

Article

Real-Time Kinematically Synchronous Planning for Cooperative Manipulation of Multi-Arms Robot Using the Self-Organizing Competitive Neural Network

Hui Zhang ¹, Hongzhe Jin ^{2,*} , Mingda Ge ² and Jie Zhao ²¹ Institute of Robotics, Henan University of Technology, Zhengzhou 450001, China; huizh2021@haut.edu.cn² State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China; 19b908045@stu.hit.edu.cn (M.G.); jzhao@hit.edu.cn (J.Z.)

* Correspondence: hongzhejin@hit.edu.cn

Abstract: This paper presents a real-time kinematically synchronous planning method for the collaborative manipulation of a multi-arms robot with physical coupling based on the self-organizing competitive neural network. This method defines the sub-bases for the configuration of multi-arms to obtain the Jacobian matrix of common degrees of freedom so that the sub-base motion converges along the direction for the total pose error of the end-effectors (EEs). Such a consideration ensures the uniformity of the EE motion before the error converges completely and contributes to the collaborative manipulation of multi-arms. An unsupervised competitive neural network model is raised to adaptively increase the convergence ratio of multi-arms via the online learning of the rules of the inner star. Then, combining with the defined sub-bases, the synchronous planning method is established to achieve the synchronous movement of multi-arms robot rapidly for collaborative manipulation. Theory analysis proves the stability of the multi-arms system via the Lyapunov theory. Various simulations and experiments demonstrate that the proposed kinematically synchronous planning method is feasible and applicable to different symmetric and asymmetric cooperative manipulation tasks for a multi-arms system.



Citation: Zhang, H.; Jin, H.; Ge, M.; Zhao, J. Real-Time Kinematically Synchronous Planning for Cooperative Manipulation of Multi-Arms Robot Using the Self-Organizing Competitive Neural Network. *Sensors* **2023**, *23*, 5120. <https://doi.org/10.3390/s23115120>

Academic Editors: Yashar Javadi and Carmelo Mineo

Received: 17 April 2023

Revised: 24 May 2023

Accepted: 25 May 2023

Published: 27 May 2023



Copyright: © 2023 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 (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: multi-arms robot; collaborative manipulation; self-organizing competitive neural network; inner star rule; synchronous planning

1. Introduction

The development of artificial intelligence technology has facilitated the research on the autonomous manipulation of robot manipulators. Meanwhile, the increase in requirement of using robots to replace human hands' manipulation has made the cooperative motion of the robot important in the autonomous operations (e.g., manipulating the rudder, using pliers or a wrench, carrying large objects, or other similar manual tasks in daily life) [1]. Various multi-arm robots, such as Baxter [2], YUMI [3], Justin [4], and Robonaut [5], have been proposed to satisfy the requirement because of their outstanding capability of cooperative manipulation in replacing humans.

Multi-arm robots can not only complete the manipulation task of single-arm robot (or robot manipulator) but also accomplish more complex cooperative manipulation task, which is attributed to their large workspace, more degrees of freedom (DOFs), and greater flexibility. The cooperative manipulation of robots has been studied a lot. The methods for the cooperative manipulation could be divided into force-based [6–9] and kinematics-based strategies [10–29].

The force-based strategies include two types: One relies on dynamics control, but the highly complicated nonlinear dynamic model makes it difficult to apply in robot system. The other one depends on position compensation control that adjusts the position of the end-effector (EE) to maintain a certain interaction force and that has been studied

extensively. The hybrid control of force and position [6,7] and the impedance control [8,9] are the typical method using position compensation control. The hybrid control of force and position needs to accurately decompose the coordinate space of the EE into the position space and force space. The impedance control changes the contact force and the EE position according to the impedance value, but the impedance parameters must be adjusted adaptively to ensure the compliance of interaction in the cooperative manipulation. Although the variable impedance is proposed to improve the flexibility of interaction, the real impedance parameters are difficult to obtain. The intelligent control methods are proposed to simplify the control schemes and reduce the difficulty of modeling the robot control system caused by the strong coupling, time-varying, and uncertainty, such as a neural network [10]. In addition, the force-based strategies rely on force/torque sensors and the corresponding control algorithms and are mainly used in the non-redundant robot system [11].

The kinematics-based strategies are simple and easy to combine with an intelligent algorithm to realize robot autonomous manipulation, thereby resulting in the kinematics-based strategies becoming a research focus of the cooperative manipulation. Thus, the kinematics-based cooperative manipulation is mainly considered in this paper. Usually, the higher requirement for the motion synchronization among the arms needs to be satisfied in the kinematics-based cooperative manipulation. The existing kinematics-based strategies for the cooperative manipulation rely on motion planning, including leader-follower [12,13], cooperative-task space (CTS) [14–19], task-oriented [20,21], and intelligent approaches [11,22,23].

The leader-follower approach defines the leader arm and the follower arm for the robot system, and the follower carries out motion planning according to the movement of the leader [12,13]. In the CTS approach, robot arms without a leader-follower relationship are shared equally to achieve symmetrical cooperative manipulation tasks by defining the relative and absolute motions. The extended CTS approach was further proposed to accomplish asymmetric behavior and uncoordinated tasks [18,19]. The task-equation-based approach uses the general formula of the defined cooperative task to transform the coordinate system kinematics among arms to plan the motion of single arm [20,21]. The intelligent approach considers the collaborative task of arms as the constrained quadratic programming problems and utilizes the advanced neural network as the solver of the problems to control the arms motion [11,22]. The intelligent approaches could simplify the solving process of inverse kinematics for dual arms, but the adjustment of the EE attitudes is neglected in cooperative tasks. In Ref. [23], the dual-arm path-planning problem was transformed into a multi-objective optimization problem, and a co-evolutionary method with shared workspace was proposed to serve as a solver. A motion planner based on the kinematic model of a dual-arm robot system in [24] was designed to ensure grasping stability and dexterity. The movement under the relative motion frame of EEs was studied in [25] for the problem of two-hand assembly. An asymmetric task-planning method based on the Lyapunov theory was proposed in [26] to solve the problems of designing the control law of absolute motion tasks and updating the distribution of relative tasks among arms. Fractional-order derivative and the uncertain fractional-order differential equations were utilized to predict and correct motion trends, and the rationality of the method is verified by different cases [27]. A state feedback robust controller based on local information was designed to ensure that the states of multiple robots converge to a common motion state [28]. An extended Kalman filter collaborative algorithm based on the error compensation was proposed to reduce the state estimation error of delayed filtering in multi robot systems [29].

The key focus of the kinematics-based strategies planning is ensuring the motion synchronization in collaborative tasks. The motion of one arm is always taken as a reference to plan the motion of other arms in the existing literature. Moreover, the application of the existing approaches is more suitable for the multi-arms robot than those coupled with the common and fixed base. Even if there is an application in the robot with a dynamic

coupling base, the dynamic coupling base is set to be stationary to ensure the cooperation between arms [24], and this is caused by the uncoordinated movement between the EEs of redundant arms. Few studies have been devoted to the kinematics-based planning of the redundant multi-arms with a dynamic coupling base to ensure the cooperation between the arms.

Therefore, on this basis of the previous studies [24,30,31], this paper proposed a novel kinematics-based synchronous planning for collaborative manipulation of the redundant multi-arms with dynamic coupling, which involves the inverse kinematics based on the sub-base method and the self-organizing competitive neural network. The following aspects differ from those in the existing literature.

A class of cooperative manipulation tasks of multi-arm robot described by generalized coordinate transformation matrix are summarized, such as carrying, manipulating the rudder, using a wrench, manipulating pliers, multi-station manipulation, and other similar cooperative manipulation. The configuration branch division of the multi-arm robot based on the sub-base method is proposed to identify each branch of the robot, and the inverse kinematics is calculated based on the damped least square method. Then, the multi-arms robot system can synchronously converge along the reducing direction of the total error. The self-organizing competitive neural network is proposed to promote motion synchronization between multiple arms, and it regards the cooperative movement between arms as the competitive relationship of neurons instead of relying on a defined arm motion as a reference in existing research. The inner star learning rule is used to change the neuron weight, and all neuron weight values are updated to adjust the motion of multi-arms in every instance of competitive learning. Thus, the multi-arm robot motion planning method is formed and realizes the synchronization of the arms' motion state. The stability of the motion-planning algorithm is analyzed by using the Lyapunov theory and the inner star learning rule principle. The feasibility of the proposed method, the synchronization of motion state, and its applicability are demonstrated by dual-arm and three-arm robots with a dynamic base in different symmetric and asymmetric cooperative manipulations.

The remainder of this paper is organized as follows. Section 2 presents a type of cooperative manipulation, the sub-base description, and the Jacobian matrix definition for the multi-arms with physical coupling. Section 3 discusses the real-time kinematically synchronous planning method of collaborative manipulation based on the self-organizing competitive neural network and the stability. Sections 4 and 5 provide the simulation and experimental results in different cases, respectively. Section 6 presents the conclusion.

2. Cooperative Manipulation of Multi-Arms

2.1. A Type of Kinematically Cooperative Manipulation

In Figure 1, a type of cooperative manipulation for multi-arms is considered for this paper, such as carrying, manipulating rudder, using pliers and multi-station manipulation, etc. Figure 2 presents the common features for the kinematically cooperative manipulations that can be concluded as follows:

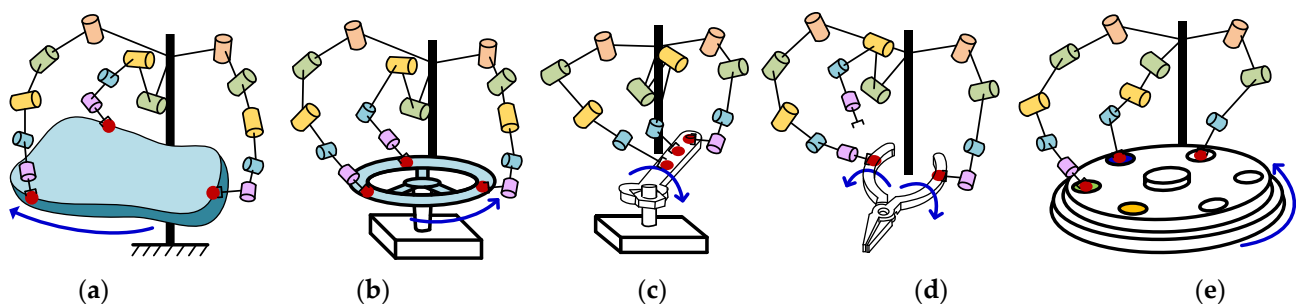


Figure 1. A type of cooperative manipulation. (a) Carrying. (b) Operating rudder. (c) Operating a wrench. (d) Using pliers. (e) Multi-station operation.

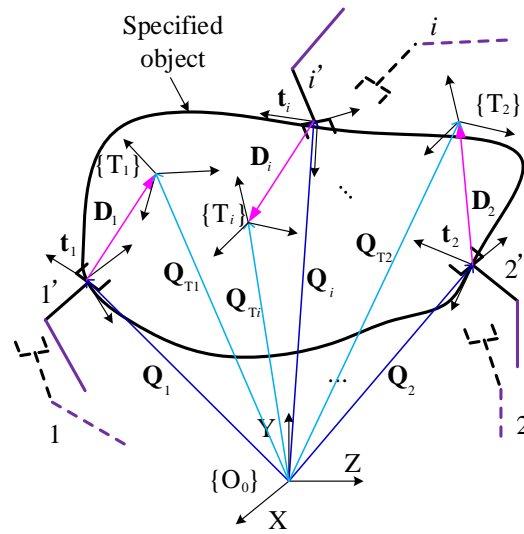


Figure 2. The diagram for the common features in the cooperative manipulation of multi-arms.

(1) The coordinate system $\{T_i\}$ in or out of the specified object is referenced for the object pose, t_i , of each arm EE in real time. The object pose, t_i , is defined as follows:

$$\begin{cases} \mathbf{Q}_i = \mathbf{Q}_{T_i} - {}^{T_i}\mathbf{R}\mathbf{D}_i \\ \mathbf{R}_{t_i}(\hat{\mathbf{f}}_{t_i}, \psi_{t_i}) = {}^{O_0}\mathbf{R}_{T_i} {}^{T_i}\mathbf{R} \\ \mathbf{t}_i = (\mathbf{Q}_i, \hat{\mathbf{f}}_{t_i}, \psi_{t_i})^T \end{cases} \quad (1)$$

where \mathbf{Q}_i is the desired position vector of the i -th EE in coordinate system $\{O_0\}$. \mathbf{Q}_{T_i} is the vector from the coordinate system $\{O_0\}$ to the i -th coordinate system $\{T_i\}$. \mathbf{D}_i is the position vector in coordinate system $\{O_0\}$ for t_i . ${}^{T_i}\mathbf{R}$ denotes the rotation matrix from the coordinate system for t_i to $\{T_i\}$. ${}^{O_0}\mathbf{R}_{T_i}$ refers to the rotation matrix from $\{T_i\}$ to $\{O_0\}$. $\mathbf{R}_{t_i}(\hat{\mathbf{f}}_{t_i}, \psi_{t_i})$ defines

the rotational operator about the axis direction $\hat{\mathbf{f}}_{t_i}$ by ψ_{t_i} radians. $t_i \in \mathbb{R}^{b \times 1}$. $b = 3$ for planar robot. $b = 6$ for spatial robot. $i = 1, 2, \dots, N$. N is the number of arms (or EEs).

(2) When the arms perform collaborative operations, the arms form a closed loop, and there is a certain motion constraint relationship between the EEs. Thus, the motion states of EEs from “1, 2, ..., i ” to “1', 2', ..., i' ” are kinematically consistent and synchronous. The descriptions for movement states are mutual during the execution of these tasks, like the motion error and the motion rate for each EE. Such common motion states of cooperative manipulations are expressed as the following problem in Equation (2), and Equation (2) is used as the judgment criteria for coordinated synchronous motion and is proved in Section 3.2.

$$\begin{cases} \lim_{t \rightarrow \infty} \mathbf{e}_i = \mathbf{t}_i - \mathbf{s}_i = 0 & \text{(a)} \\ \lim_{t \rightarrow \infty} (\|\mathbf{v}_i - \dot{\mathbf{t}}_i\|) = 0 & \text{(b)'} \end{cases} \quad (2)$$

where \mathbf{e}_i denotes the pose error of the i -th EE, \mathbf{v}_i defines the i -th EE velocity, \mathbf{s}_i is the pose of the i -th EE, and t is the time. Equation (2a) refers to the pose errors of EEs along the direction of error convergence, which not only can make the multi-arms reach the execution position of the manipulated object at the same time but also can ensure the minimization of the movement error between the arms in the process of cooperative manipulation. Equation (2b) guarantees that the movement speeds of the EEs are synchronous and that the manipulation speed of the EE is equal/close to the set or constraint value in the manipulation. Finally, the synchronization of arms during cooperative manipulation can be achieved.

The i -th EE pose, velocity, and object pose are \mathbf{s}_i , \mathbf{v}_i and \mathbf{t}_i , respectively. $\mathbf{s}_i, \mathbf{v}_i, \mathbf{t}_i \in \mathbf{R}^{b \times 1}$. The i -th EE pose error and velocity can be calculated as follows:

$$\mathbf{e}_i(T) = \mathbf{t}_i(T) - \mathbf{s}_i(T), \quad (3)$$

$$\mathbf{v}_i(T) = \dot{\mathbf{s}}_i = \frac{\mathbf{s}_i(T) - \mathbf{s}_i(T - \Delta T)}{\Delta T}, \quad (4)$$

where T signifies the current time, $(T - \Delta T)$ is the last sampling time, ΔT denotes the sampling period, $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_k)^T$, and $\mathbf{t} = (\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_k)^T$.

2.2. Multi-Arms Robot with Physical Coupling

This paper considers the general configuration of the multi-arm robot with physical coupling to achieve the cooperative manipulation, as shown in Figure 3. Suppose that there are r joints of the multi-arms and each value of θ_j is the joint angle. The completely joint configuration of the multi-arms is defined as $\Theta = (\theta_1, \dots, \theta_r)^T$, $\Theta \in \mathbf{R}^{r \times 1}$. The pose mapping of the multi-arm robot from joint space to Cartesian space can be expressed as follows:

$$\mathbf{s} = f(\Theta), \quad (5)$$

and $\mathbf{s}_i = f_i(\Theta)$ for the EE.

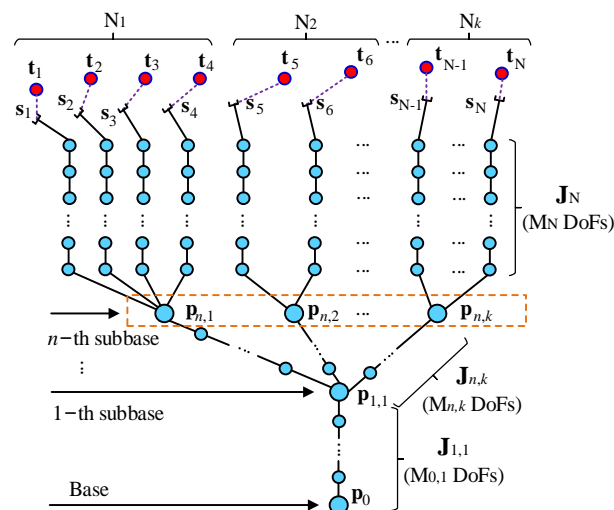


Figure 3. Simple configuration of multi-arm robot.

The corresponding inverse mapping is as follows:

$$\Theta = f^{-1}(\mathbf{s}). \quad (6)$$

where f is a highly nonlinear operator and difficult to solve. The iterative method via Jacobian matrix is used to approach the good solution of the mapping problem.

The traditional Jacobian matrix, \mathbf{J} , is the partial derivative matrix of the whole chain system relative to the EE \mathbf{s} . The Jacobian matrix is obtained via linear approximations of inverse kinematic problems. They linearly simulate the motion of the EE with respect to the instantaneous system changes of the link translation and joint angle. The traditional Jacobian matrix, \mathbf{J} , is a function of the joint angle, Θ , defined as follows:

$$\mathbf{J}(\Theta)_{ij} = \left(\frac{\partial s_i}{\partial \theta_j} \right)_{ij}, \quad (7)$$

where $I = 1, \dots, N$, $j = 1, \dots, r$, $\mathbf{J}(\Theta)_{ij} \in \mathbf{R}^{b \times 1}$.

2.3. Definition of Sub-Bases

The traditional Jacobian matrix can ensure that each EE converges along its own error reduction direction. Unlike the traditional Jacobian matrix, \mathbf{J} , the sub-bases are defined to make the EEs converge along the reducing direction of the system's total error and guarantee that the EEs converge at the same time.

This paper defines that the nodes with multiple branches as the sub-bases for the multi-arms configuration, as shown in Figure 3. The Jacobian matrix is modified as follows:

$$\mathbf{J}(\Theta) = \text{diag}(\mathbf{J}_{1,1}, \mathbf{J}_{n,1}, \dots, \mathbf{J}_{n,k}, \mathbf{J}_1, \dots, \mathbf{J}_N). \quad (8)$$

For the 1-th sub-base pose, $\mathbf{P}_{1,1}$, the corresponding element of the Jacobian matrix is as follows:

$$\mathbf{J}_{1,1}(\Theta)_j = \sum_{i=1}^N \left(\frac{\partial \mathbf{s}_i}{\partial \theta_j} \right)_j, \quad (9)$$

where $\mathbf{J}_{1,1}(\Theta)_j \in \mathbf{R}^{b \times 1}$. $\mathbf{J}_{1,1}(\Theta) \in \mathbf{R}^{b \times M_{0,1}}$. θ_j belongs to the chain $\mathbf{P}_0 \mathbf{P}_{1,1}$ with $M_{0,1}$ DoFs.

For the n,k -th sub-base pose, $\mathbf{P}_{n,k}$, the corresponding element of the Jacobian matrix is as follows:

$$\mathbf{J}_{n,k}(\Theta)_j = \left[\sum_{i=N-N_k+1}^N \left(\frac{\partial \mathbf{s}_i}{\partial \theta_j} + \dots + \frac{\partial \mathbf{s}_N}{\partial \theta_j} \right) \right]_j, \quad (10)$$

where $\mathbf{J}_{n,k}(\Theta)_j \in \mathbf{R}^{b \times 1}$. $\mathbf{J}_{n,k}(\Theta) \in \mathbf{R}^{b \times M_{n,k}}$. θ_j belongs to the chain $\mathbf{P}_{1,1} \mathbf{P}_{n,k}$ with $M_{n,k}$ DoFs. By analogy, the Jacobian matrix corresponding to other sub-bases can be obtained.

For $\mathbf{J}_1, \dots, \mathbf{J}_N$ without common degrees of freedom, the corresponding elements can be obtained according to (6).

$$N = N_1 + N_2 + \dots + N_k, \quad (11)$$

$$r = M_{0,1} + M_{n,1} + \dots + M_{n,k} + M_1 + \dots + M_N. \quad (12)$$

The velocities of 1-th sub-base and n,k -th sub-base are calculated by the following:

$$\dot{\mathbf{P}}_{1,1} = \eta_1 \cdot \frac{1}{N} \cdot \sum_{i=1}^N k_p \cdot \frac{d(\mathbf{t}_i - \mathbf{s}_i)}{dt}, \quad (13)$$

$$\dot{\mathbf{P}}_{n,k} = \eta_{n,k} \cdot \frac{1}{N_k} \cdot \sum_{i=N-N_k+1}^N k_p \cdot \frac{d(\mathbf{t}_i - \mathbf{s}_i)}{dt}, \quad (14)$$

where η_1 and $\eta_{n,k}$ are the gain coefficient. N and N_k are fixed value and related to the configuration of multi-arms.

The inverse kinematics is as follows:

$$\dot{\Theta} = \mathbf{J}^*(\Theta) \dot{\mathbf{S}} = \mathbf{J}^*(\Theta) \begin{pmatrix} \dot{\mathbf{P}}_{1,1} & \dot{\mathbf{P}}_{n,1} & \dots & \dot{\mathbf{P}}_{n,k} & \dot{\mathbf{s}} \end{pmatrix}^T, \quad (15)$$

where \mathbf{J}^* denotes the pseudo-inverse of Jacobian matrix, $\mathbf{J}(\Theta)$, based on the damped least squares method, and $\mathbf{J}^* = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T + \lambda \mathbf{I})^{-1}$. $\mathbf{J}(\Theta) \in \mathbf{R}^{b(k+N+1) \times r}$. $\mathbf{J}^* \in \mathbf{R}^{r \times b(k+N+1)}$. λ ($\lambda > 0$) represents the damping factor that can handle the ill-conditioned \mathbf{J} in the neighborhood of singular configurations for redundant manipulators and guarantee the EEs with the minimum possible deviation at all configurations. \mathbf{I} is a unit matrix with the dimension $b(k+N+1) \times b(k+N+1)$. In accordance with the traditional fixed proportion-based method [31] for the real-time tracking of a given object pose, $\dot{\mathbf{t}}$, the EE velocities are planned as follows:

$$\dot{\mathbf{s}} = \dot{\mathbf{t}} + k_p \cdot (\mathbf{t} - \mathbf{s}), \quad (16)$$

where k_p is the gain coefficient.

2.4. Iteration for Multi-Arms Robot Motion

The iterative method is utilized to achieve the real-time movement of multi-arms via updating the joint angles, Θ , according to (17).

$$\Theta(T) = \Theta(T - \Delta T) + \Delta\Theta. \tag{17}$$

where $\Delta\Theta$ deduced from (15) becomes

$$\begin{aligned} \Delta\Theta &\approx \mathbf{J}^*(\Theta)\Delta\mathbf{S} = \mathbf{J}^*(\Theta)(\Delta\mathbf{P}_{1,1} \ \Delta\mathbf{P}_{n,1} \ \cdots \ \Delta\mathbf{P}_{n,k} \ \Delta\mathbf{s})^T \\ &= \mathbf{J}^*(\Theta) \begin{bmatrix} \eta_1 \cdot \frac{1}{N} \cdot \sum_{i=1}^N \mu \cdot (\mathbf{t}_i - \mathbf{s}_i) \\ \eta_{n,1} \cdot \frac{1}{N_1} \cdot \sum_{i=1}^{N_1} \mu \cdot (\mathbf{t}_i - \mathbf{s}_i) \\ \vdots \\ \eta_{n,k} \cdot \frac{1}{N_k} \cdot \sum_{i=N-N_k+1}^N \mu \cdot (\mathbf{t}_i - \mathbf{s}_i) \\ \Delta\mathbf{t} + \mu \cdot (\mathbf{t} - \mathbf{s}) \end{bmatrix}, \end{aligned} \tag{18}$$

where $\mu = k_p \cdot \Delta T$, and $\mu < 1$. Moreover, $\Delta\mathbf{t}$ represents the changing pose of the object at a sampling time interval, ΔT .

Then, according to the sub-base method, the movement of the multi-arm robot can be achieved. The sub-base motion facilitates the synchronous convergence, and the synchronous performance is more obvious when the common DoFs are enough. The corresponding verifications are presented in Sections 4 and 5.2.

3. Kinematically Synchronous Planning

3.1. Synchronous Planning Using Self-Organizing Competitive Neural Network

The DoFs of the sub-base are not always enough to ensure the system convergence along the reducing error direction for the pose of the EEs, thereby resulting in the asynchronous EE motion. Thus, the self-organizing competitive neural network based on the rule of inner star model is proposed to adjust the synchronism of multi-arms movement. Equation (2) indicates that the error \mathbf{e}_i and the \mathbf{v}_i will tend to a stable value. In accordance with the principle of the self-organizing competitive neural network, the kinematically synchronous planning for the multi-arm robot is designed as shown in Figure 4.

The learning rule of the inner star model defines the weight updating as follows:

$$\Delta w_i = \eta(P_i - w_i)Y_i, \tag{19}$$

where η denotes the learning rate; P_i is the i -th input element of the neuron and the minimum pose velocity error norm, $\min(\|\mathbf{v}_i(T) - \dot{\mathbf{t}}_i\|)$; w_i is the weight value; $i = 1, 2, \dots, N$; N is the number of arms (or EE); and Y_i is the value of output neuron and is defined as

$$Y_i = \begin{cases} 1, & \text{if } P'_i P_i > \varepsilon \\ 0, & \text{Otherwise} \end{cases}, \tag{20}$$

where

$$\varepsilon = \frac{PP}{N}. \tag{21}$$

The input element, P_i , and the weight value, w_i , are defined as

$$P_i = \|\tilde{\mathbf{v}}\| = \min(\|\mathbf{v}_1(T) - \dot{\mathbf{t}}_1\|, \|\mathbf{v}_2(T) - \dot{\mathbf{t}}_2\|, \dots, \|\mathbf{v}_N(T) - \dot{\mathbf{t}}_N\|), \tag{22}$$

$$w_i = \|\mathbf{v}_i(T) - \dot{\mathbf{t}}_i\|. \tag{23}$$

The input vector \mathbf{P} is constituted by P_i and defined as

$$\mathbf{P} = (P_1, P_2, \dots, P_N)^T, \mathbf{P} \in \mathbf{R}^{N \times 1}. \tag{24}$$

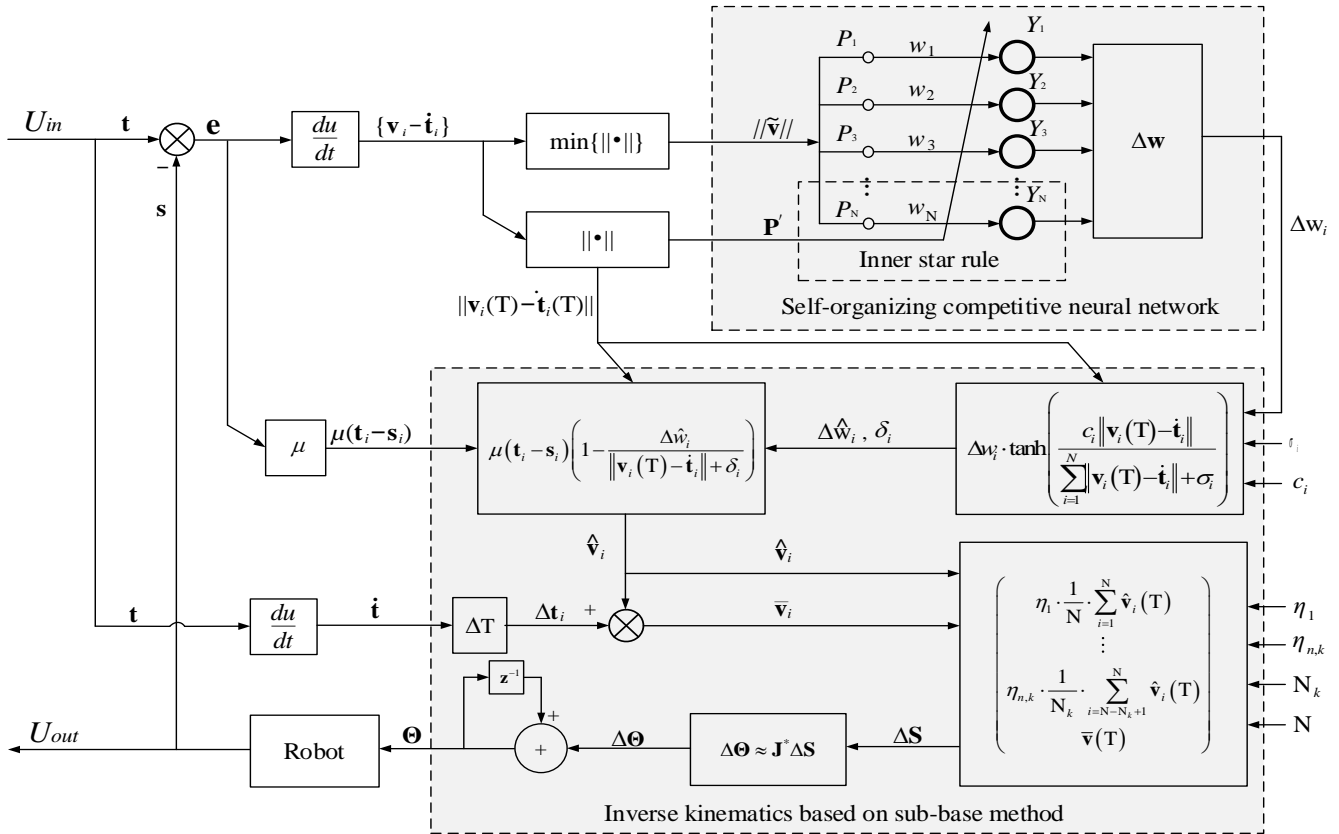


Figure 4. Kinematically synchronous planning for multi-arm robot. $U_{in} = \mathbf{t} = (t_1, t_2, \dots, t_N)^T$. $U_{out} = \mathbf{s} = (s_1, s_2, \dots, s_N)^T$.

Since the minimum of $\|\tilde{\mathbf{v}}\|$ is used as an input and each EE may become the one with the minimum of $\|\tilde{\mathbf{v}}\|$, the proposed method makes the planner no longer use a manipulator as a reference as in the existing literature. In each period, the new input vectors, \mathbf{P}' , in neural networks are defined as

$$\mathbf{P}' = (P'_1, P'_2, \dots, P'_N)^T. \tag{25}$$

where

$$P'_i = \|\mathbf{v}_i(\mathbf{T}) - \dot{\mathbf{t}}_i\|. \tag{26}$$

In order to make the system quickly reach the state of synchronization, the winning weight value will get more rewards in a cycle. The new weight value, $\Delta\hat{w}_i$, is designed as

$$\Delta\hat{w}_i = \Delta w_i \cdot \tanh\left(\frac{c_i \|\mathbf{v}_i(\mathbf{T}) - \dot{\mathbf{t}}_i\|}{\sum_{i=1}^N \|\mathbf{v}_i(\mathbf{T}) - \dot{\mathbf{t}}_i\| + \sigma_i}\right), \tag{27}$$

where σ_i is a small positive real number; and c_i denotes a positive real number to adjust the updating slope of neuron weight and contributes to the synchronization of the EE motion. The response time of converges is shorter when the parameter c_i becomes larger.

The i -th EE's planned related velocity is deduced according to the iterative method as follows:

$$\hat{\mathbf{v}}_i(\mathbf{T}) = \mu \cdot (\mathbf{t}_i - \mathbf{s}_i) \cdot \left[1 - \frac{\Delta \hat{w}_i}{\|\mathbf{v}_i(\mathbf{T}) - \mathbf{t}_i\| + \delta_i} \right]. \quad (28)$$

Then, the planned velocity for EEs is

$$\bar{\mathbf{v}}_i(\mathbf{T}) = \hat{\mathbf{v}}_i(\mathbf{T}) + \Delta \mathbf{t}_i. \quad (29)$$

Thus, (18) becomes

$$\Delta \Theta \approx \mathbf{J}^*(\Theta) \Delta \mathbf{S} = \mathbf{J}^*(\Theta) \begin{bmatrix} \eta_1 \cdot \frac{1}{N} \cdot \sum_{i=1}^N \hat{\mathbf{v}}_i(\mathbf{T}) \\ \eta_{n,1} \cdot \frac{1}{N_1} \cdot \sum_{i=1}^{N_1} \hat{\mathbf{v}}_i(\mathbf{T}) \\ \vdots \\ \eta_{n,k} \cdot \frac{1}{N_k} \cdot \sum_{i=N-N_k+1}^N \hat{\mathbf{v}}_i(\mathbf{T}) \\ \bar{\mathbf{v}}(\mathbf{T}) \end{bmatrix}, \quad (30)$$

where $\bar{\mathbf{v}}(\mathbf{T}) = [\bar{\mathbf{v}}_1(\mathbf{T}), \bar{\mathbf{v}}_2(\mathbf{T}), \dots, \bar{\mathbf{v}}_N(\mathbf{T})]^T$. The learning rate η , η_1 and $\eta_{n,k}$ are related to the step size for each iteration. The response time of converges is shorter when the learning rate becomes larger.

3.2. Stability Analysis

Supposing that the learning rate, i.e., η , η_1 , $\eta_{n,1}$, \dots , $\eta_{n,k}$, is small, based on the rule of inner star model, the weight value with the minimum EE pose error, \mathbf{e}_{\min} , is never updated, and the minimum pose error changes in a cycle is $\Delta \mathbf{e}_{\min}(\mathbf{T})$. Thus, for the EE with the minimum pose error, the Lyapunov function is defined as follows:

$$V(\mathbf{T}) = \frac{1}{2} \mathbf{e}_{\min}^2(\mathbf{T}), \quad (31)$$

$$V(\mathbf{T} + \Delta \mathbf{T}) = \frac{1}{2} \mathbf{e}_{\min}^2(\mathbf{T} + \Delta \mathbf{T}), \quad (32)$$

The change of the pose error is

$$\mathbf{e}_{\min}(\mathbf{T} + \Delta \mathbf{T}) = \mathbf{e}_{\min}(\mathbf{T}) - \Delta \mathbf{e}_{\min}(\mathbf{T}), \quad (33)$$

For the EE with the minimum pose error, $\mathbf{e}_{\min}(\mathbf{T})$, the planned iterative step based on the fixed proportion method [31] is $\mu \mathbf{e}_{\min}(\mathbf{T})$ in each control cycle, as shown in Figure 5. However, the actual motion of EE will have a small deviation, $\boldsymbol{\varphi}$, as follows:

$$\Delta \mathbf{e}_{\min}(\mathbf{T}) = \mu \mathbf{e}_{\min}(\mathbf{T}) + \boldsymbol{\varphi}, \quad (34)$$

where $0 < \mu < 1$.

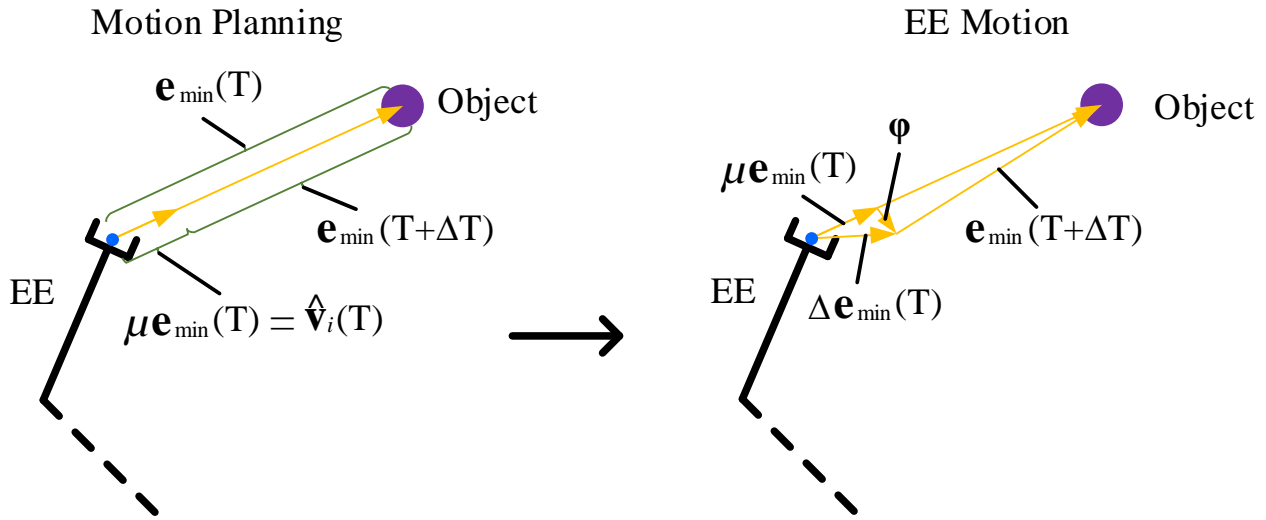


Figure 5. Motion planning and EE motion for the EE with the minimum pose error, $\mathbf{e}_{\min}(T)$. $\hat{\mathbf{v}}_i(T) = \mu \mathbf{e}_{\min}(T)$.

Due to the high-precision motion of the advanced manipulator and the small learning rate in the proposed method, the deviation, $\boldsymbol{\varphi}$, is usually in a very small range. The norm of the deviation $\|\boldsymbol{\varphi}\|$ is less than $\|\mu \mathbf{e}_{\min}(T)\|$ to ensure that the EE can move along the planned path, i.e., $\|\boldsymbol{\varphi}\| < \|\mu \mathbf{e}_{\min}(T)\|$. Therefore,

$$\begin{aligned}
 \Delta V(T) &= \frac{1}{2} \mathbf{e}_{\min}^2(T + \Delta T) - \frac{1}{2} \mathbf{e}_{\min}^2(T) \\
 &= \frac{1}{2} \left\{ [\mathbf{e}_{\min}(T) - \mu \mathbf{e}_{\min}(T) - \boldsymbol{\varphi}]^2 - \mathbf{e}_{\min}^2(T) \right\} \\
 &= \frac{1}{2} \left\{ [(1 - \mu) \mathbf{e}_{\min}(T) - \boldsymbol{\varphi}]^2 - \mathbf{e}_{\min}^2(T) \right\} \\
 &\leq \frac{1}{2} \left\{ [(1 - \mu) \mathbf{e}_{\min}(T) + \|\boldsymbol{\varphi}\|]^2 - \mathbf{e}_{\min}^2(T) \right\} \\
 &< \frac{1}{2} \left\{ [(1 - \mu) \mathbf{e}_{\min}(T) + \mu \mathbf{e}_{\min}(T)]^2 - \mathbf{e}_{\min}^2(T) \right\} \\
 &= \frac{1}{2} \left\{ [(1 - \mu) \mathbf{e}_{\min}(T) + \mu \mathbf{e}_{\min}(T)]^2 - \mathbf{e}_{\min}^2(T) \right\} \\
 &= 0
 \end{aligned} \tag{35}$$

Thus,

$$\lim_{T \rightarrow \infty} \mathbf{e}_{\min}(T) = 0, \tag{36}$$

and the minimum pose error, \mathbf{e}_{\min} , is convergent.

Based on Equations (3) and (4), the derivation of $\dot{\mathbf{e}}_{\min}(T)$ is

$$\begin{aligned}
 \dot{\mathbf{e}}_{\min}(T) &= \dot{\mathbf{t}}_{\min}(T) - \dot{\mathbf{v}}_{\min}(T) = \dot{\mathbf{t}}_{\min}(T) - \dot{\mathbf{s}}_{\min}(T) \\
 &= \frac{\mathbf{t}_{\min}(T) - \mathbf{t}_{\min}(T - \Delta T)}{\Delta T} - \frac{\mathbf{s}_{\min}(T) - \mathbf{s}_{\min}(T - \Delta T)}{\Delta T} \\
 &= \frac{\mathbf{t}_{\min}(T) - \mathbf{s}_{\min}(T)}{\Delta T} - \frac{\mathbf{t}_{\min}(T - \Delta T) - \mathbf{s}_{\min}(T - \Delta T)}{\Delta T} \\
 &= \frac{1}{\Delta T} [\mathbf{e}_{\min}(T) - \mathbf{e}_{\min}(T - \Delta T)] \\
 &= \frac{1}{\Delta T} [\mu \mathbf{e}_{\min}(T - \Delta T) + \boldsymbol{\varphi}]
 \end{aligned}$$

Since $\|\boldsymbol{\varphi}\| < \|\mu \mathbf{e}_{\min}(T)\|$ and $\lim_{T \rightarrow \infty} \mathbf{e}_{\min}(T) = 0$,

$$\lim_{T \rightarrow \infty} \dot{\mathbf{e}}_{\min}(T) = 0, \tag{37}$$

According to the rule of inner star model, the input and the weight ultimately become equal, as follows:

$$P_i = w_i. \tag{38}$$

This is

$$\|\tilde{\mathbf{v}}\| = \|\dot{\mathbf{e}}_{\min}(\mathbf{T})\| = \|\mathbf{v}_i(\mathbf{T}) - \dot{\mathbf{t}}_i\| = 0, \quad (39)$$

$$\Delta\hat{w}_i = \Delta w_i = 0. \quad (40)$$

Then,

$$\lim_{\mathbf{T} \rightarrow \infty} \|\mathbf{e}_i(\mathbf{T})\| = \lim_{\mathbf{T} \rightarrow \infty} \|\mathbf{e}_{\min}(\mathbf{T})\| = 0. \quad (41)$$

$$\lim_{\mathbf{T} \rightarrow \infty} \|\dot{\mathbf{e}}_i(\mathbf{T})\| = \lim_{\mathbf{T} \rightarrow \infty} \|\dot{\mathbf{e}}_{\min}(\mathbf{T})\| = 0. \quad (42)$$

Therefore, the proposed planning method based on the self-organizing competitive neural network is convergent, and Equation (2) is proved to be valid to achieve motion synchronization.

4. Simulation

The three-arm robot with 15-DoFs is utilized to provide the contrast simulations for inverse kinematics by using inverse kinematics based on the sub-bases and the traditional method [31], as shown in Figure 6. The common DoFs are 3 that are enough for all EEs to make the pose error converge along the decreasing direction of the total error simultaneously. The synchronization performances of EE movements are compared by tracking the static-designated location on the specified-carried object. The robot configuration parameters, initial parameters, and kinematics parameters are presented in Tables 1–3. The poses of the static object are $\mathbf{t}_1 = (2.49 \text{ m}, 2.79 \text{ m}, 1.536 \text{ rad})^T$, $\mathbf{t}_2 = (5.0 \text{ m}, -0.70 \text{ m}, 1.536 \text{ rad})^T$, and $\mathbf{t}_3 = (4.49 \text{ m}, 1.50 \text{ m}, 1.536 \text{ rad})^T$. Moreover, $\mathbf{t} = (\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3)^T$, and $\dot{\mathbf{t}} = \mathbf{0}$. The simulation results are presented in Figures 7 and 8.

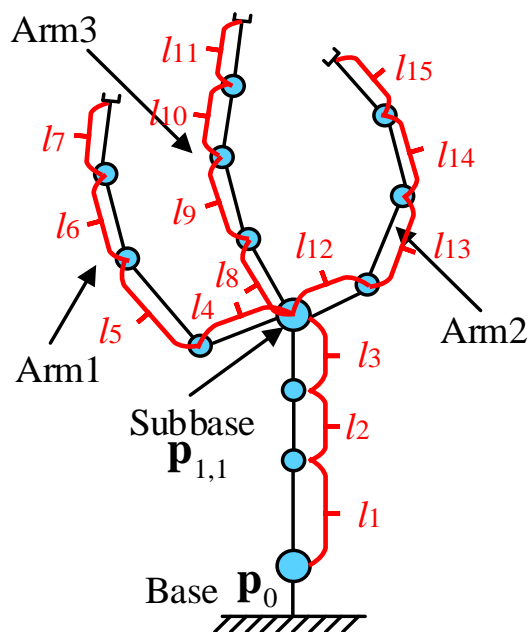


Figure 6. The configuration of three-arm robot with 15-DoFs.

Table 1. Parameters of three-arm robot.

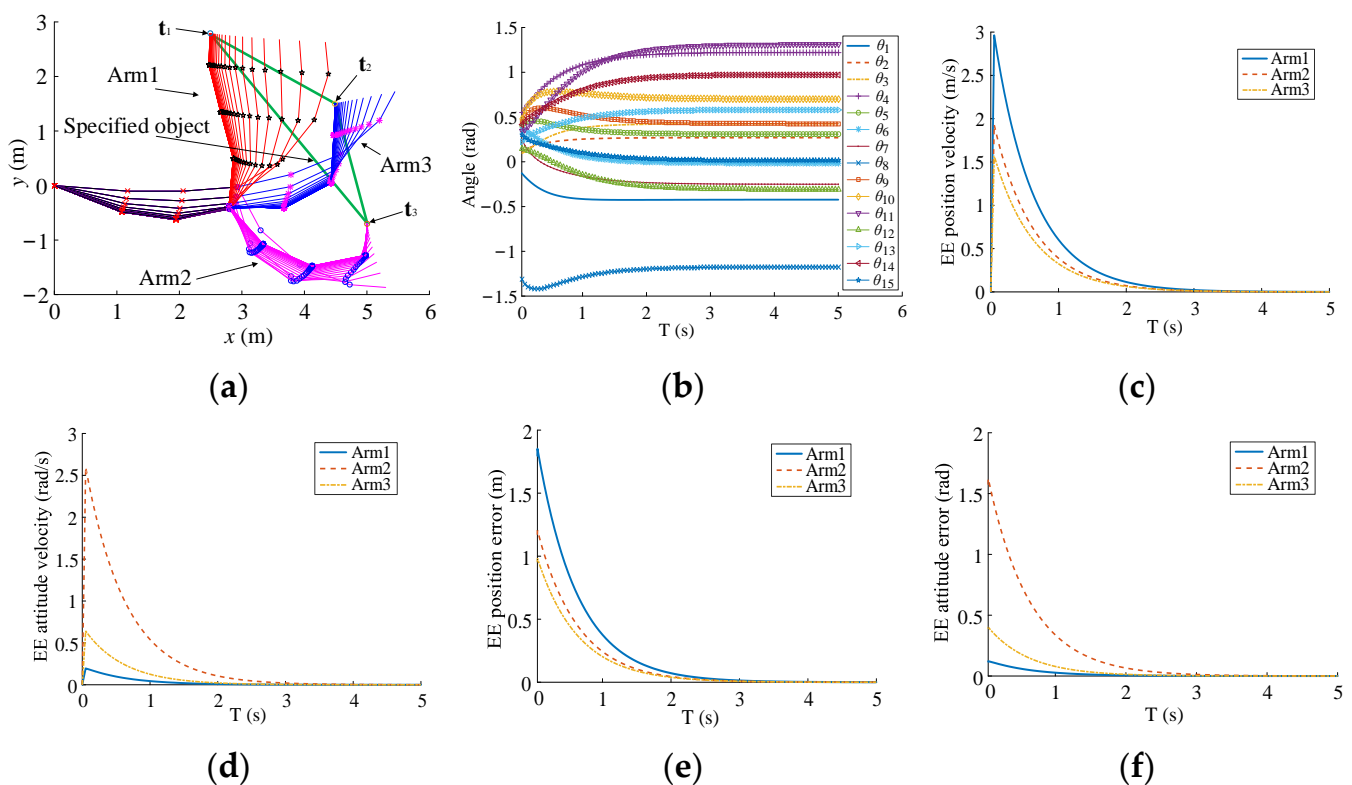
l_i	1	2	3	4	5
Length (m)	1.18	0.88	0.88	0.88	0.88
l_i	6	7	8	9	10
Length (m)	0.88	57.85	0.88	0.88	0.88
l_i	11	12	13	14	15
Length (m)	57.85	0.88	0.88	0.88	57.85

Table 2. Initial joint parameters of three-arm robot.

θ_i	1	2	3	4	5
Initial angle ($^\circ$)	-5.0	5.0	5.0	30.0	20.0
θ_i	6	7	8	9	10
Initial angle ($^\circ$)	20.0	20.0	-70.0	20.0	20.0
θ_i	11	12	13	14	15
Initial angle ($^\circ$)	20.0	10.0	10.0	20.0	20.0

Table 3. Kinematic parameters.

Parameters	μ	$\eta_{1,1}$	N	b	k	R
Value	0.08	1/6	3	3	1	15
Parameters	$M_{0,1}$	M_1	M_2	M_3	λ	ΔT
Value	3	4	4	4	0.01	0.05

**Figure 7.** Inverse kinematics based on the traditional method in real time. (a) Motion of multi-arms. (b) Joint angles. (c) EE position velocity. (d) EE attitude velocity. (e) EE position error. (f) EE attitude error.

Both methods can make the EEs arrive at the designated location of the specified object, as illustrated in Figures 7a and 8a. The joint angles are shown in Figures 7b and 8b. Due to $\dot{\mathbf{t}} = 0$, $\Delta \mathbf{e}_i(T)$ is equal to $\mathbf{v}_i(T)\Delta T$, and the velocity curves of the EE pose are similar with those of the EE pose error, as shown in Figures 7c–f and 8c–f. Because of the sub-base motion, the convergence of the EE pose error is much faster than that based on the traditional method. Furthermore, the EE pose velocities form uniform motion states before complete convergence in Figure 8c,d. Thus, the proposed sub-base method is conducive to the synchronization from the initial motion state to the cooperative motion state and improves the efficiency for the carrying of the multi-arm robot.

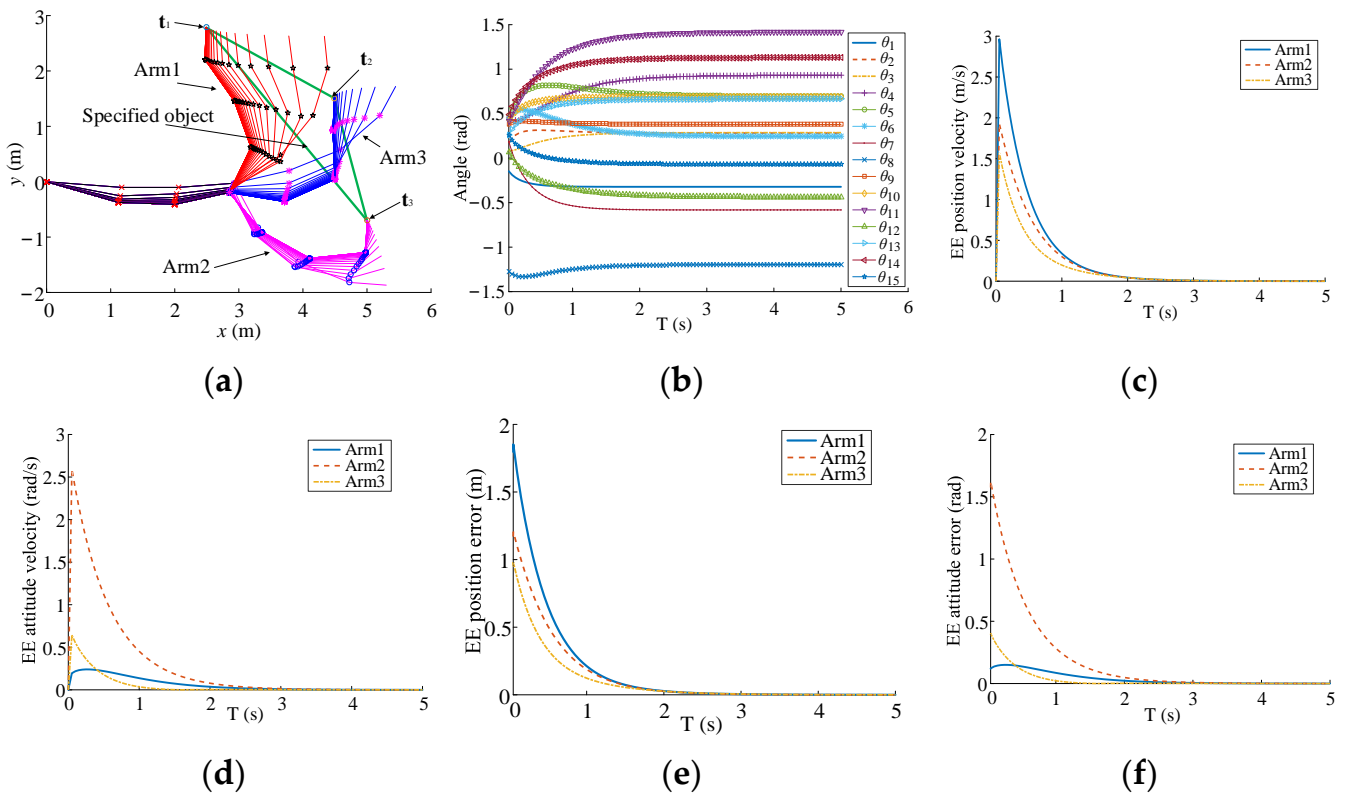


Figure 8. Inverse kinematics based on the sub-base method in real time. (a) Motion of multi-arms. (b) Joint angles. (c) EE position velocity. (d) EE attitude velocity. (e) EE position error. (f) EE attitude error.

5. Experimental Verification

5.1. Experimental Setup

The two-arm robot with 13-DoFs is used to compare the proposed synchronous planning for the collaborative manipulation, and 1-DoF is common in the base joint, as shown in Figure 9. The D-H parameters of the two-arm robot are presented in Table 4. In Figure 10, the principle of the experimental setup can be briefly described as follows.

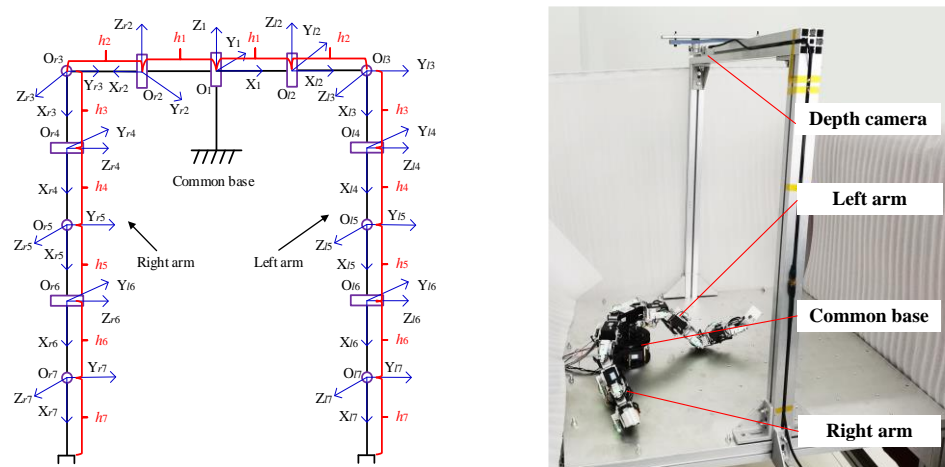
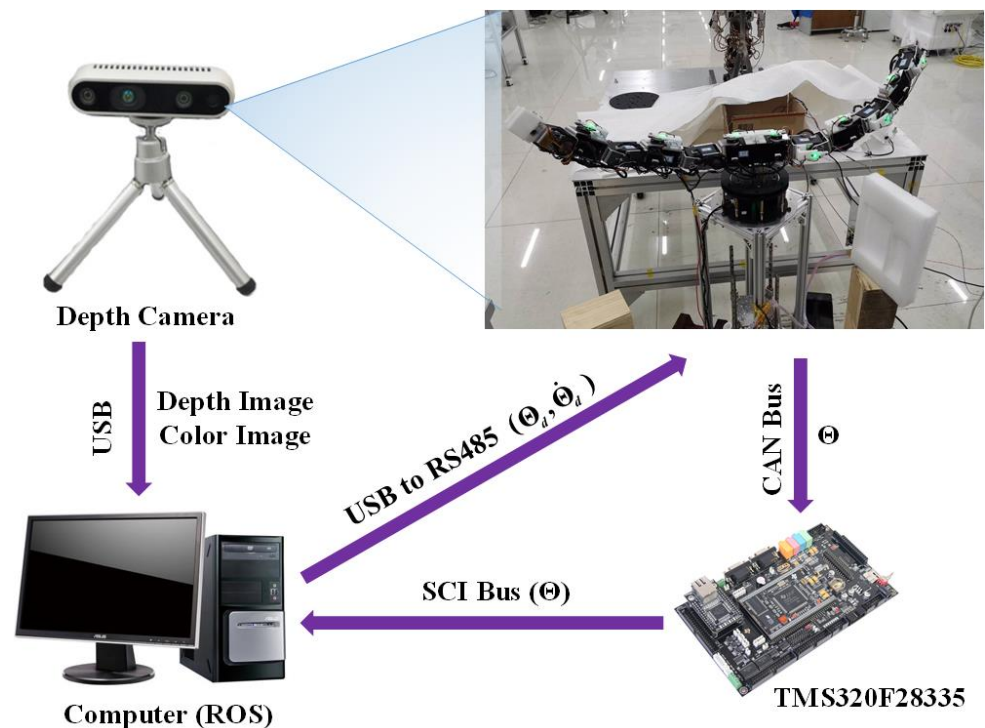


Figure 9. The configuration of two-arm robot with 13-DoFs.

Table 4. D-H parameters of two-arm robot *.

	i	d_i	a_i (mm)	α_i (rad)	β_i		i	d_i	a_i	α_i (mm)	β_i (rad)
Left Arm	1	0	$h_1 = 42$	0	0	Right Arm	1	0	$h_1 = 42$	π	0
	2	0	$h_2 = 84$	$\pi/2$	0		2	0	$h_2 = 84$	$\pi/2$	0
	3	0	$h_3 = 84$	$-\pi/2$	0		3	0	$h_3 = 84$	$-\pi/2$	0
	4	0	$h_4 = 84$	$\pi/2$	0		4	0	$h_4 = 84$	$\pi/2$	0
	5	0	$h_5 = 78$	$-\pi/2$	0		5	0	$h_5 = 78$	$-\pi/2$	0
	6	0	$h_6 = 71$	$\pi/2$	0		6	0	$h_6 = 71$	$\pi/2$	0
	7	0	$h_7 = 71$	0	0		7	0	$h_7 = 71$	0	0

* The base (i.e., θ_1) is the common joint of two arms.

**Figure 10.** The principle of two-arm robot with 13-DoFs.

- (1) The global depth camera transfers the observed frame of depth and color images to the computer by using a universal serial bus (USB) in real time. The robot operating system (ROS) node runs in the computer and extracts the position information of the object from each frame image. The vision-processing procedures in the computer are developed based on the morphology, using the Open Source Computer Vision Library and the camera Software Development Kit.
- (2) The joint feedback data, Θ , of the robot are transmitted to the TMS320F28335 controller through the Controller Area Network (CAN) bus. At the same time, the controller transfers the received joint data, Θ , to the ROS nodes through the Serial Communication Interface (SCI) bus in real time.
- (3) The ROS nodes receive the data (x_{obj} , Θ) and execute the forward and inverse kinematics calculation and motion-planning algorithm. The real-time control command data, Θ_d and $\dot{\Theta}_d$, are sent to the robot joint actuator through the RS485 bus to control the motion. The baud rate of the CAN bus, USB serial bus, SCI bus, and RS485 bus is set to 1 MHz. The control period of the whole system containing the proposed kinematically synchronous planning method is less than 10 ms.

5.2. Synchronous Planning Experiments

(1) Collaboration Carrying

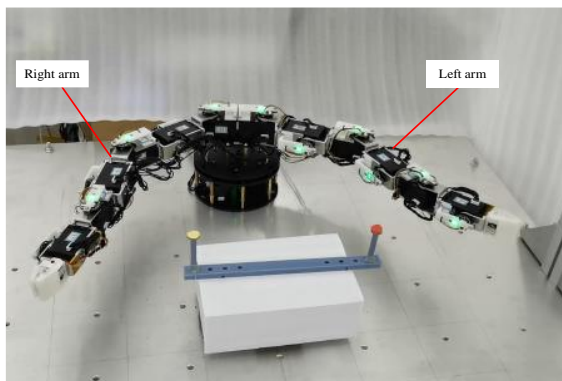
The collaboration carrying an object was provided to verify the feasibility of the proposed planning method, as illustrated in Figure 11. The parameters for carrying are shown in Tables 5 and 6, respectively. The carried object has translational and rotational motion. The rotation center position is $(-1.5 \text{ mm}, -342.7 \text{ mm}, -0.85 \text{ mm})^T$. The EEs of left and right arms reach the initial locations, as shown in Figure 11a. The tracked initial locations on the carried object are $\mathbf{t}_1^{\text{init}} = (115.841, -342.7, -0.85, 1.2092, -1.2092, -1.2092)^T$, $\mathbf{t}_2^{\text{init}} = (-118.841, -342.7, -0.85, 1.2092, -1.2092, -1.2092)^T$. At 6.75 s, two arms begin to carry the object, as shown in Figure 11b. The rotation angle relative to the rotation center is 0.0005 rad, and the translational motion of the rotation center is 0.15 mm in a cycle period, ΔT . During the collaboration process, the position of the tracked locations on the carried object moves back and forth in a 5 s cycle, and the tracked pose trajectory in the initial 2.25 s (from 6.75 s to 9 s) is as follows:

$$\mathbf{t}_1 = \mathbf{t}_1^{\text{init}} - \begin{pmatrix} 117.341 \cdot [\sin(0.0005(T - 6.75) + 0.0005\Delta T) - \sin(0.0005(T - 6.75))] + 0.15(T - 6.75) \\ 117.341 \cdot [\cos(0.0005(T - 6.75) + 0.0005\Delta T) - \cos(0.0005(T - 6.75))] \\ -0.15(T - 6.75) \\ \hat{\mathbf{f}}_1 \cdot \psi_1 \end{pmatrix} \quad (43)$$

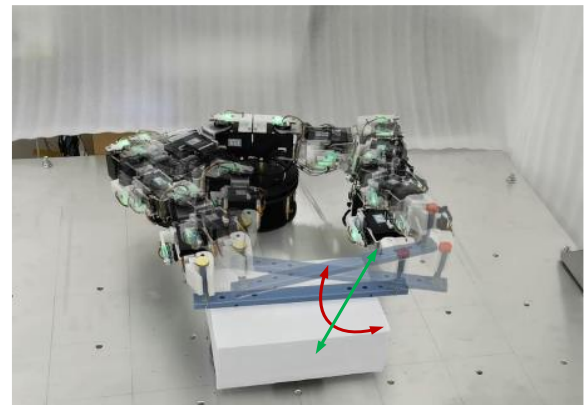
$$\mathbf{t}_2 = \mathbf{t}_2^{\text{init}} + \begin{pmatrix} 117.341 \cdot [\sin(0.0005(T - 6.75) + 0.0005\Delta T) - \sin(0.0005(T - 6.75))] + 0.15(T - 6.75) \\ 117.341 \cdot [\cos(0.0005(T - 6.75) + 0.0005\Delta T) - \cos(0.0005(T - 6.75))] \\ 0.15(T - 6.75) \\ \hat{\mathbf{f}}_2 \cdot \psi_2 \end{pmatrix} \quad (44)$$

where $\hat{\mathbf{f}}_1$, φ_1 , $\hat{\mathbf{f}}_2$, and φ_2 can be obtained according to the following equations:

$$\mathbf{R}_1(\hat{\mathbf{f}}_1, \varphi_1) = \mathbf{R}_2(\hat{\mathbf{f}}_2, \varphi_2) = \begin{bmatrix} \cos(-\frac{\pi}{2} - 0.0005(T - 6.75)) & -\sin(-\frac{\pi}{2} - 0.0005(T - 6.75)) & 0 \\ \sin(-\frac{\pi}{2} - 0.0005(T - 6.75)) & \cos(-\frac{\pi}{2} - 0.0005(T - 6.75)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad (45)$$



(a)



(b)

Figure 11. Carrying task. (a) Initial configuration. (b) Manipulating process.

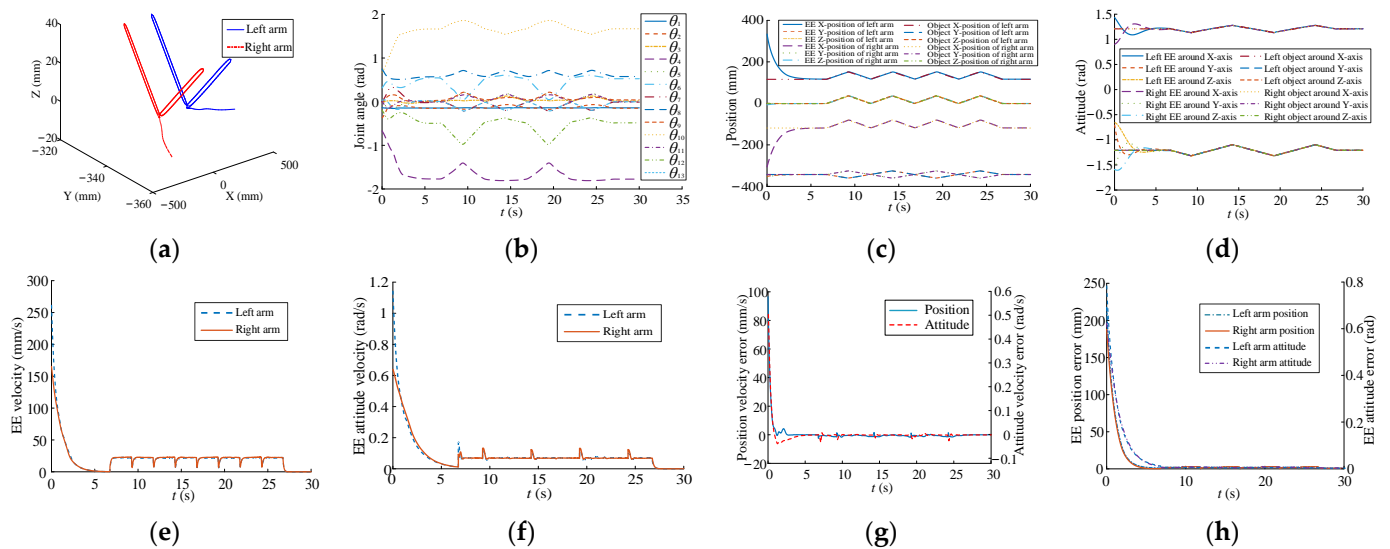
Table 5. Kinematic parameters of two-arm robot.

Parameters	μ	$\eta_{1,1}$	N	b	k	r
Value	0.005	5	2	2	1	13
Parameters	$M_{0,1}$	M_1	M_2	λ	ΔT	—
Value	1	6	6	0.01	0.05	—

Table 6. The parameters of self-organizing competitive neural network.

Parameters	δ_i	σ_i	c_i	η
Value	1×10^{-5}	1×10^{-5}	2.0	0.03

All the parameters in Tables 5 and 6, the rotation center position, $\mathbf{t}_1^{\text{init}}$, and $\mathbf{t}_1^{\text{init}}$, were used in Figure 4. The inverse kinematics is based on Equation (30) to obtain the joint angle, Θ . The change of the new weight value, $\Delta \hat{w}_i$, of the self-organizing competitive neural network is obtained according to Equation (27). The learning rate, i.e., η , η_1 , $\eta_{n,1}$, \dots , $\eta_{n,k}$, is used to calculate the carrying step size of EE motion in Cartesian space. With the continuous learning and competition of the proposed planning method, the position velocities of EEs gradually form a consistent movement. Figure 12a,b show the motion paths of EEs and joint trajectories, respectively. Figure 12c,d show the periodic change of pose in Cartesian space where the Y-axis position, the Z-axis position, and the attitude of the EEs remain the same. The position velocity and the attitude velocity reach synchronous motion states when the position velocity error and the attitude velocity error decrease to 0, as illustrated in Figure 12e–h.

**Figure 12.** Trajectories for dual arms in carrying task. (a) EE movement. (b) Joint trajectory. (c) EE position. (d) EE attitude. (e) Position velocity. (f) Attitude velocity. (g) Pose velocity error. (h) EE pose error.

(2) Manipulating the pliers

Figure 13 shows that the two arms manipulated a pair of pliers. The parameters are illustrated in Tables 5 and 6. The rotation center position is $(-50.0 \text{ mm}, -347.55 \text{ mm}, \text{ and } 0.85 \text{ mm})^T$. The length of the pliers' handle is 135 mm. The tracked initial locations on the carried object for the EEs of left and right arms are $\mathbf{t}_1^{\text{init}} = (86.841, -212.7, 0.85, 1.2092, -1.2092, -1.2092)^T$ and $\mathbf{t}_2^{\text{init}} = (-186.841, -212.7, 0.85, 1.2092, -1.2092, -1.2092)^T$. At 6.25 s, two arms begin to manipulate the pliers, as shown in Figure 14b. During the collaboration process, the position of the tracked locations on the pliers moves back and

forth in a 3 s cycle, and the tracked pose trajectory in the initial 1.5 s (from 6.25 s to 7.75 s) is as follows:

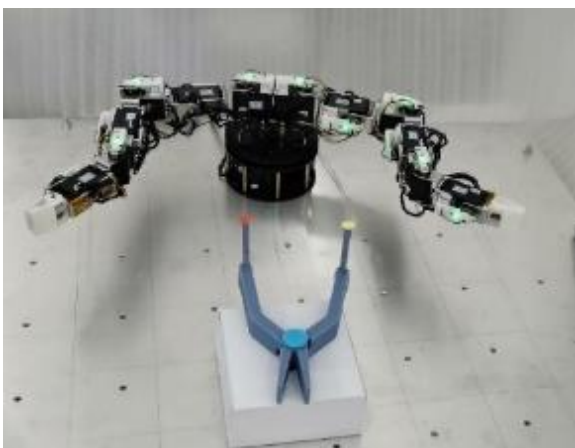
$$\mathbf{t}_1 = \mathbf{t}_1^{\text{init}} + \begin{pmatrix} 135 \cdot [\sin(0.0005(T - 6.25) + 0.0005\Delta T) - \sin(0.0005(T - 6.25))] \\ 135 \cdot [\cos(0.0005(T - 6.25) + 0.0005\Delta T) - \cos(0.0005(T - 6.25))] \\ 0 \\ \hat{\mathbf{f}}_1 \cdot \psi_1 \end{pmatrix} \quad (46)$$

$$\mathbf{t}_2 = \mathbf{t}_2^{\text{init}} + \begin{pmatrix} 135 \cdot [\sin(-0.0005(T - 6.25) - 0.0005\Delta T) - \sin(-0.0005(T - 6.25))] \\ 135 \cdot [\cos(0.0005(T - 6.25) + 0.0005\Delta T) - \cos(0.0005(T - 6.25))] \\ 0 \\ \hat{\mathbf{f}}_2 \cdot \psi_2 \end{pmatrix} \quad (47)$$

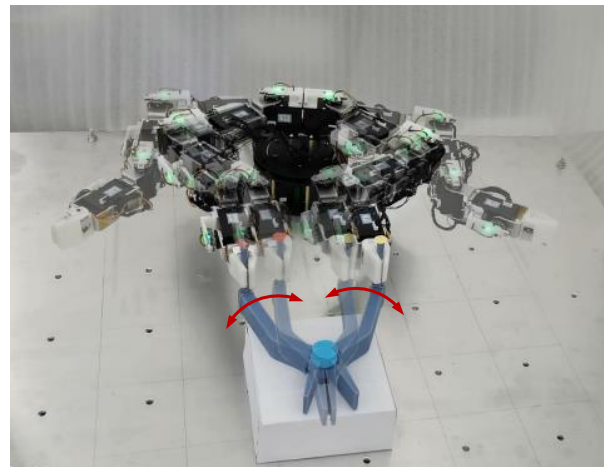
where $\hat{\mathbf{f}}_1$, ψ_1 , $\hat{\mathbf{f}}_2$, and ψ_2 can be obtained according to the following equations:

$$\mathbf{R}_1(\hat{\mathbf{f}}_1, \varphi_1) = \begin{bmatrix} \cos(-\frac{\pi}{2} - 0.0005(T - 6.25)) & -\sin(-\frac{\pi}{2} - 0.0005(T - 6.25)) & 0 \\ \sin(-\frac{\pi}{2} - 0.0005(T - 6.25)) & \cos(-\frac{\pi}{2} - 0.0005(T - 6.25)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad (48)$$

$$\mathbf{R}_2(\hat{\mathbf{f}}_2, \varphi_2) = \begin{bmatrix} \cos(-\frac{\pi}{2} + 0.0005(T - 6.25)) & -\sin(-\frac{\pi}{2} + 0.0005(T - 6.25)) & 0 \\ \sin(-\frac{\pi}{2} + 0.0005(T - 6.25)) & \cos(-\frac{\pi}{2} + 0.0005(T - 6.25)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad (49)$$



(a)



(b)

Figure 13. Manipulating pilers. (a) Initial configuration. (b) Manipulating process.

The inverse kinematics is based on Equation (30) to obtain the joint angle, Θ . The change of the new weight value, $\cdot \hat{w}_i$, is obtained according to Equation (27) and used to adjust the EE motions according to Equation (28). The learning rate, i.e., η , η_1 , $\eta_{n,1}$, \dots , $\eta_{n,k}$, is used to calculate the step size of manipulating the pilers in the Cartesian space. The proposed planning method makes the position velocities of EEs gradually form a consistent movement. Figure 14a,b show the motion paths of EEs and joint trajectories, respectively. Figure 14c,d show the periodic change of pose in Cartesian space, where the Y-axis position, the Z-axis position, and the attitude of the EEs remain the same. The position velocity and the attitude velocity of the EEs are almost the same and showed a small error when synchronous motion states were reached, as illustrated in Figure 14e–h.

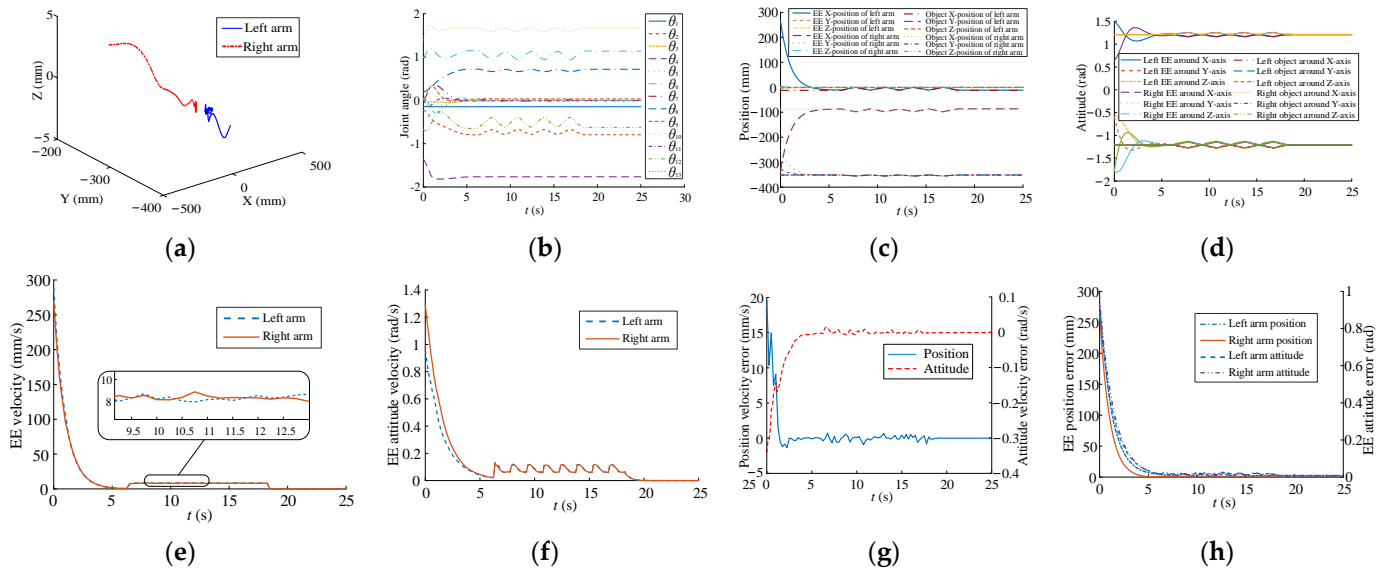


Figure 14. Trajectories for dual arms in manipulating pilers. (a) EE movement. (b) Joint trajectory. (c) EE position. (d) EE attitude. (e) Position velocity. (f) Attitude velocity. (g) Pose velocity error. (h) EE pose error.

(3) Manipulating a rudder

Manipulating a rudder is illustrated in Figure 15. The parameters are shown in Tables 5 and 6. The rotation center position is $(0.0 \text{ mm}, -342.7 \text{ mm}, -0.85 \text{ mm})^T$. The diameter of the rudder is 136.84 mm. The tracked initial locations on the carried object for the EEs of left and right arms are $\mathbf{t}_1^{\text{init}} = (136.84, -342.7, -0.85, 1.2092, -1.2092, -1.2092)^T$ and $\mathbf{t}_2^{\text{init}} = (-136.84, -342.7, 0.85, 1.2092, -1.2092, -1.2092)^T$. At 6.75 s, two arms begin to manipulate the rudder, as shown in Figure 16b. During the collaboration process, the position of the tracked locations on the rudder moves back and forth in a 2.5 s cycle, and the tracked pose trajectory in the initial 1.25 s (from 6.75 s to 8 s) is as follows:

$$\mathbf{t}_1 = \mathbf{t}_1^{\text{init}} - \begin{pmatrix} 136.841 \cdot [\sin(0.0005(T - 6.75) + 0.0005\Delta T) - \sin(0.0005(T - 6.75))] \\ 136.841 \cdot [\cos(0.0005(T - 6.75) + 0.0005\Delta T) - \cos(0.0005(T - 6.75))] \\ 0 \\ \hat{\mathbf{f}}_1 \cdot \psi_1 \end{pmatrix} \quad (50)$$

$$\mathbf{t}_2 = \mathbf{t}_2^{\text{init}} + \begin{pmatrix} 136.841 \cdot [\sin(0.0005(T - 6.75) + 0.0005\Delta T) - \sin(0.0005(T - 6.75))] \\ 136.841 \cdot [\cos(0.0005(T - 6.75) + 0.0005\Delta T) - \cos(0.0005(T - 6.75))] \\ 0 \\ \hat{\mathbf{f}}_2 \cdot \psi_2 \end{pmatrix} \quad (51)$$

where $\hat{\mathbf{f}}_1$, ψ_1 , $\hat{\mathbf{f}}_2$, and ψ_2 can be obtained according to the following equations:

$$\mathbf{R}_1(\hat{\mathbf{f}}_1, \psi_1) = \mathbf{R}_2(\hat{\mathbf{f}}_2, \psi_2) = \begin{bmatrix} \cos(-\frac{\pi}{2} - 0.0005(T - 6.75)) & -\sin(-\frac{\pi}{2} - 0.0005(T - 6.75)) & 0 \\ \sin(-\frac{\pi}{2} - 0.0005(T - 6.75)) & \cos(-\frac{\pi}{2} - 0.0005(T - 6.75)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad (52)$$

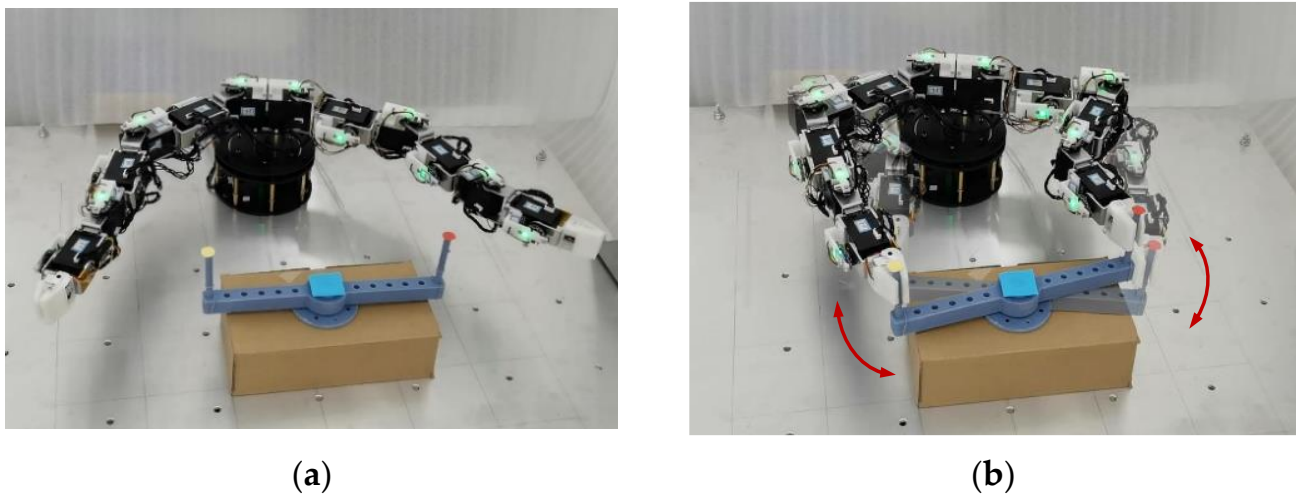


Figure 15. Manipulating rudder. (a) Initial configuration. (b) Manipulating process.

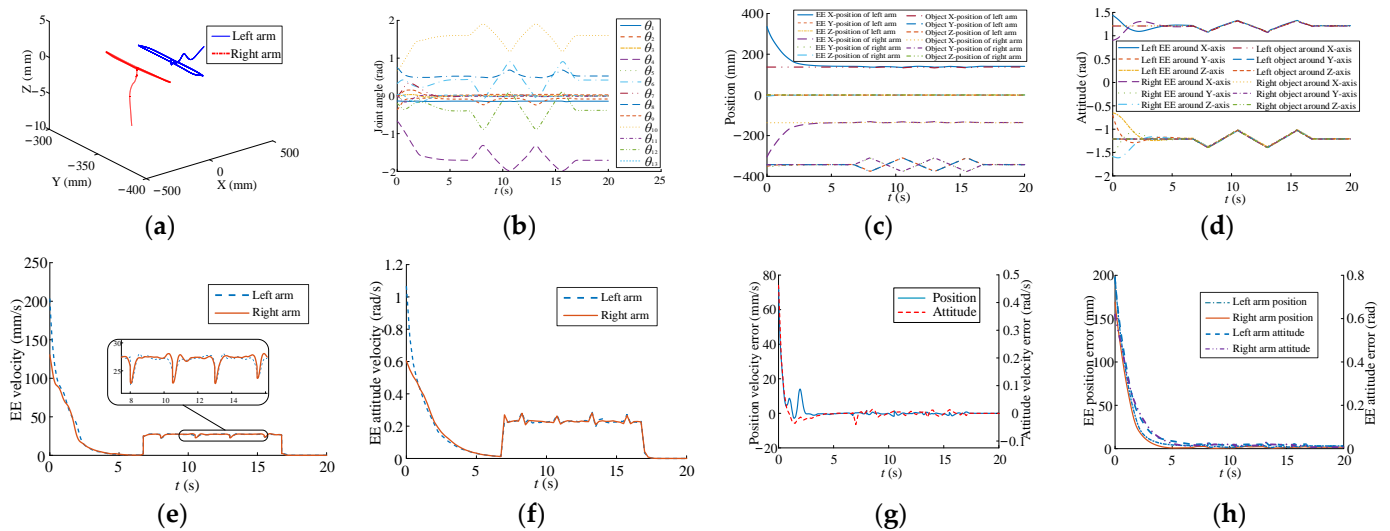


Figure 16. Trajectories for dual arms in manipulating rudder. (a) EE movement. (b) Joint trajectory. (c) EE position. (d) EE attitude. (e) Position velocity. (f) Attitude velocity. (g) Pose velocity error. (h) EE pose error.

Similar to the previous experiments, the inverse kinematics is based on Equation (30) to obtain the joint angle, Θ . The change of the new weight value, $\Delta\hat{w}_i$, is obtained according to Equation (27) and used to adjust the EE motions according to Equation (28). The learning rate, i.e., $\eta, \eta_1, \eta_{n,1}, \dots, \eta_{n,k}$, is used to calculate the step size of manipulating the rudder in Cartesian space. Since the shape of the rudder is symmetrical at the center, the tracked locations are also symmetrical, and the EE attitude changes are the same. In Figure 16c,d, the two arms begin to manipulate the rudder at 6.75 s. In Figure 16e–h, the EE pose errors can converge to 0, and the corresponding EE motions own almost the same states ultimately by using the proposed planning method with the learning of the self-organizing competitive neural network.

Considering (1)–(3) in Section 5.2 comprehensively, the pose speed and pose error converge to 0 for the successful execution of coordination manipulation and correspond to Equation (2), and no arm motion is set as the reference for the other arms. Hence, the proposed planning method based on the self-organizing competitive neural network owns the feasibility, synchronism, and effectiveness in achieving the collaboration manipulations for the arms with physical coupling and has the contributions to the practical application.

6. Conclusions

This paper presents a real-time kinematically synchronous planning method for collaborative manipulation through the self-organizing competitive neural network. This method considers a type of collaborative manipulation known as the synchronization of EE motion. The sub-bases are defined for the configuration of multi-arms to obtain the Jacobian matrix of common DoFs and ensure the pose errors converging along the reducing direction of the EE total pose errors. The simulations of multi-arms with common DoFs display the consistency before the pose errors converge completely and make contributions to the collaborative manipulation of multi-arms. On this basis, an unsupervised competitive neural network is raised to regard the EE synchronous motion as the competition of neurons and adaptively increase the convergence ratio of multi-arms through the mutual learning and competition of neurons by using the inner star rules. The stability of multi-arms system is analyzed through the Lyapunov theory. Various simulations and experiments confirm that the proposed synchronous planning method is feasible, synchronous, and has the application potentiality in different cooperative manipulation tasks.

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

Funding: This work was supported in part by the Science and Technology Innovation 2030-"Brain Science and Brain-like Research" Major Project: 2021ZD0201400; the National Natural Science Foundation of China under Grants 92048301 and 61473102; the Science Foundation of Henan University of Technology under Grants 2021BS060 and 2020BS059; and the Key Scientific Research Project Plans of Higher Education Institutions in Henan Province under Grants 23B410002 and 22A410002.

Data Availability Statement: Not applicable.

Conflicts of Interest: No conflicts of interest exist in regard to this manuscript, and this manuscript was approved by all authors for publication. I would like to declare, on behalf of my co-authors, that the work described was original research that was not published previously and not under consideration for publication elsewhere, either in whole or in part. All the authors listed approved the manuscript that is enclosed.

References

1. Li, Z.; Yuan, W.; Zhao, S.; Yu, Z.; Kang, Y.; Chen, C.L.P. Brain-Actuated Control of Dual-Arm Robot Manipulation With Relative Motion. *IEEE Trans. Cogn. Dev. Syst.* **2019**, *11*, 51–62. [[CrossRef](#)]
2. Yang, C.; Chen, C.; Wang, N.; Ju, Z.; Fu, J.; Wang, M. Biologically Inspired Motion Modeling and Neural Control for Robot Learning From Demonstrations. *IEEE Trans. Cogn. Dev. Syst.* **2019**, *11*, 281–291.
3. Nicolis, D.; Palumbo, M.; Zanchettin, A.M.; Rocco, P. Occlusion-Free Visual Servoing for the Shared Autonomy Teleoperation of Dual-Arm Robots. *IEEE Robot. Autom. Lett.* **2018**, *3*, 796–803. [[CrossRef](#)]
4. Schmaus, P.; Leidner, D.; Bayer, R.; Pleintinger, B.; Krüger, T.; Lii, N.Y. Continued Advances in Supervised Autonomy User Interface Design for METERON SUPVIS Justin. In Proceedings of the 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2–9 March 2019; pp. 1–11.
5. Ambrose, R.; Savely, R.; Goza, S.; Strawser, P.; Radford, N. Mobile Manipulation using NASA's Robonaut. In Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, 26 April–1 May 2004; Volume 2, pp. 2104–2109.
6. Chen, B.; Wang, Y.; Lin, P. A Hybrid Control Strategy for Dual-arm Object Manipulation Using Fused Force/Position Errors and Iterative Learning. In Proceedings of the 2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Auckland, New Zealand, 9–12 July 2018; pp. 39–44.
7. Nozawa, S.; Kakiuchi, Y.; Okada, K.; Inaba, M. Controlling the Planar Motion of a Heavy Object by Pushing with a Humanoid Robot using Dual-Arm Force Control. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 1428–1435.
8. Suarez, A.; Heredia, G.; Ollero, A. Physical-Virtual Impedance Control in Ultralightweight and Compliant Dual-Arm Aerial Manipulators. *IEEE Robot. Autom. Lett.* **2018**, *3*, 2553–2560. [[CrossRef](#)]

9. Lee, J.; Chang, P.H.; Jamisola, R.S. Relative Impedance Control for Dual-Arm Robots Performing Asymmetric Bimanual Tasks. *IEEE Trans. Ind. Electron.* **2014**, *61*, 3786–3796. [[CrossRef](#)]
10. Kong, L.; He, W.; Yang, C.; Li, Z.; Sun, C. Adaptive Fuzzy Control for Coordinated Multiple Robots With Constraint Using Impedance Learning. *IEEE T. Cybern.* **2019**, *49*, 3052–3063. [[CrossRef](#)] [[PubMed](#)]
11. Zhang, Z.; Lin, Y.; Li, S.; Li, Y.; Yu, Z.; Luo, Y. Tricriteria Optimization-Coordination Motion of Dual-Redundant-Robot Manipulators for Complex Path Planning. *IEEE Trans. Control Syst. Technol.* **2018**, *26*, 1345–1357. [[CrossRef](#)]
12. Li, X.; Tan, S.; Feng, X.; Rong, H. LSPB Trajectory Planning: Design for the Modular Robot Arm Applications. In Proceedings of the 2009 International Conference on Information Engineering and Computer Science, Wuhan, China, 19–20 December 2009; pp. 1–4.
13. Bouteraa, Y.; Ghommam, J.; Poisson, G. Adaptive Backstepping Synchronization for Networked Lagrangian Systems. *Int. J. Comput. Appl. Technol.* **2012**, *42*, 1–8. [[CrossRef](#)]
14. Uchiyama, M.; Dauchez, P. Symmetric Kinematic Formulation and Non-master/Slave Coordinated Control of Two-Arm Robots. *Adv. Robot.* **1992**, *7*, 361–383. [[CrossRef](#)]
15. Caccavale, F.; Chiacchio, P.; Chiaverini, S. Task-Space Regulation of Cooperative Manipulators. *Automatica* **2000**, *36*, 879–887. [[CrossRef](#)]
16. Adorno, B.; Fraise, P.; Druon, S. Dual Position Control Strategies Using the Cooperative Dual Task-Space Framework. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 3955–3960.
17. Park, H.; Lee, C. Cooperative-Dual-Task-Space-based Whole-Body Motion Balancing for Humanoid Robots. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 4797–4802.
18. Park, H.; Lee, C. Extended Cooperative Task Space for Manipulation Tasks of Humanoid Robots. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 6088–6093.
19. Park, H.; Lee, C. Dual-Arm Coordinated-Motion Task Specification and Performance Evaluation. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; pp. 929–936.
20. Basile, F.; Caccavale, F.; Chiacchio, P.; Coppola, J.; Marino, A. A Decentralized Kinematic Control Architecture for Collaborative and Cooperative Multi-Arm Systems. *Mechatronics* **2013**, *23*, 1100–1112. [[CrossRef](#)]
21. Basile, F.; Caccavale, F.; Chiacchio, P.; Coppola, J.; Curatella, C. Task-Oriented Motion Planning for Multi-Arm Robotic Systems. *Robot. Comput.-Integr. Manuf.* **2012**, *28*, 569–582. [[CrossRef](#)]
22. Zhang, Z.; Kong, L.; Niu, Y. A Time-Varying-Constrained Motion Generation Scheme for Humanoid Robot Arms. In *Advances in Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 10878, pp. 757–767.
23. Curkovic, P.; Jerbic, B. Dual-arm Robot Motion Planning based on Cooperative Coevolution. *Emerg. Trends Technol. Innov.* **2010**, *314*, 169–178.
24. Caccavale, F.; Lippiello, V.; Muscio, G.; Pierri, F.; Ruggiero, F.; Villani, L. Grasp Planning and Parallel Control of A Redundant Dual-Arm/Hand Manipulation System. *Robotica* **2013**, *31*, 1169–1194. [[CrossRef](#)]
25. Stavridis, S.; Doulgeri, Z. Bimanual Assembly of Two Parts with Relative Motion Generation and Task Related Optimization. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 7131–7136.
26. Almeida, D.; Karayiannidis, Y. A Lyapunov-Based Approach to Exploit Asymmetries in Robotic Dual-Arm Task Resolution. In Proceedings of the 2019 IEEE 58th Conference on Decision and Control (CDC), Nice, France, 3 May 2019; pp. 4252–4258.
27. Jin, T.; Xia, H. Lookback Option Pricing Models Based on the Uncertain Fractional-Order Differential Equation with Caputo Type. *J. Ambient Intell. Hum. Comput.* **2021**, *14*, 6435–6448. [[CrossRef](#)]
28. Zhang, S.; Cao, Y. Consensus in Networked Multi-robot Systems Via Local State Feedback Robust Control. *Int. J. Adv. Robot Syst.* **2019**, *16*, 45–52. [[CrossRef](#)]
29. Zhang, S.; Cao, Y. Cooperative Localization Approach for Multi-Robot Systems Based on State Estimation Error Compensation. *Sensors* **2019**, *19*, 3842. [[CrossRef](#)] [[PubMed](#)]
30. Aristidou, A.; Lasenby, J. *Inverse Kinematics: A Review of Existing Techniques and Introduction of a New Fast Iterative Solver*; Technical Report; Cambridge University Engineering Department: Cambridge, UK, 2009.
31. Buss, S.R.; Kim, J.-S. Selectively Damped Least Squares for Inverse Kinematics. *J. Graph. Tools* **2005**, *10*, 37–49. [[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.