; Zhong, L.; Xu, P.; Shen, Y.; Tang, Q. A Task-Adaptive Deep Reinforcement Learning Framework for Dual-Arm Robot Manipulation. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 466–479. [CrossRef]
20. Yu, P.; Tan, N.; Gu, X. A Hybrid Neurodynamic Scheme for Bimanual Synchronized Tracking Control of Robotic Manipulators With Uncertain Kinematics. *IEEE Trans. Ind. Inform.* **2025**, *21*, 2967–2976. [CrossRef]
21. Li, S.; Chen, S.; Liu, B.; Li, Y.; Liang, Y. Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks. *Neurocomputing* **2012**, *91*, 1–10. [CrossRef]
22. Zhang, Z.; Yang, Y.; Zuo, W.; Song, G.; Song, A.; Shi, Y. Image-Based Visual Servoing for Enhanced Cooperation of Dual-Arm Manipulation. *IEEE Robot. Autom. Lett.* **2025**, *10*, 3374–3381. [CrossRef]
23. Zhang, X.; Yu, Y.; Cang, N.; Guo, D.; Li, S.; Zhang, W.; Zheng, J. Adaptive Noise Rejection Strategy for Cooperative Motion Control of Dual-Arm Robots. *IEEE Robot. Autom. Lett.* **2025**, *10*, 868–874. [CrossRef]
24. Siciliano, B.; Khatib, O. *Springer Handbook of Robotics*, 2nd ed.; Springer Publishing Company: Berlin/Heidelberg, Germany, 2016.
25. Zhang, Y.; Yan, X.; Chen, D.; Li, S.; Guo, D. QP-based refined manipulability-maximizing scheme for coordinated motion planning and control of physically constrained wheeled mobile redundant manipulators. *Nonlinear Dyn.* **2016**, *85*, 245–261. [CrossRef]

26. Yoshikawa, T. Dynamic manipulability of robot manipulators. In *Proceedings of the 1985 IEEE International Conference on Robotics and Automation, St. Louis, MI, USA, 25–28 March 1985*; IEEE: New York, NY, USA, 1985; Volume 2, pp. 1033–1038. [CrossRef]
27. Haviland, J.; Corke, P. A Purely-Reactive Manipulability-Maximising Motion Controller. *arXiv* **2020**, arXiv:2002.11901. [CrossRef]

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