

Article

Decentralized Adaptive Control of Closed-Kinematic Chain Mechanism Manipulators

Tri T. Nguyen ^{1,2}, Charles C. Nguyen ^{1,*}, Tuan M. Nguyen ¹ , Tu T. C. Duong ^{1,3} , Ha Tang T. Ngo ¹ and Lu Sun ⁴

¹ Department of Electrical Engineering and Computer Science, The Catholic University of America, Washington, DC 20064, USA; 65nguyent@cua.edu (T.T.N.); 11nguyen@cua.edu (T.M.N.); duongt@cua.edu (T.T.C.D.); ngot@cua.edu (H.T.T.N.)

² Department of Electrical Engineering, University of Science and Technology-The University of Danang, Da Nang City 500000, Vietnam

³ Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology and Education, Ho Chi Minh City 700000, Vietnam

⁴ Department of Civil Engineering Technology, Environmental Management and Safety, Rochester Institute of Technology, Rochester, New York, NY 14623, USA; lxsite@rit.edu

* Correspondence: nguyen@cua.edu

Abstract: This paper presents a new decentralized adaptive control scheme for motion control of robot manipulators built based on a closed-kinematic chain mechanism (CKCM). By employing the synchronization technique and model reference adaptive control (MRAC) based on the Lyapunov direct method, the Decentralized Adaptive Synchronized Control scheme (DASCS) is developed. The DASCS can ensure global asymptotic convergence of tracking errors while forcing all active joints to move in a predefined synchronous manner in the presence of uncertainties and sudden changes in payload. Furthermore, the control scheme has a simple structure, independent of the manipulator's dynamic model, ensuring computational efficiency. Results of computer simulations conducted to evaluate the performance of the control scheme applied to controlling the motion of a CKCM manipulator with six degrees of freedom are reported and discussed.

Keywords: decentralized control; model reference adaptive control; closed-kinematic chain mechanism; parallel robots; synchronized tracking control; Stewart Platform



Academic Editor: Zheng Chen

Received: 13 March 2025

Revised: 9 April 2025

Accepted: 12 April 2025

Published: 18 April 2025

Citation: Nguyen, T.T.; Nguyen, C.C.; Nguyen, T.M.; Duong, T.T.C.; Ngo, H.T.T.; Sun, L. Decentralized Adaptive Control of Closed-Kinematic Chain Mechanism Manipulators. *Machines* **2025**, *13*, 331. <https://doi.org/10.3390/machines13040331>

Copyright: © 2025 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/>).

1. Introduction

Robot manipulators whose structure assumes a closed-kinematic chain mechanism (CKCM) have attracted great attention from robotic researchers due to their advantages over their counterparts, namely manipulators with open kinematic chain mechanisms (OKCM) [1,2]. CKCM manipulators can mitigate the weaknesses of OKCM manipulators such as low stiffness, poor stability, small payload, and accumulated and amplified errors from link to link [3,4]. The CKCM was first employed in the design and construction of the Stewart Platform (SP), which possesses six degrees of freedom (DOF) [5]. Nowadays, the SP has a wide application area in spacecraft simulation, vehicle dynamics, high-speed train dynamics, airplane dynamics, drone manipulation, medical rehabilitation, and high-speed assembly operation [6–8].

In order for a robot to track a desired trajectory, a closed-loop feedback controller employs sensors to gain awareness of its surroundings and then adjusts the appropriate control strategy to achieve a desired performance, much like the way a human brain works. In general, robotic systems are complicated, time-varying, and highly nonlinear systems with uncertainties and disturbances in their working environment [9]. Moreover, due to the

closed-loop structure of CKCM manipulators, the motion of each active joint is constrained by the motion of other active joints. Thus, the lack of synchronization between active joints will lead to a large coupling effect, which degrades the performance of the whole system or even damages its mechanical structure, especially at high speeds and with large payloads. Therefore, it is essential for a CKCM manipulator to possess a synchronized controller that has the capacity to adapt and improve itself under changing conditions without human intervention and to force each active joint to follow its desired trajectory as close as possible while synchronizing its motion with the other active joints' motions in a defined, synchronous manner.

Recently, a synchronization method based on cross-coupling control technology has been developed to solve the abovementioned problem by allowing each control loop to receive feedback from itself as well as from the others to achieve better coordination and then improve the manipulator performance significantly [10]. This approach is a powerful tool for not only CKCM manipulators but also systems that require multiple objects to operate simultaneously to achieve a common goal, such as swarms of mobile robot [11,12] or multiple robotic manipulator systems [13–15]. While the work in reference [10], dealing with cross-coupled control of biaxial systems in which an error in an axis impacts the control loops of both axes, is a time-based approach, the recently introduced method for synchronization in reference [16] is based on position domain. The authors in reference [16] introduced a position-domain control method rather than a time-domain one to enhance contour tracking performance of a multi-axis motion system. The synchronization of the axes is achieved by formulating the motion system as a master-slave motion system in which the master motion is used as an independent variable and the slave motions are expressed as functions of the master motion according to the required contour trajectories.

Most existing synchronized control schemes are model-based adaptive control schemes [17,18] and their implementation requires a precise dynamical model of the manipulator [19–21]. Due to the fact that it is difficult if not impossible to acquire a precise dynamical model of the manipulator and that its associated mathematical calculation is computationally intensive, these synchronized control schemes are not suitable for real-time applications in CKCM manipulators, particularly for those with more than 2 DOFs. To tackle the above dynamic modelling issue, robotics researchers have considered intelligent approaches such as fuzzy logic [22,23] and artificial neural network control schemes [24–26]. However, these intelligent controllers exhibited shortcomings such as assurance of stability, high computational demand, slow convergence rate, and time-consuming training process. Finally, the authors in references [27–29] considered synchronized control schemes that have simple structures and do not require the knowledge of the manipulator dynamics in the implementation of their control laws. However, their controller gains must still be selected based on conservative estimates of the manipulator dynamic model, which could be problematic and impractical.

During the trajectory tracking processes, CKCM manipulators can be treated as a group of multi OKCM manipulators holding the same payload. Hence, it is possible to develop a decentralized control scheme in which each active joint is controlled independently to minimize the computational cost and move synchronously to achieve the desired tracking performance of the moving platform [29].

In this paper, we develop a new decentralized adaptive synchronized control scheme (DASCS) for CKCM manipulators. Unlike conventional decentralized control schemes [30], each sub-controller of the DASCS receives feedback from not only its active joint but also from two neighboring ones. Compared to the work in references [10,16], the DASCS developed in the present paper is a time-domain approach providing synchronization of one particular joint with its two neighboring joints, while the work in reference [10] dealt with

synchronization of two axes and the work in reference [16] is a position-domain approach. Then, by applying the model reference adaptive control (MRAC) technique based on the Lyapunov direct method into the synchronized control method, the DASCs guarantees the global asymptotic convergence of tracking errors to zero while synchronizing all active joints' motion and overcoming uncertainties and disturbances. Moreover, the DASCs, whose controller gains are updated by an adaptation law driven only by the actual and desired trajectories of active joints, is computationally efficient and thus is suitable for real-time applications.

The structure of this paper is as follows. Section 2 describes the synchronization control method and Section 3 presents the development of the DASCs. Section 4 presents and discusses the results of a computer simulation study conducted to evaluate the effectiveness of the DASCs in comparison with another existing adaptive control scheme. Section 5 concludes the paper with summary and future research directions.

2. Synchronization Control

For a CKCM manipulator with n active joints, to achieve synchronization, we define the following relationship:

$$\frac{q_1(t)}{q_{d1}(t)} = \frac{q_2(t)}{q_{d2}(t)} = \dots = \frac{q_n(t)}{q_{dn}(t)} \quad (1)$$

where q_{di} and $q_i(t)$ denote the desired and actual trajectory of the i th active joint, respectively.

If all ratios in (1) are equal, or in other words, if Equation (1) is valid, then all active joints will move in a synchronous manner [27]. However, considering synchronization of all active joints may lead to a heavy computational burden, especially when the number of active joints is large.

To avoid synchronization of all active joints, we proceed to rewrite (1) as a set of subgoals:

$$\frac{q_i(t)}{q_{di}(t)} = \frac{q_{i+1}(t)}{q_{di+1}(t)} \quad (2)$$

From subgoals (2), synchronization functions are defined as follows:

$$\begin{cases} f_1[q_1(t), q_2(t)] = \frac{q_1(t)}{q_{d1}(t)} - \frac{q_2(t)}{q_{d2}(t)} = 0 \\ f_2[q_2(t), q_3(t)] = \frac{q_2(t)}{q_{d2}(t)} - \frac{q_3(t)}{q_{d3}(t)} = 0 \\ \vdots \\ f_i[q_i(t), q_{i+1}(t)] = \frac{q_i(t)}{q_{di}(t)} - \frac{q_{i+1}(t)}{q_{di+1}(t)} = 0 \\ \vdots \\ f_n[q_n(t), q_1(t)] = \frac{q_n(t)}{q_{dn}(t)} - \frac{q_1(t)}{q_{d1}(t)} = 0 \end{cases} \quad (3)$$

We use Taylor Series [31] to expand (3) at the desired joint trajectories $q_{di}(t)$ to obtain the following:

$$\begin{aligned} f_i[q_i(t), q_{i+1}(t)] &= f_i[q_{di}(t), q_{di+1}(t)] + \sum_{j=i}^{j=i+1} \left[\frac{\partial f_i(\cdot)}{\partial q_j} \right]_{q_{di}(t)} \{q_j(t) - q_{dj}(t) + \mathcal{O}[q_j(t)]\} = \\ &\quad - \sum_{j=i}^{j=i+1} \left[\frac{\partial f_i(\cdot)}{\partial q_j} \right]_{q_{di}(t)} q_{ej}(t) + \mathcal{O}[q_j(t)] \end{aligned} \quad (4)$$

where $\mathcal{O}[q_j(t)]$ denotes the higher order terms.

Considering only the first-order derivative, then (4) becomes the following:

$$f_i[q_i(t), q_{i+1}(t)] = \frac{q_{ei}(t)}{q_{di}(t)} - \frac{q_{ei+1}(t)}{q_{di+1}(t)} = 0 \quad (5)$$

where

$$q_{ei}(t) = q_{di}(t) - q_i(t). \quad (6)$$

and $q_{ei}(t)$, $q_{di}(t)$, $q_i(t)$ denote the tracking error trajectory, desired trajectory, and actual trajectory of the i th active joint, respectively. In general, a controller acts on errors which are the differences between desired and actual variables. Thus, a set of synchronization errors should be defined as follows [31]:

$$\begin{aligned} \varepsilon_1(t) &= p_1(t)q_{e1}(t) - p_2(t)q_{e2}(t) \\ \varepsilon_2(t) &= p_2(t)q_{e2}(t) - p_3(t)q_{e3}(t) \\ &\vdots \\ \varepsilon_i(t) &= p_i(t)q_{ei}(t) - p_{i+1}(t)q_{ei+1}(t) \\ &\vdots \\ \varepsilon_n(t) &= p_n(t)q_{en}(t) - p_1(t)q_{e1}(t) \end{aligned} \quad (7)$$

where ε_i represents the synchronization error of the i th active joint and $p_i(t) = \frac{1}{q_{di}(t)}$ and $p_i(t)$ is bounded. Obviously, if all synchronization errors in (7) are equal to zero, then the synchronization goal (1) is automatically achieved.

It is noted that by making all $\varepsilon_i(t) = 0$, only the synchronization goal is achieved. However, the goal of the control scheme is to drive both tracking and synchronization errors to zero. Consequently, an error encompassing both synchronization and tracking should be defined. It is noted that generally in control of multi-axis motion, cross-axis couplings can cause error force and position disturbances in an axis when a desired motion is generated along another axis. Thus, for our case, the cross-coupling errors that combine both tracking errors and synchronization errors should be defined as follow [32]:

$$\begin{aligned} e_1^*(t) &= q_{e1}(t) + \beta \int_0^t [\varepsilon_1(w) - \varepsilon_n(w)]dw \\ e_2^*(t) &= q_{e2}(t) + \beta \int_0^t [\varepsilon_2(w) - \varepsilon_1(w)]dw \\ &\vdots \\ e_i^*(t) &= q_{ei}(t) + \beta \int_0^t [\varepsilon_i(w) - \varepsilon_{i-1}(w)]dw \\ &\vdots \\ e_n^*(t) &= q_{en}(t) + \beta \int_0^t [\varepsilon_n(w) - \varepsilon_{n-1}(w)]dw \end{aligned} \quad (8)$$

where β is a positive coupling parameter and $e_i^*(t)$ represents the cross-coupling error of the i th active joint. From (7) and (8), it is noted that the local controller of each joint receives feedback information from not only itself but also the two neighboring ones. We proceed now to present the development of the DASCs.

3. Development of the DASCs

The dynamics of a robot manipulator having n active joints can be expressed in joint-space as follows [33]:

$$M(q)\ddot{q} + N(q, \dot{q}) + G(q) + H(\dot{q}) = T(t) \quad (9)$$

where q is the $n \times 1$ vector of joint displacement, $M(q)$ is the $n \times n$ symmetric positive definite inertia matrix, $N(q, \dot{q})$ is the $n \times 1$ Coriolis and centrifugal torque vector, $G(q)$ is the $n \times 1$ gravitational torque vector, and $H(\dot{q})$ is the $n \times 1$ frictional torque vector.

The i th subsystem of the dynamical system (9) can be expressed in the following decentralized form [30]:

$$m_{ii}(q)\ddot{q}_{ii}(t) + \left[\sum_{\substack{j=1 \\ j \neq i}}^6 m_{ij}(q)\ddot{q}_j(t) \right] + n_i(q, \dot{q}) + g_i(q) + h_i(\dot{q}) = T_i(t) \quad (10)$$

where T_i is the control input of the i th active joint.

As seen in (10), the i th subsystem not only contains the gravity, friction, Coriolis, and centrifugal torque, namely $n_i(q, \dot{q}) + g_i(q) + h_i(q)$ for the i th active joint, but also is coupled between it and the remaining subsystems shown by the term $\left[\sum_{\substack{j=1 \\ j \neq i}}^6 m_{ij}(q)\ddot{q}_j(t) \right]$.

Let

$$d_i(q, \dot{q}, \ddot{q}) = \sum_{\substack{j=1 \\ j \neq i}}^6 m_{ij}(q)\ddot{q}_j(t) + n_i(q, \dot{q}) + g_i(q) + h_i(\dot{q}) \quad (11)$$

Then, (10) becomes the following:

$$m_{ii}(q)\ddot{q}_{ii}(t) + d_i(q, \dot{q}, \ddot{q}) = T_i(t) \quad (12)$$

Next, we define a command vector $u_i(t)$ as follows:

$$u_i(t) = q_{di}(t) + \beta \int_0^t [\varepsilon_n(w) - \varepsilon_{n-1}(w)]dw + \alpha \int_0^t e_i^*(w)dw \quad (13)$$

where $q_{di}(t)$ denotes the desired trajectory of the i th active joint and α is a positive constant. We also define a generalized error $r_i(t)$ as follows [30]:

$$r_i(t) = u_i(t) - q_i(t) = e_i^*(t) + \alpha \int_0^t e_i^*(w)dw \quad (14)$$

Now a control law for the i th subsystem in (10) can be defined as follows:

$$T_i(t) = f_i(t) + [k_{0i}(t)r_i(t) + k_{1i}(t)\dot{r}_i(t)] + [a_i(t)u_i(t) + b_i(t)\dot{u}_i(t) + c_i(t)\ddot{u}_i(t)] \quad (15)$$

Substituting (15) into (10) in light of (12) and dropping the subscript i for simplicity, we obtain the error differential equation in terms of the generalized error $r(t)$, which is indeed $r_i(t)$ as follows:

$$m\ddot{r}(t) + k_1(t)\dot{r}(t) + k_0(t)r(t) = d - f(t) - a(t)u(t) - b(t)\dot{u}(t) + (m - c)\ddot{u}(t) \quad (16)$$

Let us define the 2×1 position-velocity error vector $X(t) = \begin{pmatrix} r(t) \\ \dot{r}(t) \end{pmatrix}$ and then express Equation (16) in a state space form as follows:

$$\dot{X}(t) = \begin{pmatrix} 0 & 1 \\ -\Delta_1 & -\Delta_2 \end{pmatrix} X(t) + \begin{pmatrix} 0 \\ \Delta_0 \end{pmatrix} + \begin{pmatrix} 0 \\ \Delta_3 \end{pmatrix} u(t) + \begin{pmatrix} 0 \\ \Delta_4 \end{pmatrix} \dot{u}(t) + \begin{pmatrix} 0 \\ \Delta_5 \end{pmatrix} \ddot{u}(t) \quad (17)$$

where the variables $\Delta_1 = \frac{k_0}{m}$; $\Delta_2 = \frac{k_1}{m}$; $\Delta_0 = \frac{d-f}{m}$; $\Delta_3 = \frac{-a}{m}$; $\Delta_4 = \frac{-b}{m}$; $\Delta_5 = \frac{(m-c)}{m}$ contain the adjustable controller gains $k_1(t)$, $k_0(t)$, $f(t)$, $a(t)$, $b(t)$, and $c(t)$.

Equation (17) represents the “adjustable system” in the framework of MRAC. Next, the desired performance of the i th active joint can be specified in terms of a second-order homogeneous differential equation:

$$\ddot{r}_m(t) + 2\omega\zeta\dot{r}_m(t) + \omega^2 r_m(t) = 0 \quad (18)$$

where ζ is the damping ratio, ω is the natural frequency and $r_m(t)$ represents the desired trajectory of $r(t)$.

Equation (18) can be written in a state space form as follows:

$$\dot{X}_m(t) = \begin{pmatrix} 0 & 1 \\ -D_1 & -D_2 \end{pmatrix} X_m(t) = D X_m(t) \quad (19)$$

where $D = \begin{pmatrix} 0 & 1 \\ -D_1 & -D_2 \end{pmatrix}$, $X_m(t) = \begin{pmatrix} r_m(t) \\ \dot{r}_m(t) \end{pmatrix}$, $D_1 = \omega^2$, $D_2 = 2\omega\zeta$ and $r_m(t)$ and $\dot{r}_m(t)$ are the desired generalized position error and velocity errors, respectively.

Equation (19) represents the “reference model” and its solution is found as follows [30]:

$$X_m(t) = \exp(Dt) X_m(0) \quad (20)$$

which is under the assumption that the initial values of the actual and the desired trajectory are identical. In other words, $X_m(0) = 0$ yields $X(0) = 0$.

In order for the system to achieve adaptation, we define the adaptation error vector $E(t)$ as follows:

$$E(t) = X_m(t) - X(t) \quad (21)$$

Then, from (17) and (19), we obtain the following:

$$\dot{E}(t) = \begin{pmatrix} 0 & 1 \\ -D_1 & -D_2 \end{pmatrix} E(t) + \begin{pmatrix} 0 & 1 \\ \Delta_1 - D_1 & \Delta_1 - D_1 \end{pmatrix} X(t) + \begin{pmatrix} 0 \\ -\Delta_0 \end{pmatrix} + \begin{pmatrix} 0 \\ -\Delta_3 \end{pmatrix} u(t) + \begin{pmatrix} 0 \\ -\Delta_4 \end{pmatrix} \dot{u}(t) + \begin{pmatrix} 0 \\ -\Delta_5 \end{pmatrix} \ddot{u}(t) \quad (22)$$

Now, the controller adaptation laws will be derived to achieve the control objective, expressed by $X(t) \rightarrow X_m(t)$ or $E(t) \rightarrow 0$ as $t \rightarrow \infty$. To achieve this, we select a Lyapunov function candidate $v(t)$ [30] such that

$$v(t) = E^T P E + Q_0(\Delta_0 - \Delta_0^*)^2 + Q_1(\Delta_1 - D_1 - \Delta_1^*)^2 + Q_2(\Delta_2 - D_2 - \Delta_2^*)^2 + Q_3(\Delta_3 - \Delta_3^*)^2 + Q_4(\Delta_4 - \Delta_4^*)^2 + Q_5(\Delta_5 - \Delta_5^*)^2 \quad (23)$$

where $\Delta_0^*, \Delta_1^*, \dots, \Delta_6^*$ are function of time and $Q_0, Q_1, \dots, Q_5$ are arbitrary positive scalar.

To stabilize the linear part of (22), it is sufficient to choose ζ_i and ω_i such that the matrix D becomes Hurwitz, or all eigenvalues of D have negative real parts. If so, there exists a symmetric positive definite matrix $P = \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix}$ to satisfy the following Lyapunov Equation:

$$PD + D^T P = -Q \quad (24)$$

for any arbitrary symmetric positive definite matrix Q .

Finally, from [30], particularly in the section entitled Derivation of the Adaption Laws, the derivative of $v(t)$, namely $\dot{v}(t)$ will be negative definite of $E(t)$ when we choose the following adaptation laws:

$$\begin{aligned}
f(t) &= f(0) + \eta_2 \Omega(t) + \eta_1 \int_0^t \Omega(w) dw \\
k_0(t) &= k_0(0) + \gamma_2 \Omega(t) r(t) + \gamma_1 \int_0^t \Omega(w) r(w) dw \\
k_1(t) &= k_1(0) + \lambda_2 \Omega(t) \dot{r}(t) + \lambda_1 \int_0^t \Omega(w) \dot{r}(w) dw \\
a(t) &= a(0) + \mu_2 \Omega(t) u(t) + \mu_1 \int_0^t \Omega(w) u(w) dw \\
b(t) &= b(0) + \rho_2 \Omega(t) \dot{u}(t) + \rho_1 \int_0^t \Omega(w) \dot{u}(w) dw \\
c(t) &= c(0) + \sigma_2 \Omega(t) \ddot{u}(t) + \sigma_1 \int_0^t \Omega(w) \ddot{u}(w) dw
\end{aligned} \tag{25}$$

where $\eta_1, \gamma_1, \lambda_1, \mu_1, \rho_1, \sigma_1$ are positive constant and $\eta_2, \gamma_2, \lambda_2, \mu_2, \rho_2, \sigma_2$ are zero or positive constant, $\Omega(t) = P_2 r(t) + P_3 \dot{r}(t)$ where P_2 and P_3 are positive constant, depending on the reference model. Furthermore, $f(0), k_0(0), k_1(0), a(0), b(0), c(0)$ which are the initial conditions of $f(t), k_0(t), k_1(t), a(t), b(t), c(t)$, respectively, can be set arbitrarily.

As a result, $X(t) \rightarrow 0$ as $t \rightarrow \infty$ results in $\begin{bmatrix} e_i^*(t) + \beta \int_0^t e_i^*(w) dw \\ \dot{e}_i^*(t) + \beta e_i^*(t) \end{bmatrix} \rightarrow 0$ as $t \rightarrow \infty$.

Then, $e_i^*(t) \rightarrow 0$ and $\dot{e}_i^*(t) \rightarrow 0$ as $t \rightarrow \infty$ [34]. Thus, $q_{ei}(t), \dot{q}_{ei}(t)$ are bounded from (9) and from differentiating (9). Moreover, $\dot{\varepsilon}_i(t)$ are bounded from differentiating (8), then $\varepsilon_i(t)$ are uniformly continuous since every function which is differentiable and has bounded derivative is uniformly continuous [35,36].

From Barbalat's Lemma [36], suppose that $f: [0, \infty] \rightarrow \mathbb{R}$ is uniformly continuous and its derivative is bounded. Then, $f(t) \rightarrow 0$ as $t \rightarrow \infty$ holds. Therefore, $\varepsilon_i(t) \rightarrow 0$ as $t \rightarrow \infty$ since $\varepsilon_i(t)$ are uniformly continuous and $\dot{\varepsilon}_i(t)$ are bounded. From (8) and (9), $\varepsilon_i(t) \rightarrow 0$ and $e_i^*(t) \rightarrow 0$ results in $e_i(t) \rightarrow 0$ as $t \rightarrow \infty$ for all $i = 1 - n$. Therefore, the control objective is achieved.

It is noted that the adaptation laws are solutions for controller gain matrices of control law (15) and based on the actual, desired performances and their derivatives. Furthermore, we note that the control law (15) consists of three terms:

- The first term $f_i(t)$ represents auxiliary signal to improve the tracking performance and partly compensate for disturbance $d(t)$.
- The second term $\tau_i^{fb}(t) = [k_{0i}(t)r_i(t) + k_{1i}(t)\dot{r}_i(t)] = [k_{0i}(t) + \alpha]e_i^*(t) + \alpha k_{0i}(t) \int_0^t e_i^*(w) dw + k_{1i}(t)\dot{e}_i^*(t)$ represents the PID feedback controller.
- The last term $\tau_i^{ff}(t) = [a_i(t)u_i(t) + b_i(t)\dot{u}_i(t) + c_i\ddot{u}_i(t)]$ represent the feedforward controller.

The structure of the controller is shown in Figure 1. The required inputs of the i th controller are the desired position, velocity, and acceleration of its active joint. The required measurements are the actual positions and velocities of its active joint and the two adjacent ones.

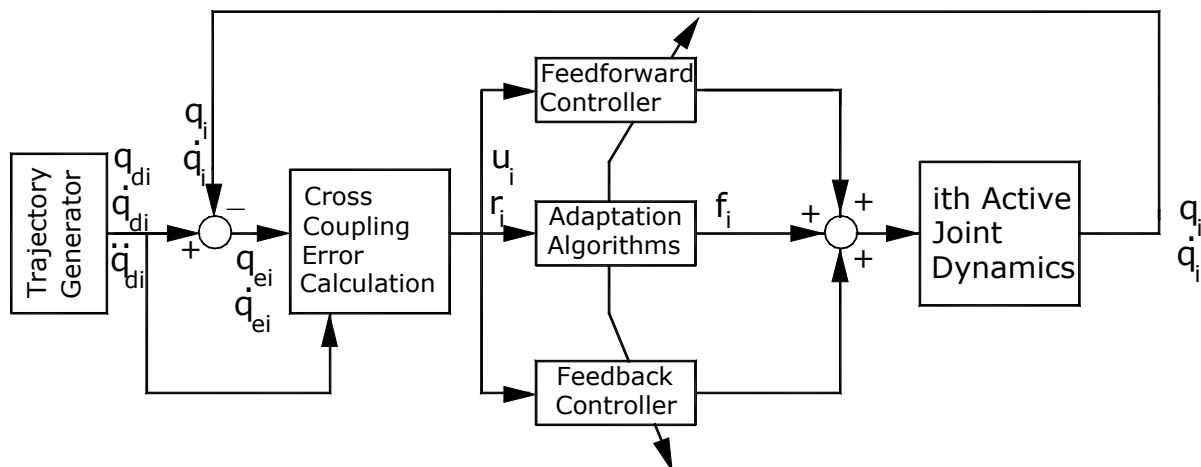


Figure 1. Decentralized adaptive synchronized control scheme of the i th active joint.

4. Computer Simulation Study of a 6-DOF CKCM Manipulator

This section presents a computer simulation study conducted to evaluate the effectiveness of the developed DASCs applied to control the motion of a 6-DOF CKCM manipulator. A brief kinematic analysis of this manipulator will first be presented, followed by a presentation and discussion of the computer simulation results.

4.1. Kinematic Analysis of the 6-DOF CKCM Manipulator

Figure 2 shows the CKCM manipulator that possesses six DOFs and comprises a stationary base platform and a moving platform coupled together through a number of links. Each link is a linear actuator consisting of a DC motor and a linear ball screw drive and its ends are connected to the base and moving platforms through ball joints.



Figure 2. The six DOF CKCM manipulator.

Figure 3 illustrates the manipulator frame assignment. To represent the position and orientation of the moving platform relative to the base platform, the fixed frame {B} whose origin is at the centroid of the base platform, and the moving frame {A} whose origin is at the centroid of the moving platform are assigned. The assignment of the above coordinate systems complies with the right-hand rule. Points A_i and points B_i are the attachment points of actuator i to the moving platform and base platform, respectively. In addition, each pair of ball joints on the base and moving platform is distributed symmetrically with a separation angle of 120° .

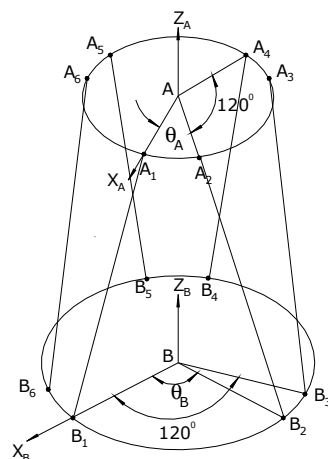


Figure 3. Frame assignment for the CKCM manipulator.

Referring to Figure 4, the position vectors $\mathbf{a}_i^A$, $\mathbf{b}_i^A$ describe the position of the ball joints at A_i , B_i with reference to the moving frame $\{A\}$ and the stationary base frame $\{B\}$, respectively. The radii of the moving and base platforms are denoted by r_A and r_B , respectively. In Figure 5, let θ_A represent the angle between ball joints A_1 and A_2 , which is the same between A_3 and A_4 , and A_5 and A_6 . Furthermore, if the angle between AA_i and X_A -axis is denoted by λ_i then the position vector $\mathbf{a}_i^A = (a_{ix} \ a_{iy} \ a_{iz})^T$ is computed by the following:

$$\mathbf{a}_i^A = [r_A \cos(\lambda_i) \ r_A \sin(\lambda_i) \ 0]^T \quad (26)$$

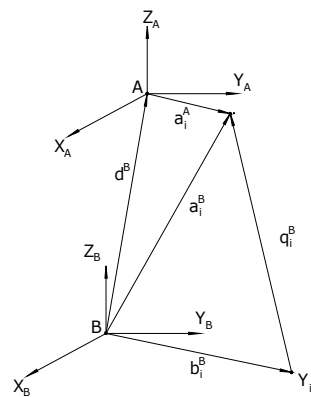


Figure 4. Vector diagram for the i th actuator.

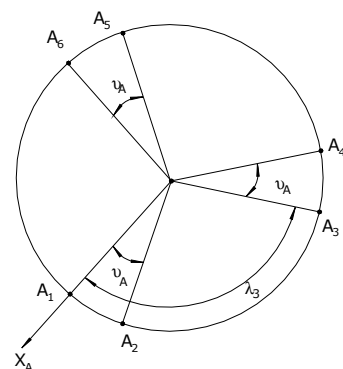


Figure 5. Position of the ball joints A_i with respect to the X_A -axis.

Similarly, if the angle between BB_i and X_B -axis is denoted by Λ_i as seen in Figure 6, then the position vector $\mathbf{b}_i^B = (b_{ix} \ b_{iy} \ b_{iz})^T$ is given by the following:

$$\mathbf{b}_i^B = [r_B \cos(\Lambda_i) \ r_B \sin(\Lambda_i) \ 0]^T \quad (27)$$

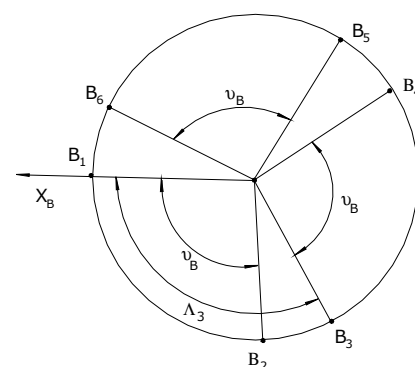


Figure 6. Position of the ball joints B_i with respect to the X_B -axis.

The angles λ_i and Λ_i are determined by the following:

$$\Lambda_i = 60(i-1) \text{ deg}; \lambda_i = 60(i-1) \text{ deg}, \text{ for } i = 1, 3, 5 \quad (28)$$

$$\Lambda_i = \Lambda_{i-1} + \theta_B \text{ deg}; \lambda_i = \lambda_{i-1} + \theta_A \text{ deg}, \text{ for } i = 2, 4, 6 \quad (29)$$

The kinematic equations can be derived from the vector diagram for each actuator in Figure 4. The length vector $\mathbf{q}_i^B = (q_{ix} \ q_{iy} \ q_{iz})^T$, expressing the vector B_iA_i with respect to frame {B}, can be computed as follows:

$$\mathbf{q}_i^B = \mathbf{a}_i^B - \mathbf{b}_i^B \quad (30)$$

where vector $\mathbf{a}_i^B$ represents the coordinates of point A_i with respect to frame {B}. Vector $\mathbf{d}_B$ denoting the position of point A with respect to frame {B} is given by the following:

$$\mathbf{d}_B = (x \ y \ z)^T \quad (31)$$

where x , y , and z are the Cartesian coordinates of point A in frame {B}. If the orientation of the moving frame {A} with respect to the base frame is specified by Z-Y-X Euler angles $(\alpha \ \beta \ \gamma)$, then the corresponding Euler angle orientation matrix ${}^B_A\mathbf{R}$ can be computed as follows:

$${}^B_A\mathbf{R} = \mathbf{R}_{zyx, \alpha\beta\gamma} = \begin{bmatrix} C\alpha C\beta & C\alpha S\beta S\gamma - S\alpha C\gamma & C\alpha S\beta C\gamma + S\alpha S\gamma \\ S\alpha C\beta & S\alpha S\beta S\gamma + C\alpha C\gamma & S\alpha S\beta C\gamma - C\alpha S\gamma \\ -S\beta & C\beta S\gamma & C\beta C\gamma \end{bmatrix} \quad (32)$$

Now vector $\mathbf{a}_i^B$ can be computed as follows:

$$\mathbf{a}_i^B = {}^B_A\mathbf{R}\mathbf{a}_i^A + \mathbf{d}_B \quad (33)$$

Substituting (33) into (30) yields the following:

$$\mathbf{q}_i^B = {}^B_A\mathbf{R}\mathbf{a}_i^A + \mathbf{d}_B - \mathbf{b}_i^B \quad (34)$$

Using (34), vector $\mathbf{q}_i^B$ can be solved for a desired Cartesian configuration of the moving platform, specified by $x, y, z, \alpha \ \beta \ \gamma$ as follows:

$$\mathbf{q}_i^B = \begin{bmatrix} q_{ix} \\ q_{iy} \\ q_{iz} \end{bmatrix} = \begin{bmatrix} r_{11}a_{ix} + r_{12}a_{iy} + r_{13}a_{iz} + x - b_{ix} \\ r_{21}a_{ix} + r_{22}a_{iy} + r_{23}a_{iz} + y - b_{iy} \\ r_{31}a_{ix} + r_{32}a_{iy} + r_{33}a_{iz} + z - b_{iz} \end{bmatrix} \quad (35)$$

where r_{ij} are the elements of the orientation matrix ${}^B_A\mathbf{R}$ given by (32). After solving for vector $\mathbf{q}_i^B$, the square of the required corresponding length of the i th actuator, q_i can be computed by the following:

$$q_i^2 = (q_{ix}^2 + q_{iy}^2 + q_{iz}^2) \text{ for } i = 1, 2, \dots, 6 \quad (36)$$

We proceed to substitute (35) into (36) to obtain the following:

$$\begin{aligned} q_i^2 = & x^2 + y^2 + z^2 + a_{ix}^2 + a_{iy}^2 + a_{iz}^2 + b_{ix}^2 + b_{iy}^2 + b_{iz}^2 \\ & + 2(r_{11}a_{ix} + r_{12}a_{iy} + r_{13}a_{iz})(x - b_{ix}) \\ & + 2(r_{21}a_{ix} + r_{22}a_{iy} + r_{23}a_{iz})(y - b_{iy}) \\ & + 2(r_{31}a_{ix} + r_{32}a_{iy} + r_{33}a_{iz})(z - b_{iz}) - 2(xb_{ix} + yb_{iy} + zb_{iz}) \end{aligned} \quad (37)$$

In Figure 6, since a_i^A and b_i^B lie on the $x_A - y_A$ plane and $x_B - y_B$ plane, respectively, we have the following:

$$a_{iz} = b_{iz} = 0 \quad (38)$$

$$a_{ix}^2 + a_{iy}^2 + a_{iz}^2 = r_A^2 \quad (39)$$

$$b_{ix}^2 + b_{iy}^2 + b_{iz}^2 = r_B^2 \quad (40)$$

Now in light of (39) and (40), (37) becomes the following:

$$q_i^2 = x^2 + y^2 + z^2 + r_A^2 + r_B^2 + 2(r_{11}a_{ix} + r_{12}a_{iy})(x - b_{ix}) + 2(r_{21}a_{ix} + r_{22}a_{iy})(y - b_{iy}) + 2(r_{31}a_{ix} + r_{32}a_{iy})z - 2(xb_{ix} + yb_{iy}) \quad \text{for } i = 1, 2, \dots, 6 \quad (41)$$

Equation (41) presents the closed-form solution of the inverse kinematics problem, in the sense that the actuator lengths q_i for $i = 1, 2, \dots, 6$ can be determined from the Cartesian configuration of the moving platform, specified by $x, y, z, \alpha, \beta, \gamma$.

The forward kinematics of the above six DOF CKCM manipulator deals with the determination of the Cartesian configuration of the moving platform with respect to the base platform, specified by $x, y, z, \alpha, \beta, \gamma$ based on the actuator lengths, q_i for $i = 1, 2, \dots, 6$. Unfortunately, (41) provides six nonlinear simultaneous equations with six unknowns that lead to no closed-form solutions for the forward kinematic problem. As a result, this problem is extremely difficult and needs to be solved by using iterative numerical methods such as the Newton Raphson's Method [37,38].

4.2. Computer Simulation Results

Table 1 shows the parameters of the above 6 DOF CKCM manipulator employed in the computer simulation study. The computer simulation study is carried out using SimMechanics™, version 2019 which is one of the toolboxes in MATLAB-Simulink, version 19. It provides the ability to easily model and visualize real mechanical systems that consist of multiple rigid bodies. Furthermore, the toolbox takes advantage of the Simulink environment to simulate the physical behaviors and motions of the modelled systems. The SimMechanics library, having the form of interconnected blocks, shows how the physical components with geometric and kinematic relationships of robots are mutually interconnected. Therefore, this approach does not require sophisticated mathematical knowledge about the system to be simulated and presents an easy and precise way to obtain the dynamic model of the system. Figure 7 shows a visualization of the SimMechanics™ model of the manipulator.

Table 1. Parameters of the 6-DOF CKCM manipulator.

Plant Parameters	Value
Base radius (m)	0.36
Platform radius(m)	0.27
Initial height (m)	0.5
Base offset angle (deg)	2.5
Platform offset angle (deg)	10
Mass of the platform (kg)	4.92
Mass of the leg cylinder (kg)	10.29
Inertia coefficient of the platform, I_{xx} (kg*m ²)	0.09
Inertia coefficient of the platform, I_{yy} (kg*m ²)	0.09
Inertia coefficient of the platform, I_{zz} (kg*m ²)	0.18

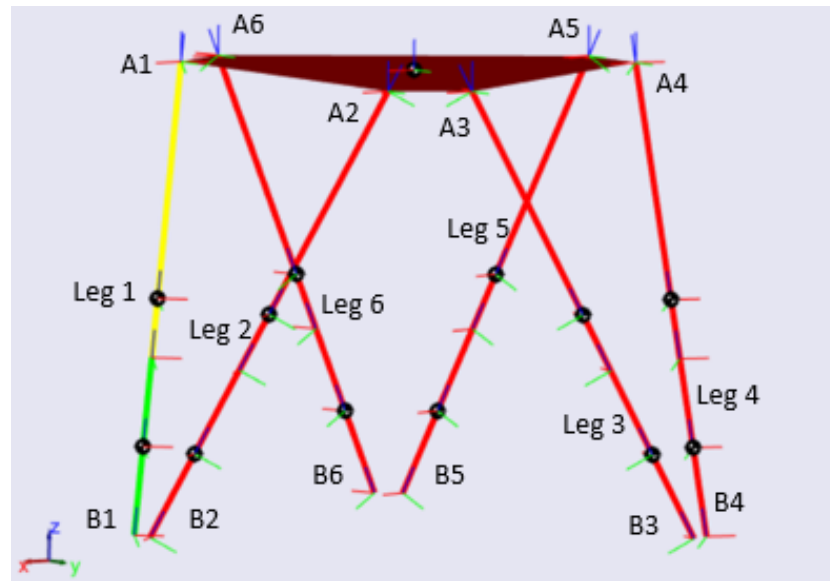


Figure 7. Simulink model of the 6-DOF CKCM.

For comparison purpose, the developed DASC Scheme (DASCS) and the traditional MRAC scheme (SMRAC) developed by Seraji in [30] were employed to track a reference model with the desired damping ratios and natural frequencies chosen as $\omega_i = 10$ rad/s; $\zeta_i = 1$, for $i = 1, 2, 3, \dots, 6$. We aim to compare the performance of the DASCS with the SMRAC when tracking a circular path in the X-Y plane, described by the following equations:

$$\begin{aligned} x(t) &= 0.1\cos(2\pi t) \quad (\text{m}) \\ y(t) &= 0.1\sin(2\pi t) \quad (\text{m}) \\ z(t) &= 0.5 \quad (\text{m}) \end{aligned} \quad (42)$$

$$\alpha(t) = \beta(t) = \gamma(t) = 0(\text{rad}) \quad (43)$$

where $\alpha(t), \beta(t), \gamma(t)$ are rotations about the $x(t), y(t), z(t)$ axes. To evaluate the adaptive ability of the above schemes, a 40 kg payload was suddenly applied at the center of the mass of the moving platform at time $t = 3$ s.

Figure 8 shows the time trajectories of the synchronization errors of the six legs (actuator lengths, q_i for $i = 1, 2, \dots, 6$) for the DASCS and the SMRAC, indicated in red and blue, respectively. As we can see, the synchronization errors of the DASCS are much smaller than those of the SMRAC. As tabulated in Table 2, the average synchronization errors (ASE) of the DASCS are 0.0021 mm, 0.0122 mm, 0.0156 mm, 0.0176 mm, 0.0117 mm, and 0.0211 mm for Leg 1, Leg 2, Leg 3, Leg 4, Leg 5, and Leg 6, respectively, while the ASEs of the SMRAC are 0.5109 mm, 0.6245 mm, 0.7342 mm, 0.3961 mm, 0.54 mm, and 0.8946 mm for Leg 1, Leg 2, Leg 3, Leg 4, Leg 5, and Leg 6, respectively. We also observe that after the introduction of the sudden payload, the DASCS takes 5 s to make the legs converge to their steady states while the SMRAC fails to do so. Moreover, as shown in Figures 9 and 10, the DASC scheme enables the synchronization errors to globally and asymptotically converge to zero while the SMRAC fails to do so. After observing the time trajectories of the synchronization errors of the DASCS and SMRAC in Figures 8–10, it is clear that the synchronization errors of the DASCS are “synchronized” while those of the SMRAC are not. This is expected in light of the different characteristics of the two control schemes, with the DASCS having the capability to control the synchronization errors and the SMRAC not having this capability at all.

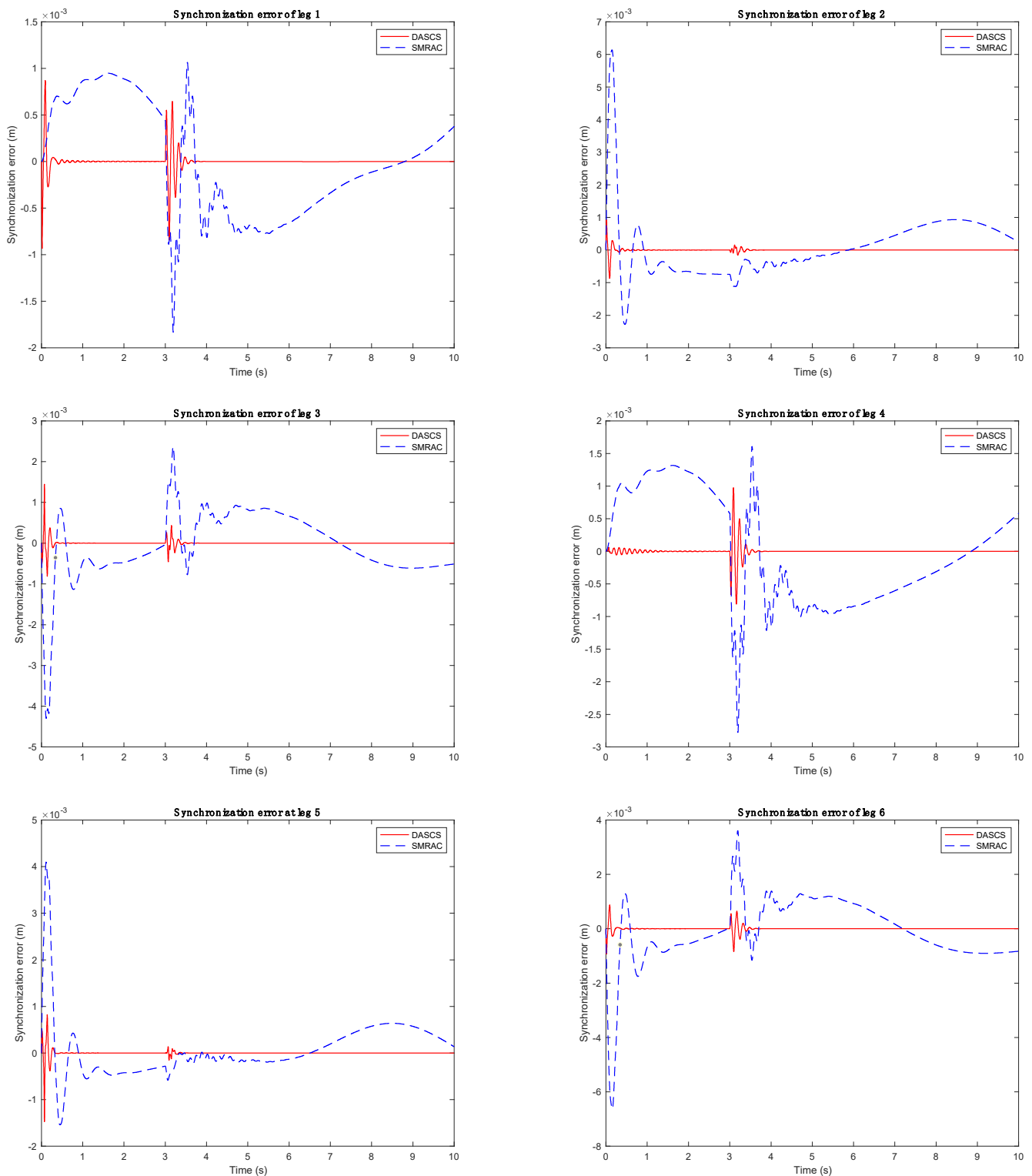


Figure 8. Time responses of synchronization errors for SMRAC and DASCS.

Figures 11 and 12 show the time trajectories of the tracking errors of the manipulator legs for both control schemes. From the figures, we see that the tracking errors of the DASCS have better transient and steady state responses than those of the SMRAC. As illustrated in Table 2, the average tracking errors (ATE) of the DASCS are 0.0988 mm, 0.0905 mm, 0.988 mm, 0.0909 mm, 0.0972 mm, 0.0832 mm, and 0.0860 mm for Leg 1, Leg 2, Leg 3, Leg 4, Leg 5, and Leg 6, respectively, while the ATEs of the SMRAC are much higher, being 0.379 mm, 0.323 mm, 0.318 mm, 0.375 mm, 0.331 mm, and 0.34 mm for Leg 1, Leg 2, Leg 3, Leg 4, Leg 5, and Leg 6, respectively.

Leg 3, Leg 4, Leg 5, and Leg 6, respectively. In other words, the ATEs of SMRAC are 383%, 356%, 321%, 385%, 397%, and 395% of those of the DASCs. Regarding reaction to the sudden payload change at $t = 3$ s, the leg time trajectories with the DASCs enter the steady state after $t \geq 3.5$ s while those with the SMRAC fail to reach the steady state. Finally, according to the simulation data obtained for the tracking errors, as shown in Table 2, the maximum tracking errors (MTE) for the DASCs are 3.0340 mm, 3.0347 mm, 3.3442 mm, 3.2212 mm, 2.93 mm, and 3.3394 mm for the six actuator legs while the MTEs for the SMRAC are 6.2 mm, 6.1 mm, 6.2 mm, 5 mm, 4.8 mm, and 6.3 mm.

Table 2. Errors for SMRAC and DASCs.

	Maximum Errors		Average Errors	
	SMRAC	DASCs	SMRAC	DASCs
X(mm)	4.6	0.9602	0.508	0.0226
Y(mm)	1.8	1.63	0.501	0.0284
Z(mm)	5.1	3.37	0.28	0.0947
q_{e1} (mm)	6.2	3.0340	0.379	0.0988
q_{e2} (mm)	6.1	3.0347	0.323	0.0905
q_{e3} (mm)	6.2	3.3442	0.318	0.0909
q_{e4} (mm)	5	3.2212	0.375	0.0972
q_{e5} (mm)	4.8	2.93	0.331	0.0832
q_{e6} (mm)	6.3	3.3394	0.341	0.0860
ε_1 (mm)	1.8325	0.9329	0.5109	0.0021
ε_2 (mm)	6.1393	0.9292	0.6637	0.0122
ε_3 (mm)	4.3010	1.4477	0.6245	0.0156
ε_4 (mm)	2.7799	0.9756	0.7342	0.0176
ε_5 (mm)	4.098	1.47	0.3961	0.0117
ε_6 (mm)	6.6267	0.9329	0.8946	0.0211

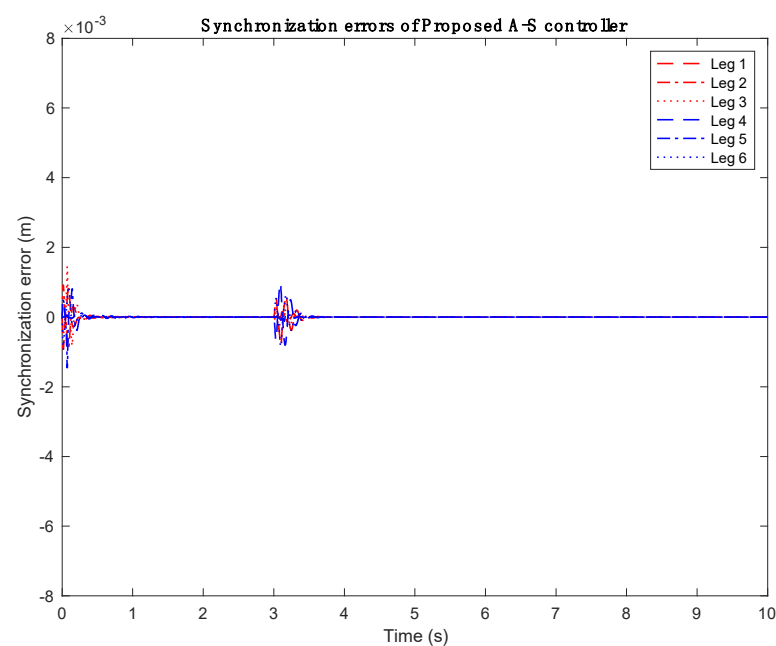


Figure 9. Synchronization errors of DASCs.

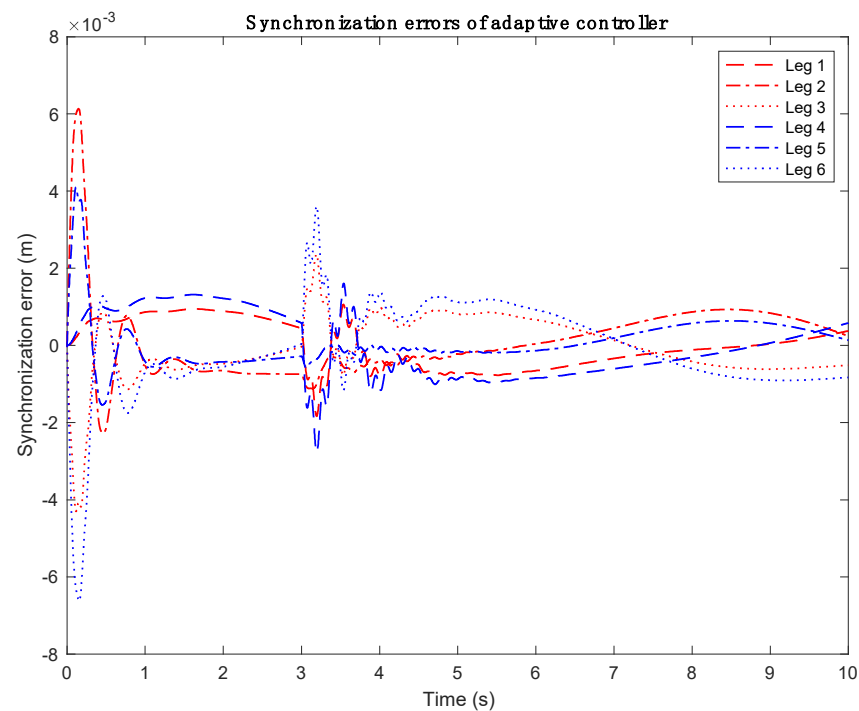


Figure 10. Synchronization errors of the SMRAC.

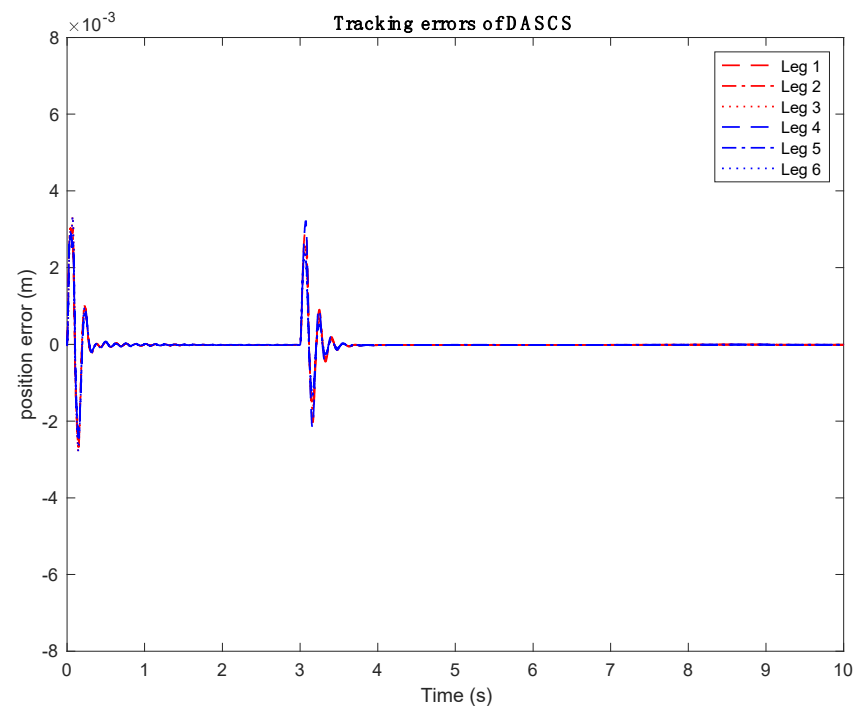


Figure 11. Tracking error of the DASCs.

Figures 13 and 14 show the time trajectories of the tracking of the two control schemes in the X-Y plane and the X-Y-Z space, respectively. As the figures show, after the introduction of the sudden payload change, the SMRAC gets off track farther than the DASCs. Moreover, the DASCs has a better ability to track the motion with a much smaller deviation from the desired trajectory until the end of the motion as compared to the SMRAC. We think that the DASCs has a better tracking ability than that of the SMRAC because it employs the configuration error control via the cross-coupling error control.

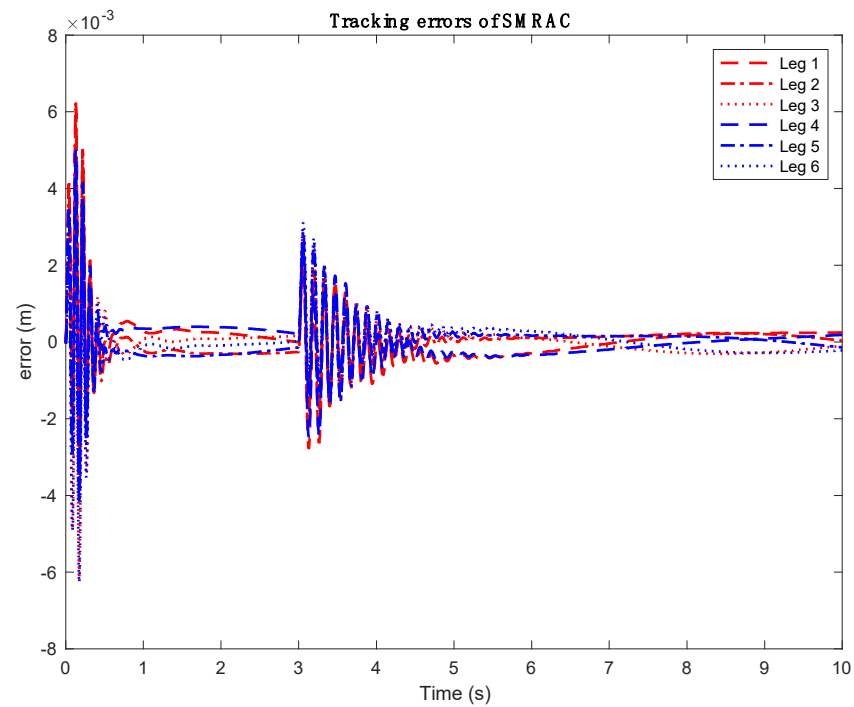


Figure 12. Tracking errors of the SMRAC.

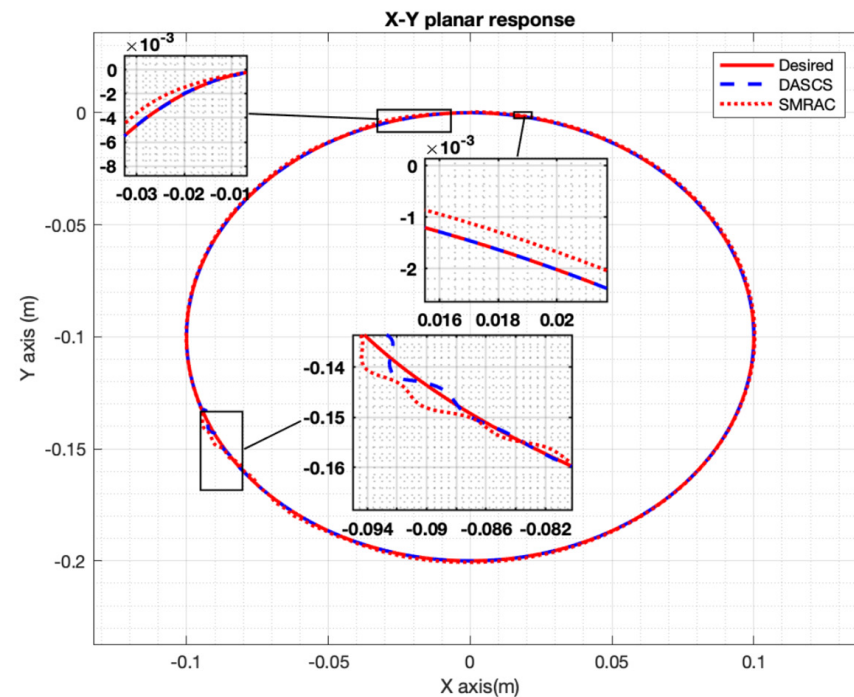


Figure 13. Motion tracking in X-Y plane.

In conclusion, from the above results of the computer simulation study, overall, the DASCS outperforms the SMRAC when controlling the tracking and synchronization errors due to the fact that the SMRAC only reacts to tracking errors while the DASCS not only reacts to the tracking errors but also to the synchronization errors. In other words, the DASCS controls each active leg to follow its desired trajectory as close as possible while synchronizing its motion with that of other active legs in a defined synchronous manner. We believe that the SMRAC does not effectively react to the sudden change in payload because it lacks the synchronization error control that the DASCS has.

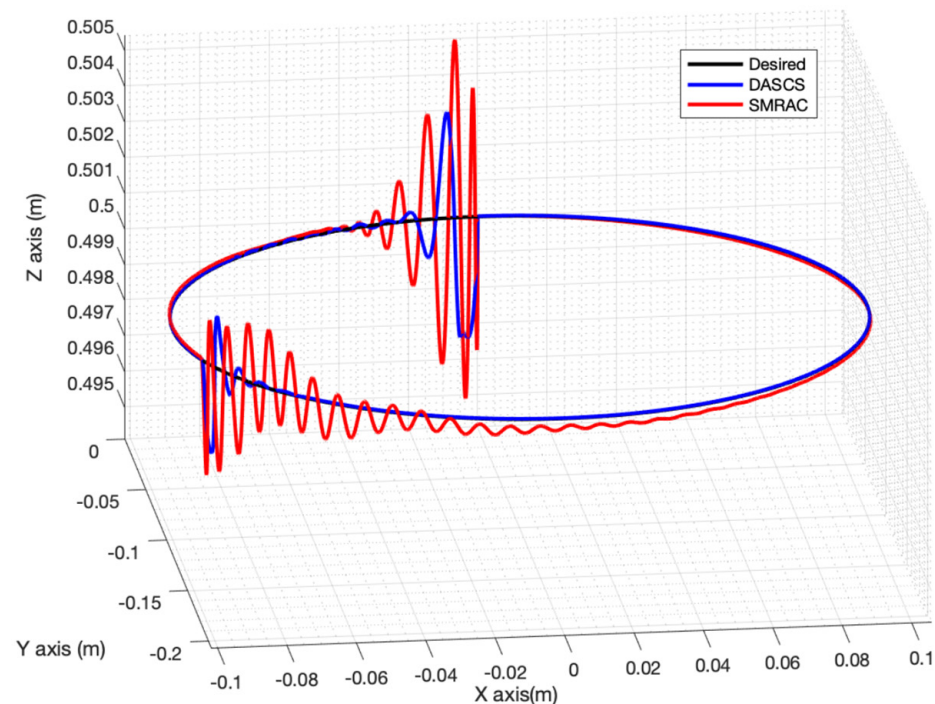


Figure 14. Motion tracking in 3-D (X-Y-Z).

5. Conclusions

This paper considered the problem of motion control of CKCM manipulators because they have several advantages over OKCM manipulators such as higher stiffness, better positioning accuracy, greater payload, and better stability. Solving control problems for CKCMs would result in higher efficiency and productivity of production processes. By treating the trajectory tracking control of a CKCM manipulator as synchronized control of multi OKCM manipulators, we developed a new decentralized adaptive synchronized control scheme, called DASCS, for CKCM manipulators. The development of the DASCS was based on the decentralized concept so that each joint actuator has its own independent controller working with two neighboring ones to achieve the desired motion for the manipulator. In other words, in contrast to conventional decentralized control schemes, each local controller of this new control scheme receives joint position feedback not only from its own joint but also from two neighboring ones. We then applied the model reference adaptive control (MRAC) method based on the Lyapunov direct method into the above synchronized control method to achieve the global asymptotic convergence of tracking errors to zero while synchronizing all active joints' motions. It is important to note that while conventional MRAC schemes act on joint errors alone, the developed DASCS not only acts on joint errors via synchronization errors but also on cross-coupling errors, which mainly contributes to its enhanced performance as compared to other conventional schemes. Moreover, the developed DASCS, whose controller gains are updated by an adaptation law without the burden of calculating the manipulator dynamics, is computationally efficient and thus is suitable for real-time applications.

To comparatively evaluate the performance of the developed DASCS, we conducted a computer simulation study on a 6-DOF CKCM manipulator whose motions were controlled by the DASCS and a MRAC scheme developed by Seraji [30], namely SMRAC. Simulation results confirmed that the DASCS had better tracking performance for tracking desired motions with and without sudden payload changes as compared to the SMRAC. We concluded that the main reason that the DASCS outperformed the SMRAC was the application of synchronized error control by the DASCS.

Future work on this new control scheme is recommended as follows.

- Experimental investigation of the performance of DASC on real CKCM manipulators in comparison with other conventional MRAC schemes.
- Application of DASCs on OKCM manipulators and evaluation of its performance using computer simulation and experimentation.
- Combination of DASC with other intelligent control methods including fuzzy control, neuro fuzzy control, and neural network control to improve performance.
- Addressing robustness and resilience of synchronized error control of robot manipulators.

Author Contributions: Conceptualization, T.T.N., C.C.N. and T.M.N.; methodology, T.T.N., C.C.N. and T.M.N.; software, T.T.N., C.C.N. and T.M.N.; validation, T.T.N., C.C.N. and T.M.N.; results analysis, T.T.N., C.C.N. and T.M.N.; investigation, T.T.N., C.C.N., T.M.N., T.T.C.D., H.T.T.N. and L.S.; data curation, T.T.N., C.C.N., T.M.N., T.T.C.D., H.T.T.N. and L.S.; writing—original draft preparation, T.T.N., C.C.N. and T.M.N.; writing—review and editing, T.T.N., C.C.N., T.M.N. and T.T.C.D.; visualization, T.T.N., C.C.N., T.M.N., T.T.C.D., H.T.T.N. and L.S.; supervision, C.C.N. and L.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article..

Conflicts of Interest: All authors announce that they have no conflicts of interest in relation to the publication of this article.

References

1. Bi, Z.M.; Lin, Y.; Zhang, W.J. The general architecture of adaptive robotic systems for manufacturing applications. *Robot. Comput.-Integr. Manuf.* **2010**, *26*, 461–470. [\[CrossRef\]](#)
2. Wu, F.X.; Zhang, W.J.; Li, Q.; Ouyang, P.R. Integrated design and PD control of high-speed closed-loop mechanisms. *J. Dyn. Syst. Meas. Control* **2002**, *124*, 522–528. [\[CrossRef\]](#)
3. Merlet, J.P. *Parallel Robots*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006; Volume 128.
4. Dasgupta, B.; Mruthyunjaya, T.S. The Stewart platform manipulator: A review. *Mech. Mach. Theory* **2000**, *35*, 15–40. [\[CrossRef\]](#)
5. Stewart, D. A Platform with Six Degrees of Freedom. *Proc. Inst. Mech. Eng.* **1965**, *180*, 371–386. [\[CrossRef\]](#)
6. Tsai, L.W. *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*; John Wiley & Sons: Hoboken, NJ, USA, 1999.
7. Yanna, Y.; Huiying, W.; Lu, S.; Wei, H. The Influence of Road Geometry on Vehicle Rollover and Skidding. *Int. J. Environ. Res. Public Health* **2020**, *17*, 1648. [\[CrossRef\]](#)
8. Sun, L.; Zhao, X. Coupled Dynamics of Vehicle-Bridge Interaction System Using High Efficiency Method. *Adv. Civ. Eng.* **2021**, *2021*, 1964200. [\[CrossRef\]](#)
9. Kelly, R.; Davila, V.S.; Perez, J.A.L. *Control of Robot Manipulators in Joint Space*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006.
10. Koren, Y. Cross-coupled biaxial computer control for manufacturing systems. *J. Dyn. Syst. Meas. Control* **1980**, *102*, 265–272. [\[CrossRef\]](#)
11. Li, Y.; Nielsen, C. Position synchronized path following for a mobile robot and manipulator. In Proceedings of the 52nd IEEE Conference on Decision and Control, Firenze, Italy, 10–13 December 2013; pp. 3541–3546.
12. Do, K.D. Synchronization motion tracking control of multiple underactuated ships with collision avoidance. *IEEE Trans. Ind. Electron.* **2016**, *63*, 2976–2989. [\[CrossRef\]](#)
13. Bouteraa, Y.; Ghommam, J.; Poisson, G.; Derbel, N. Distributed synchronization control to trajectory tracking of multiple robot manipulators. *J. Robot.* **2011**, *2011*, 652785. [\[CrossRef\]](#)
14. Wang, C.; Sun, D. A synchronization control strategy for multiple robot systems using shape regulation technology. In Proceedings of the 2008 7th World Congress on Intelligent Control and Automation, Chongqing, China, 25–27 June 2008; pp. 467–472.
15. Sun, D.; Shao, X.; Feng, G. A model-free cross-coupled control for position synchronization of multi-axis motions: Theory and experiments. *IEEE Trans. Control Syst. Technol.* **2007**, *15*, 306–314. [\[CrossRef\]](#)

16. Ouyanga, P.R.; Dam, T.; Huang, J.; Zhang, W.J. Contour tracking control in position domain. *Mechatronics* **2012**, *22*, 934–944. [\[CrossRef\]](#)
17. Shang, W.; Cong, S.; Ge, Y. Coordination motion control in the task space for parallel manipulators with actuation redundancy. *IEEE Trans. Autom. Sci. Eng.* **2012**, *10*, 665–673. [\[CrossRef\]](#)
18. Sun, D.; Wang, C.; Shang, W.; Feng, G. A synchronization approach to trajectory tracking of multiple mobile robots while maintaining time-varying formations. *IEEE Trans. Robot.* **2009**, *25*, 1074–1086.
19. Sun, D.; Mills, J.K. Adaptive synchronized control for coordination of two robot manipulators. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292), Washington, DC, USA, 11–15 May 2002; Cat. No. 02CH37292; Volume 1, pp. 976–981.
20. Zhu, W.H. On adaptive synchronization control of coordinated multi robots with flexible/rigid constraints. *IEEE Trans. Robot.* **2005**, *21*, 520–552.
21. Wang, H. Task-space synchronization of networked robotic systems with uncertain kinematics and dynamics. *IEEE Trans. Autom. Control* **2013**, *58*, 3169–3174. [\[CrossRef\]](#)
22. Long, Y.; Yang, X.J. Robust adaptive fuzzy sliding mode synchronous control for a planar redundantly actuated parallel manipulator. In Proceedings of the 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), Guangzhou, China, 11–14 December 2012; pp. 2264–2269.
23. Fuh, C.C.; Tsai, H.H.; Huang, C.C.A. fuzzy cross-coupled linear quadratic regulator for improving the contour accuracy of bi-axis machine tools. In Proceedings of the 48th IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference, Shanghai, China, 15–18 December 2009; pp. 4156–4161.
24. Zhao, D.; Zhu, Q.; Li, N.; Li, S. Neural networked based synchronized control for multiple robotic manipulators. In Proceedings of the 10th IEEE International Conference on Control and Automation (ICCA), Hangzhou, China, 12–14 June 2013; pp. 1950–1955.
25. Yang, G.; Yao, J. Multilayer neurocontrol of high-order uncertain nonlinear systems with active disturbance rejection. *Int. J. Robust Nonlinear Control* **2024**, *34*, 2972–2987. [\[CrossRef\]](#)
26. Le, D.K.; Ahn, K.K. Synchronization algorithm for controlling 3-R planar parallel pneumatic artificial muscle robot. In Proceedings of the 2011 11th International Conference on Control, Automation and Systems, Gyeonggi-do, Republic of Korea, 26–29 October 2011; pp. 1588–1593.
27. Su, Y.X.; Sun, D.; Ren, L.; Wang, X.; Mills, J.K. Nonlinear PD synchronized control for parallel manipulators. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 1374–1379.
28. Sun, D.; Tong, M.C. A synchronization approach for the minimization of contouring errors of CNC machine tools. *IEEE Trans. Autom. Sci. Eng.* **2009**, *6*, 720–729.
29. Zhao, D.; Li, S.; Gao, F. Fully adaptive feedforward feedback synchronized tracking control for Stewart Platform systems. *Int. J. Control Autom. Syst.* **2008**, *6*, 689–701.
30. Seraji, H. Decentralized adaptive control of manipulators: Theory, simulation, and experimentation. *IEEE Trans. Robot. Autom.* **1989**, *5*, 183–201. [\[CrossRef\]](#)
31. Sun, D.; Mills, J.K. Adaptive Synchronized Control for Coordination of Multirobot Assembly Tasks. *IEEE Trans. Robot. Autom.* **2002**, *18*, 498–510.
32. Sun, D. Position synchronization of multiple motion axes with adaptive coupling control. *Automatic* **2003**, *39*, 997–1005. [\[CrossRef\]](#)
33. Craig, J. *Robotics: Mechanics and Control*; Addison-Wesley: Reading, MA, USA, 1986.
34. Slotine, J.J.E.; Li, W. On the adaptive control of robot manipulators. *Int. J. Robot. Res.* **1987**, *6*, 49–59. [\[CrossRef\]](#)
35. O’Searcoid, M. *Metric Spaces*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006.
36. Khalil, H.K.; Grizzle, J.W. *Nonlinear Systems*; Prentice Hall: Upper Saddle River, NJ, USA, 2002; Volume 3.
37. Nguyen, C.C.; Zhou, Z.; Antrazi, S.S.; Campbell, C.E. Efficient computation of forward kinematics and Jacobian matrix of a Stewart platform-based manipulator. In Proceedings of the IEEE Proceedings of the SOUTHEASTCON ’91, Williamsburg, VA, USA, 7–10 April 1991; Volume 2, pp. 869–874.
38. Merlet, J.P. Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis. *Int. J. Robot. Res.* **2004**, *23*, 221–235. [\[CrossRef\]](#)

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.