machines

MDPI

Article

Real-Time Experiments for Decentralized Adaptive Synchronized Motion Control of a Closed-Kinematic Chain Mechanism Robot Manipulator

Charles C. Nguyen [1,*], Tri T. Nguyen [2], Tu T. C. Duong [1,3], Tuan M. Nguyen [1], Ha T. T. Ngo [1] and Lu Sun [4]

[1] Department of Electrical Engineering, The Catholic University of America, Washington, DC 20064, USA; duongt@cua.edu or tu.duong@hcmute.edu.vn (T.T.C.D.); 11nguyen@cua.edu (T.M.N.); ngot@cua.edu (H.T.T.N.)
[2] Department of Electrical Engineering, Danang University of Science and Technology, Danang City 50000, Vietnam; 65nguyent@cua.edu
[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, NY 14623, USA; lxsite@rit.edu
* Correspondence: nguyen@cua.edu

Abstract

This paper presents the results of real-time experiments conducted to evaluate the performance of a developed adaptive control scheme applied to control the motion of a real closed-kinematic chain mechanism (CKCM) robot manipulator with two degrees of freedom (DOFs). The developed control scheme, referred to as the decentralized adaptive synchronized control scheme (DASCS), was the result of the combination of model reference adaptive control (MRAC) based on the Lyapunov direct method and the synchronization technique. CKCM manipulators were considered in the experimental study due to their advantages over their open-kinematic chain mechanism (OKCM) manipulator counterparts, such as higher stiffness, better stability, and greater payload. The conducted computer simulation study showed that the DASCS was able to asymptotically converge tracking errors to zero, with all the active joints moving synchronously in a prescribed way. One of the important properties of the DASCS is the independence of robot manipulator dynamics, making it computationally efficient and therefore suitable for real-time applications. The present paper reports findings from experiments in which the DASCS was applied to control the above manipulator and carry out various paths. The DASCS's performance was compared with that of a traditional adaptive control scheme, namely the SMRACS, when both schemes were applied to track the same paths.

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

check for updates

Received: 24 June 2025
Revised: 22 July 2025
Accepted: 23 July 2025
Published: 25 July 2025

Citation: Nguyen, C.C.; Nguyen, T.T.; Duong, T.T.C.; Nguyen, T.M.; Ngo, H.T.T.; Sun, L. Real-Time Experiments for Decentralized Adaptive Synchronized Motion Control of a Closed-Kinematic Chain Mechanism Robot Manipulator. *Machines* **2025**, *13*, 652. https://doi.org/10.3390/machines13080652

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

According to their kinematic structure, robot manipulators can be divided into two categories: open-kinematic chain mechanism (OKCM) manipulators and closed-kinematic chain mechanism (CKCM) manipulators [1]. OKCM manipulators consist of a series of links interconnected by joints and extend from a base to an end-effector. This type of manipulator has a long reach and a large workspace. However, it has several disadvantages,

such as low rigidity and poor performance at high speed and high loading conditions. CKCM manipulators, also well known as parallel manipulators, in general consist of a moving platform and a fixed base platform interconnected by several links. In these manipulators, all links act in a parallel manner and share the same payload. Therefore, they can improve their tracking performance by avoiding accumulated and amplified errors from link to link [2–4], as occur in the case of OKCM manipulators. As a result, they are superior to OKCM manipulators in applications that require high force/torque capability, high position accuracy, fast motion, and large payload-carrying capacity in a relatively small workspace for tasks including arc welding, cutting, sealing, medical rehabilitation, high-speed assembly operation, and microchip assembly operation [3–6].

Despite the advantages presented above, the complexity of the dynamical model and the closed-kinematic structure of CKCM manipulators make the development of controllers for motion control very challenging. Control schemes developed for CKCM manipulators can be classified into two main groups, namely model-based and performance-based methods [7]. Model-based control methods such as the computed torque method [8] require precise manipulator dynamics in their control law formulation. Consequently, the performance of these control methods can be deteriorated due to imprecise mathematical manipulator models. In addition, the implementation of model-based control schemes is computationally intensive due to the required computation of complicated manipulator dynamics. On the other hand, performance-based control schemes such as adaptive control techniques do not need the complex mathematical model of manipulator dynamics in their control law formulation. As a result, these control schemes are very computationally efficient and suitable for real-time applications. One of the well-known adaptive control techniques is model reference adaptive control (MRAC) [9], the main idea of which is to obtain a closed-loop system with adjustable controller parameters capable of changing the behavior of the closed-loop system. The main purpose of MRAC is to force the controlled system to act like a desired reference model to effectively deal with uncertainties and changes in the system or its environment [10]. While numerous MRAC-based control schemes have been developed for OKCM manipulators [11,12], only a few control schemes have been proposed for CKCM manipulators. In [13], an MRAC-based adaptive control scheme was developed for motion control of a 6-DOF parallel manipulator built as a prototype to study in space operations such as robotic maintenance and repair of spacecrafts. This developed adaptive control scheme included a proportional-derivative (PD) feedback controller with gains adjusted by an adaptation law. Using the Lyapunov direct method, the authors derived an adaptation law that stabilized the error dynamics. Furthermore, neural networks were employed to control the motion of parallel robot manipulators. A decentralized proportional integral derivative (PID) controller in Cartesian space was enhanced by an artificial neural network to improve the tracking performance of a 2-DOF redundant parallel manipulator [14].

Unlike OKCM manipulators, one unique property of CKCM manipulators is that due to their closed-kinematic structure, the motion of each of their active joints is affected by that of other active joints. However, in most conventional controllers for CKCM manipulators, the control loops of the actuators do not communicate with each other, and as a result, each control loop only handles errors of its own actuator regardless of those of other actuators. Thus, synchronization among actuators is needed to avoid a reduction in motion control performance. In addition, because the actuators of CKCM manipulators are arranged in parallel, the lack of synchronization might produce large interactional forces, which in turn could wear out the actuators prematurely. In the search for a solution for the above synchronization issue, the cross coupling control method came up as a potential candidate. A synchronization technique based on the concept of cross coupling control

was designed to tackle the above issue by letting each control loop receive feedback from itself as well as from the others to achieve better coordination and improve the manipulator performance significantly [15]. Other studies considering cross coupling errors to improve the synchronization performance of two-axis motions can be found in [16–18]. Synchronized control schemes with control law implementation that do not require knowledge of the manipulator dynamics were developed by researchers in [19–21]. Recently, researchers in [22] developed a new adaptive control scheme for CKCM robot manipulators, which combined both the concepts of synchronization control and MRAC, called the decentralized adaptive synchronized control scheme (DASCS). In [22], computer simulation results obtained showed that the DASCS had the ability to effectively control a 6-DOF CKCM manipulator without knowledge of the manipulator dynamics and thus was suitable for real-time implementation. Focusing on the experimental aspect of the DASCS, the present paper will report and discuss results of real-time experiments conducted to evaluate the performance of the DASCS in comparison with that of a traditional MRAC scheme developed by Seraji in [7], namely the SMRACS. The specific contributions of the present paper include an experimental investigation of the effectiveness of the DASCS for a 2-DOF CKCM manipulator and an experimental evaluation of its control performance in comparison to that of a traditional adaptive control scheme, namely SMRACS. Furthermore, this study experimentally proves that the incorporation of synchronized control into an adaptive control scheme enhances the control performance of the original control scheme when applied to control CKCM manipulators.

The present paper is organized as follows: Section 2 describes the testbed employed in our experimental investigation of the DASCS. Section 3 gives an overview of the development of the DASCS consisting of the synchronization control method and the MRAC method. Section 4 presents and discusses the results of real-time experiments conducted to evaluate the control performance of the DASCS in comparison with the SMRACS. Section 5 concludes the paper with a summary of the study and recommendations for future research.

2. The Control Scheme Testbed

The testbed used to evaluate the control performance of the DASCS consisted of a real CKCM robot manipulator possessing two DOFs (Figure 1) [23] and its closed-loop control system (Figure 2). This manipulator was designed and built to prove a proposed concept implying that CKCM robot manipulators were capable of performing high-precision motions for potential space applications such as assembling and maintaining NASA spacecrafts. Figure 1 shows that the manipulator is composed of two links, each of which consists of a ball screw linear actuator activated by a DC motor. Two pin joints acting as revolute joints mount the top end of each link under a fixed platform. The same type of revolute joints are used to couple the other ends of the links, on which a tool can be mounted to carry out robotic tasks. The actual lengths of the two actuators are measured by two linear voltage differential transformers (LVDTs) mounted along the links. The LVDTs serve as joint position sensors to provide real-time information on actual link lengths for feedback control. Table 1 contains the specifications of the ball screw actuators and the manipulator parameters, and Table 2 contains the specifications of the LVDTs.

The control scheme testbed is presented in a block diagram in Figure 2, where a data acquisition system (DAQ) serves as an interface between a computer and the outside world composed of the manipulator, LVDTs as sensors, and the actuator drivers. LabView version 2019, which is a graphical programming environment, allows for data acquisition and analysis to be performed in real time. As illustrated in Figure 3, the two LVDTs provide real-time information on the manipulator lengths as joint variables $q_1(t)$ and $q_2(t)$, which are fed to the DAQ. The DAQ provides joint variables $q_1(t)$ and $q_2(t)$ to LabView, which

in turn carries out all necessary computation for the DASCS. The DAQ then takes $T_1(t)$ and $T_2(t)$, computed by the DASCS adaptation law as developed in Section 2 below, and sends them to the respective DC motors and actuators. $T_1(t)$ and $T_2(t)$, which are low-level control signals expressed in DC voltages (in mV), are sent to a DC amplifier, which in turn amplifies the signals to higher-level DC voltages (from −24 V to +24 V) to reach an appropriate voltage level to activate the DC motors.

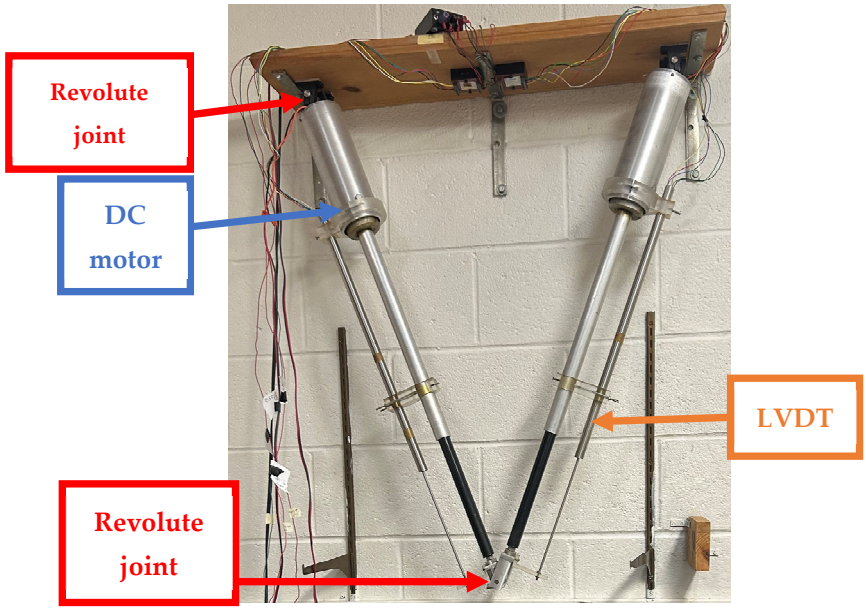


Figure 1. The CKCM manipulator of the control scheme testbed.

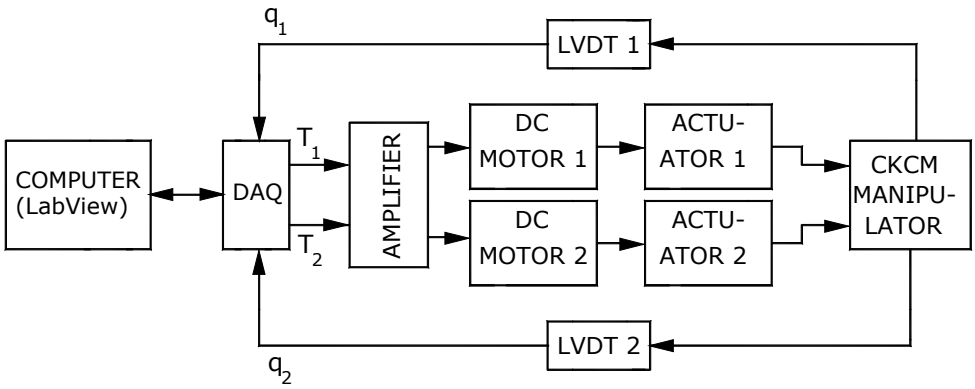


Figure 2. Block diagram of the control scheme testbed.

Figure 3 shows the frame assignment of the manipulator, where the lengths of its links, $q_1(t)$ and $q_2(t)$, are denoted as its joint variables. A fixed Reference Frame $\{X_{ref}, Y_{ref}, O_{ref}\}$ is assigned using the right-hand rule so that its origin O_{ref} coincides with the revolute joint of Actuator 1; X_{ref}-Axis aligns with the revolute joints of Actuator 1 and Actuator 2, and Y_{ref}-Axis is perpendicular to X_{ref}-Axis. The variables x(t) and y(t), representing the horizontal and vertical positions of the manipulator end-effector with respect to the Reference Frame, respectively, are the Cartesian variables of the manipulator. In addition, a fixed Home Frame $\{X_h, Y_h, O_h\}$ is assigned using the right-hand rule so that its origin O_h is at the center of the robot end-effector when the manipulator is in the home position. We note that the X_h-Axis and the Y_h-Axis of the Home Frame are parallel to the X_{ref}-Axis and the Y_{ref}-Axis of the Reference Frame, respectively. Any position in the Home Frame can be transformed to the Reference Frame by a fixed homogeneous transformation. The

Home Frame will be used in the next section to represent the data collected by LabView via the DAQ.

Table 1. Actuator specifications and manipulator parameters.

Manipulator Parameters	Description	Value	Unit
m	Total mass of links	4.91	kg
m_1	Mass of link moving part	0.59	kg
d	Distance between the top ends	0.74	m
l_s	Fixed length of links	0.26	m
G	Gravitational acceleration constant	9.81	m/s^2
	Minimum length of links	0.84	m
	Maximum length of links	1.22	m
Actuator Specifications	**Description**	**Value**	**Unit**
	Motor	Permanent magnet	
	Voltage	24 VDC	VDC
	Speed	3000 RPM	RPM
	Diameter	0.076	m
	Stroke	0.406	m
	Gear reduction	10:1	
	Max velocity	1.78	cm/second

Table 2. LVDT specifications.

LVDT	Description	Value	Unit
	Linear range	±7.5	inch
	Linearity	Best-fit straight line	
	Resolution	Infinite (theoretically)	
	Input	±14.5 to ±28 VDC, ±100 mA	
	Output	±5	VDC
	Operating temperature range	−67 to 257	F

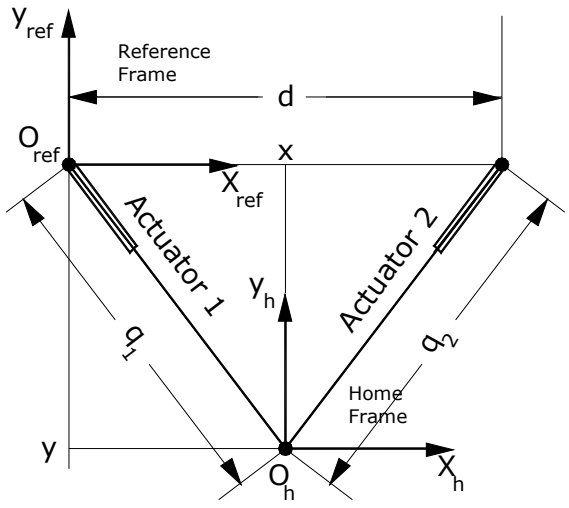


Figure 3. Frame assignment of the planar 2-DOF CKCM robot manipulator.

We proceed to derive kinematic solutions for the manipulator. From Figure 3, the joint variables $q_1(t)$ and $q_2(t)$ of Actuator 1 and Actuator 2, respectively, can be computed from the Cartesian variables $x(t)$ and $y(t)$ by

$$q_1 = \sqrt{x^2 + y^2} \tag{1}$$

and

$$q_2 = \sqrt{(d-x)^2 + y^2} \tag{2}$$

where d is the distance between the top ends of the links, which are at the centers of the pin joints suspending the two actuators.

By definition, Equations (1) and (2) provide a closed-form solution for the manipulator inverse kinematics, meaning that they can be used to calculate the joint variables $q_1(t)$ and $q_2(t)$ from the Cartesian variables x(t) and y(t).

In addition, from Equations (1) and (2), the Cartesian variables x(t) and y(t) can be derived as follows:

$$x = \frac{q_1^2 - q_2^2 + d^2}{2d} \tag{3}$$

$$y = \frac{-\sqrt{4d^2q_1^2 - \left(q_1^2 - q_2^2 + d^2\right)}}{2d} \tag{4}$$

By definition, Equations (3) and (4) represent a closed-form solution of the manipulator forward kinematics, allowing the determination of the Cartesian variables x(t) and y(t) from the joint variables $q_1(t)$ and $q_2(t)$.

3. Overview of the DASCS

The complete and detailed development of the DASCS can be found in [22]. Since this paper focuses on the experimental evaluation of the above control scheme, this section contains only relevant mathematical expressions and equations for the adaptive control law of the developed control scheme. From [22], a block diagram of the DASCS with its controllers is presented in Figure 4, where each ith active joint of the above manipulator is controlled by a separate controller in a decentralized structure. The joint variable error $q_{ei}(t)$ is computed as

$$q_{ei}(t) = q_{di}(t) - q_i(t) \tag{5}$$

where q_{di} and $q_i(t)$ denote the desired and actual time trajectory of the i^{th} active joint, respectively. The Cross Coupling Error Calculation Block calculates the command signal $u_i(t)$ as

$$u_i(t) = q_{di}(t) + \beta \int_0^t [\varepsilon_n(w) - \varepsilon_{n-1}(w)]dw + \alpha \int_0^t {e_i}^*(w)dw \tag{6}$$

and generates the generalized error $r_i(t)$ as

$$r_i(t) = u_i(t) - q_i(t) = {e_i}^*(t) + \alpha \int_0^t {e_i}^*(w)dw \tag{7}$$

where the cross coupling error ${e_i}^*(t)$ is determined by

$${e_i}^*(t) = q_{ei}(\mathrm{t}) + \beta \int_0^t [\varepsilon_i(w) - \varepsilon_{i-1}(w)]dw \tag{8}$$

and the synchronization error $\varepsilon_i(t)$ is given by

$$\varepsilon_i(t) = p_i(t)q_{ei}(t) - p_{i+1}(t)q_{ei+1}(t) \tag{9}$$

where $p_i(\mathrm{t}) = \frac{1}{q_{di}(t)}$.

From [22], the control signal $T_i(t)$ that is sent to the ith joint actuator is given by

$$T_i(t) = f_i(t) + \left[k_{0i}(t)r_i(t) + k_{1i}(t)\dot{r}_i(t)\right] + \left[a_i(t)u_i(t) + b_i(t)\dot{u}_i(t) + c_i(t)\ddot{u}_i(t)\right] \tag{10}$$

We see that $T_i(t)$, representing the adaptive control law, is composed of three terms:

- The first term $f_i(t)$, produced by the Auxiliary Signal Computation Block represents the auxiliary signal to partly compensate for disturbance and to improve tracking performance.
- The second term $\tau_i^{fb}(t) = \left[k_{0i}(t)r_i(t) + k_{1i}(t)\dot{r}_i(t)\right] = \left[k_{0i}(t) + \alpha\right]e_i{}^*(t) + \alpha k_{0i}(t)\int_0^t e_i{}^*(w)dw] + k_{1i}(t)\dot{e_i}{}^*(t)$ represents the PID feedback controller. This term is generated by the PID Feedback Controller Block.
- The last term $\tau_i^{ff}(t) = \left[a_i(t)u_i(t) + b_i(t)\dot{u}_i(t) + c_i\ddot{u}_i(t)\right]$ represents the feedforward controller and is computed by the Feedforward Controller Block.

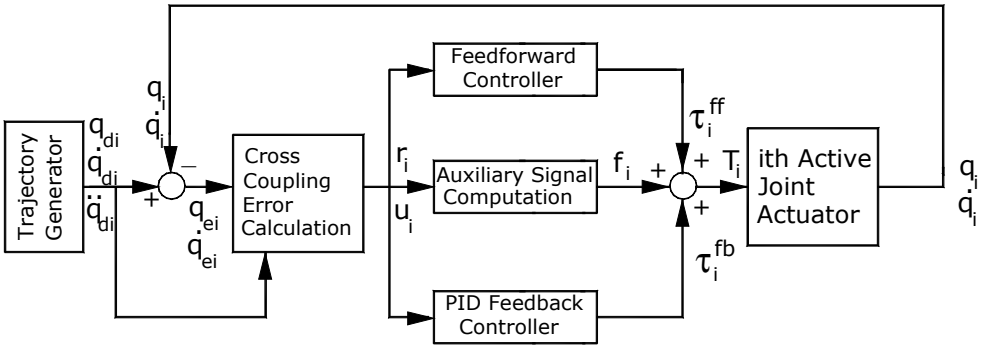


Figure 4. The decentralized adaptive synchronized control scheme.

Various functions and parameters of the above terms can be determined by

$$\begin{aligned} f(t) &= f(0) + \eta_2\Omega(\mathrm{t}) + \eta_1\textstyle\int_0^t \Omega(\mathrm{w})\mathrm{dw} \\ k_0(t) &= k_0(0) + \gamma_2\Omega(\mathrm{t})\mathrm{r}(t) + \gamma_1\textstyle\int_0^t \Omega(\mathrm{w})\mathrm{r}(\mathrm{w})\mathrm{dw} \\ k_1(t) &= k_1(0) + \lambda_2\Omega(\mathrm{t})\dot{r}(t) + \lambda_1\textstyle\int_0^t \Omega(\mathrm{w})\dot{r}(\mathrm{w})\mathrm{dw} \\ a(t) &= \mathrm{a}(0) + \mu_2\Omega(\mathrm{t})\mathrm{u}(\mathrm{t}) + \mu_1\textstyle\int_0^t \Omega(\mathrm{w})\mathrm{u}(\mathrm{w})\mathrm{dw} \\ b(t) &= b(0) + \rho_2\Omega(\mathrm{t})\dot{u}(t) + \rho_1\textstyle\int_0^t \Omega(\mathrm{w})\dot{u}(w)\mathrm{dw} \\ c(t) &= c(0) + \sigma_2\Omega(\mathrm{t})\ddot{u}(t) + \sigma_1\textstyle\int_0^t \Omega(\mathrm{w})\ddot{u}(\mathrm{w})\mathrm{dw} \end{aligned} \tag{11}$$

where $\eta_1, \gamma_1, \lambda_1, \mu_1, \rho_1, \sigma_1$ are positive constants and $\eta_2, \gamma_2, \lambda_2, \mu_2, \rho_2, \sigma_2$ are zero or positive constants; $\Omega(t) = P_2r(t) + P_3\dot{r}(t)$ where P_2 and P_3 are positive constants. Furthermore, $f(0)$, $k_0(0)$, $k_1(0)$, $a(0)$, $b(0)$, and $c(0)$, which are the initial conditions of $f(t)$, $k_0(t)$, $k_1(t)$, $a(t)$, $b(t)$, and $c(t)$, respectively, can be set arbitrarily. We note that in (11), the subindex i was dropped for presentation simplicity. It is noted that $T_i(t)$, as seen in Equation (10), was a result of applying the Lyapunov direct method in the development of the DASCS and thus ensuring asymptotic stability of the closed-loop feedback control system.

As seen in Figure 4, the desired position, velocity, and acceleration of the active joint, which are generated by the Trajectory Generator, are the required inputs of the ith controller. In addition, the actual positions and velocities of its active joint and of the two adjacent ones are the required measurements for feedback control.

Since the results of experiments conducted to evaluate the performance of the DASCS in comparison to that of SMRACS will be presented and discussed in the next section, it is pertinent now to give an overview of the above control schemes to highlight their differences. Detailed development of the control schemes can be found in [22] for the DASCS and in [7] for the SMRACS. Motivated to alleviate the disadvantages of model-based control techniques, including the heavy computational requirements of manipulator dynamics, reliance on the fidelity of manipulator models, and non-suitability for real-time applications, Seraji in [7] developed an adaptive control scheme, namely the SMRACS, for OKCM manipulators. The development of the SMRACS was based on the concept of MRAC [9], and the Lypunov direct method was employed to ensure the stability of this

control scheme. However, being intended for motion control of OKCM manipulators, the SMRACS lacks the synchronized control of active joints of CKCM manipulators, causing it to have poor control performance when applied to control this type of manipulator. On the other hand, the DASCS developed in [22] has an adaptive control law presented in Equation (10) that is very similar to that of the SMRACS. The main difference between these two adaptive control schemes is that the DASCS incorporates the concept of synchronized control into its control law, as seen in Equations (5)–(9), while the SMRACS uses the conventional joint variable errors. Computer simulation studies conducted in [22] showed that the DASCS had better control performance than the SMRACS when both control schemes were applied to control the same motion of a 6-DOF CKCM manipulator because the DASCS included synchronization error control in its adaptation control law. The next section will prove this finding experimentally.

4. Real-Time Experiments and Results

A computer simulation study conducted to comparatively evaluate the performance of the DASCS and the SMRACS when they were applied to control the motion of a 6-DOF CKCM robot manipulator [22] and of the above 2-DOF CKCM robot manipulator [24] showed that the DASCS performed better than the SMRACS. We now present and discuss the results of completed real-time experiments in which the DASCS and the SMRAC were applied to control the above manipulator to track two different motions, namely a straight-line motion and a circular motion. From the testbed setup seen in Figure 2, the real-time data for the joint variables $q_1(t)$ and $q_2(t)$, measured by the LVDTs, were read by LabView during the experiments. Then, the joint variables were used to compute the Cartesian positions x(t) and y(t) of the manipulator using the forward kinematic Equations (3) and (4). In addition, LabView used them to calculate various variables, including joint variable errors $q_{e1}(t)$ and $q_{e2}(t)$ using Equation (5), command signals $u_1(t)$ and $u_2(t)$ using Equation (6), generalized errors $r_1(t)$ and $r_2(t)$ using Equation (7), cross coupling errors ${e_1}^*(t)$ and ${e_2}^*(t)$ using Equation (8), synchronization errors $\varepsilon_1(t)$ and $\varepsilon_2(t)$ using Equation (9), and finally the control signals $T_1(t)$ and $T_2(t)$ using Equation (10). The control signals $T_1(t)$ and $T_2(t)$ sent to the DC motors of Actuators 1 and 2, respectively, are given by

$$T_1(t) = f_1(t) + \left[k_{01}(t)r_1(t) + k_{11}(t)\dot{r}_1(t)\right] + \left[a_1(t)u_1(t) + b_1(t)\dot{u}_1(t) + c_1(t)\ddot{u}_1(t)\right] \quad (12)$$

and

$$T_2(t) = f_2(t) + \left[k_{02}(t)r_1(t) + k_{12}(t)\dot{r}_2(t)\right] + \left[a_2(t)u_2(t) + b_2(t)\dot{u}_2(t) + c_2(t)\ddot{u}_2(t)\right] \quad (13)$$

where $f_1(t)$ and $f_2(t)$ and the above parameters were computed using Equation (11). Before conducting the experiments, the constants of Equation (11) were selected and set after a few trial-and-error tests to be $\eta_1 = 5, \gamma_1 = 4, \lambda_1 = 6, \mu_1 = 0, \rho_1 = 7, \sigma_1 = 6$ and $\eta_2 = 0, \gamma_2 = 8, \lambda_2 = 0, \mu_2 = 5, \rho_2 = 6, \sigma_2 = 3$. The initial values of the controller gains were chosen to be zero, that is, $k_{01}(0) = k_{11}(0) = k_{02}(0) = k_{12}(0) = 0$.

All graphs in the following experiments are presented with respect to the Home Frame $\{X_h, Y_h, O_h\}$, as seen in Figure 3, and their data can be straightforwardly transformed to the Reference Frame $\{X_{ref}, Y_{ref}, O_{ref}\}$ by a fixed homogeneous transformation.

4.1. Experiment 1: Tracking a Straight-Line Motion

In this experiment, the manipulator end-effector was controlled by the DASCS and the SMRACS to carry out a straight-line motion from (0 mm, 0 mm) to (50.8, 50.8) (mm) in the Home Frame in 15 s. The Trajectory Generator in Figure 4 first formulated the desired

Cartesian motion in terms of $x_d(t)$ and $y_d(t)$ and then employed the manipulator inverse kinematic Equations (1) and (2) to determine the desired joint variables $q_{d1}(t)$ and q_{d2}(t).

Figures 5 and 6 illustrate the time trajectories of the tracking errors $q_{e1}(t)$ (Actuator 1 in solid red line) and $q_{e2}(t)$ (Actuator 2 in dashed blue line) of the DASCS and SMRACS, respectively. As one can see, the time trajectories of that DASCS's tracking errors follow each other very closely, while those of the SMRACS do not. For this case, the DASCS has maximum and average synchronization errors of 0.8913 mm and 0.2828 mm, respectively, while the SMRACS has maximum and average synchronization errors of 1.1559 mm and 0.6897 mm, respectively. Studying the transient and steady-state behaviors of the control schemes, we found that the DASCS has better transient and steady-state responses than the SMRACS. In particular, the average tracking errors of the DASCS and the SMRACS for Actuator (leg) 1 are 0.1786 mm and 0.4335 mm, respectively, resulting in the SMRACS's average tracking errors being 242% of the DASCS's errors. The average tracking errors of the DASCS and the SMRACS for Actuator (leg) 2 are 0.2231 mm and 1.1233 mm, respectively, resulting in the SMRACS's average tracking errors being 503% of the DASCS's errors.

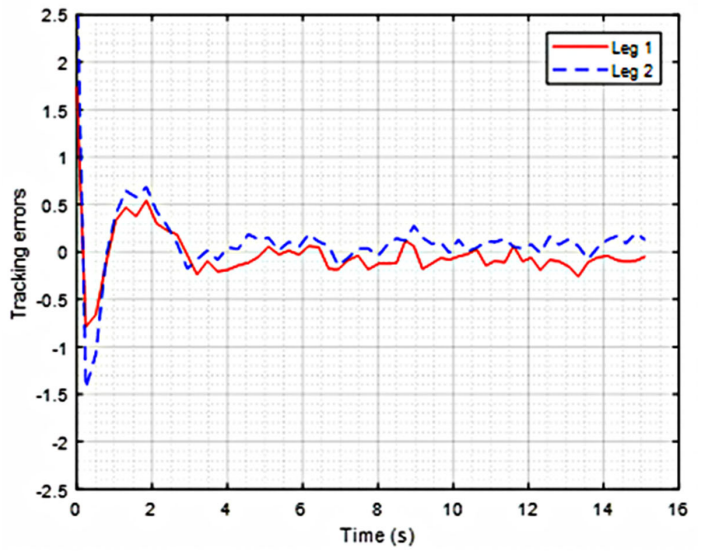


Figure 5. Time trajectories of tracking errors of straight-line motion under the DASCS.

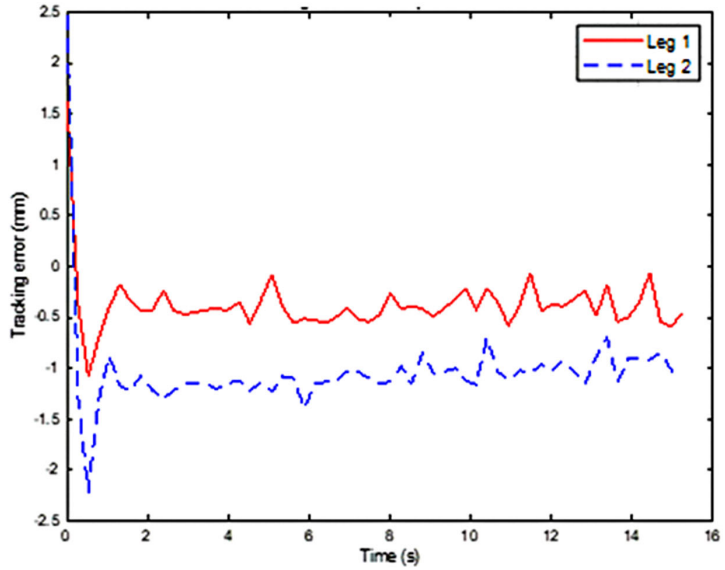


Figure 6. Time trajectories of tracking errors of straight-line motion under the SMRACS.

Figures 7 and 8 show the tracking performance of the DASCS and the SMRACS in the Home Frame, respectively. As one can see, the DASCS provides a better performance in tracking a straight line than the SMRACS.

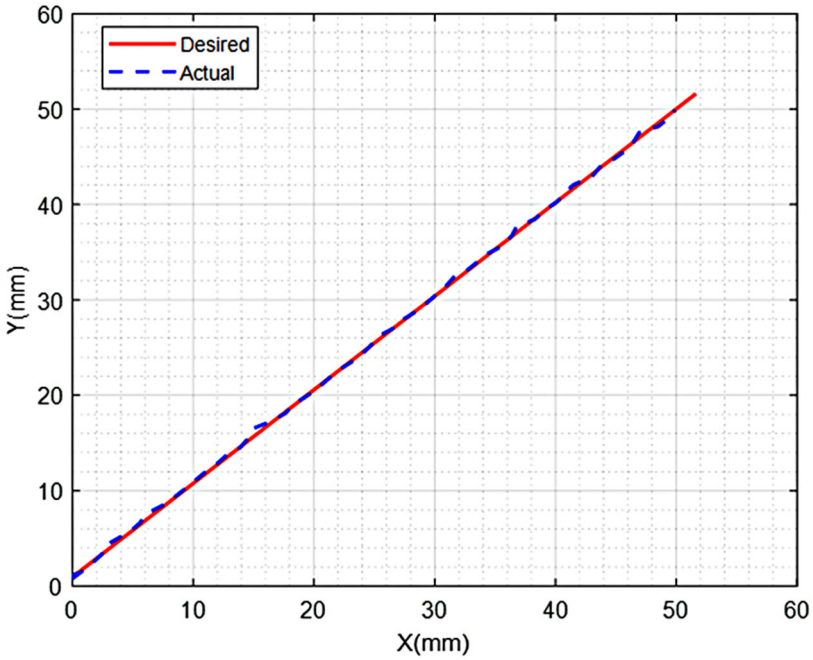


Figure 7. Cartesian straight-line motions under the DASCS.

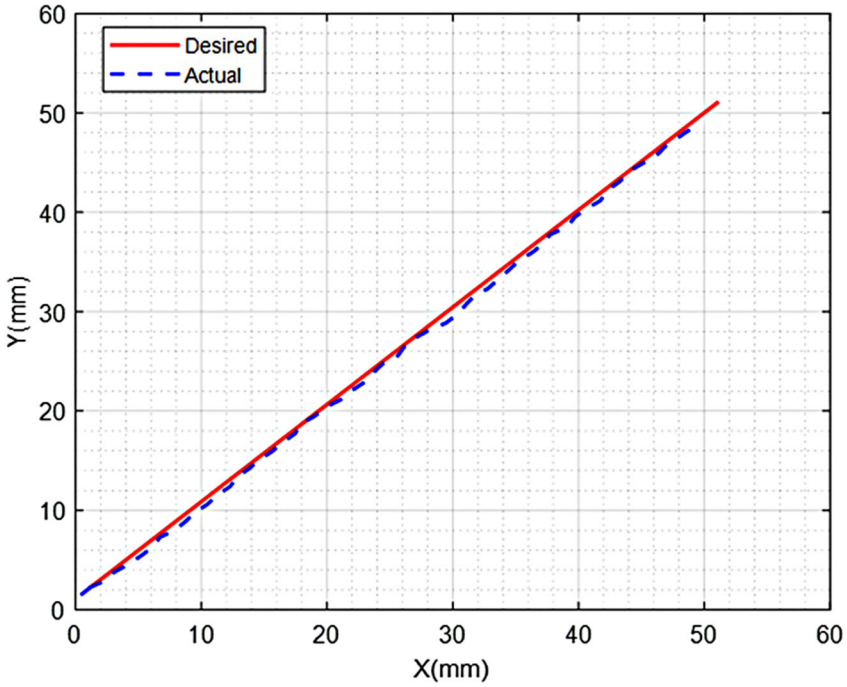


Figure 8. Cartesian straight-line motions under the SMRACS.

Table 3 shows that the absolute maximum and average errors of joint and Cartesian variables and the absolute synchronization errors generated in the case of the DASCS are significantly smaller than those of the SMRACS.

Table 3. Absolute maximum and average errors of straight-line motion.

	Maximum Error		Average Error	
	SMRACS	**DASCS**	**SMRACS**	**DASCS**
x_e (mm)	2.9872	2.0657	2.3780	1.1735
y_e (mm)	3.0586	2.5998	2.8056	1.1102
q_{e1} (mm)	1.6249	1.7359	0.4335	0.1786
q_{e2} (mm)	2.4713	2.5	1.1233	0.2231
ε (mm)	1.1559	0.8913	0.6897	0.1833

4.2. Experiment 2: Tracking a Circular Motion

In this experiment, the manipulator's end-effector was controlled by the DASCS and the SMRACS to carry out a circular path with a radius of 38.1 mm with respect to the Home Frame in 45 s. The Trajectory Generator in Figure 4 first formulated the desired Cartesian motion in terms of $x_d(t)$ and $y_d(t)$ and employed the manipulator inverse kinematic Equations (1) and (2) to determine the desired joint variables $q_{d1}(t)$ and q_{d2}(t).

Figures 9 and 10 illustrate time trajectories of the tracking errors $q_{e1}(t)$ (Actuator 1 in solid red line) and $q_{e2}(t)$ (Actuator 2 in dashed blue line) of the DASCS and SMRACS, respectively. As one can see, the time trajectories of the DASCS's tracking errors follow each other very closely, while those of the SMRACS do not. For this case, the DASCS has maximum and average synchronization errors of 1.1823 mm and 0.3736 mm, respectively, while the SMRACS has maximum and average synchronization errors of 2.8422 mm and 1.2515 mm, respectively. Studying the transient and steady-state behaviors of the control schemes, we found that the DASCS has better transient and steady-state responses than the SMRACS. In particular, the average tracking errors of the DASCS and the SMRACS for Actuator 1 are 0.4536 mm and 0.5228 mm, respectively, resulting in the SMRACS's average tracking errors being 115% of the DASCS's errors. The average tracking errors of the DASCS and the SMRACS for Actuator 2 are 0.4497 mm and 0.7461 mm, respectively, resulting in the SMRACS's average tracking errors being 165% of the DASCS's errors.

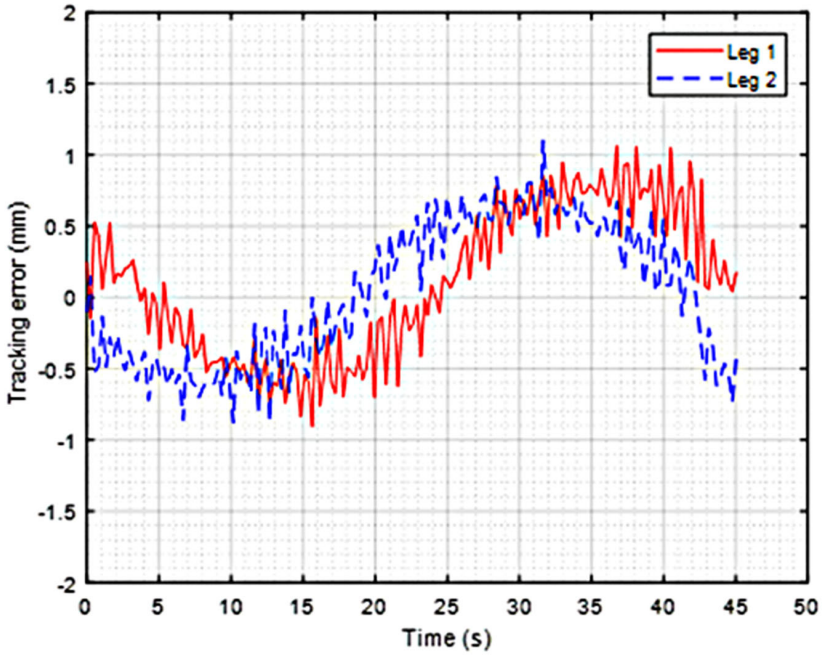


Figure 9. Time trajectories of tracking errors of circular motion under the DASCS.

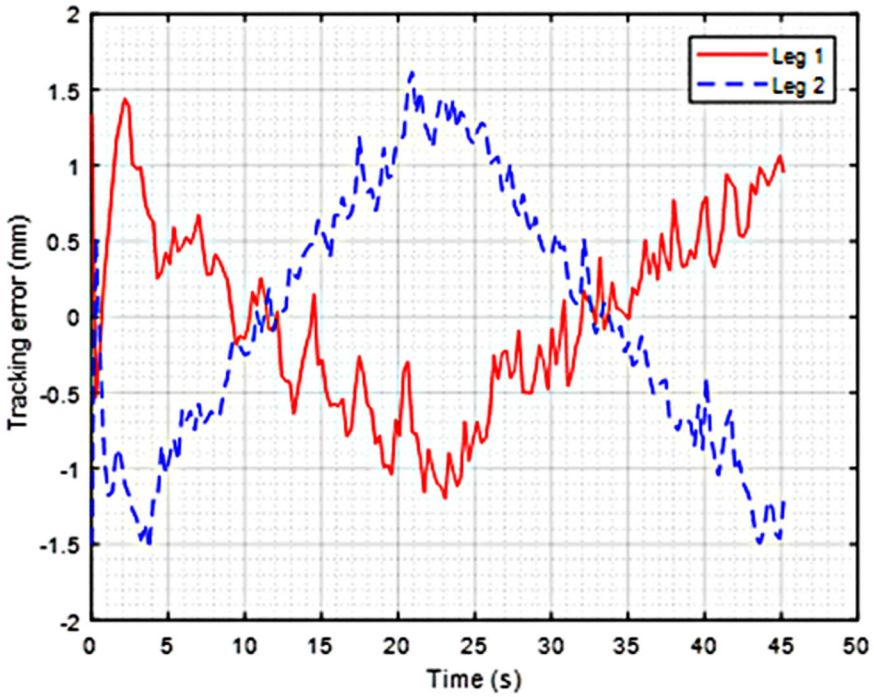


Figure 10. Time trajectories of tracking errors of circular motion under the SMRACS.

Additionally, the experimental data show that the DASCS has maximum tracking errors of 1.0612 mm and 1.1050 mm, while the maximum tracking errors of the SMRACS are 1.4396 mm and 1.6191 mm, respectively.

Figures 11 and 12 show the tracking performance of the two control schemes in the Home Frame. We see that the DASCS provides a better tracking performance than the SMRACS.

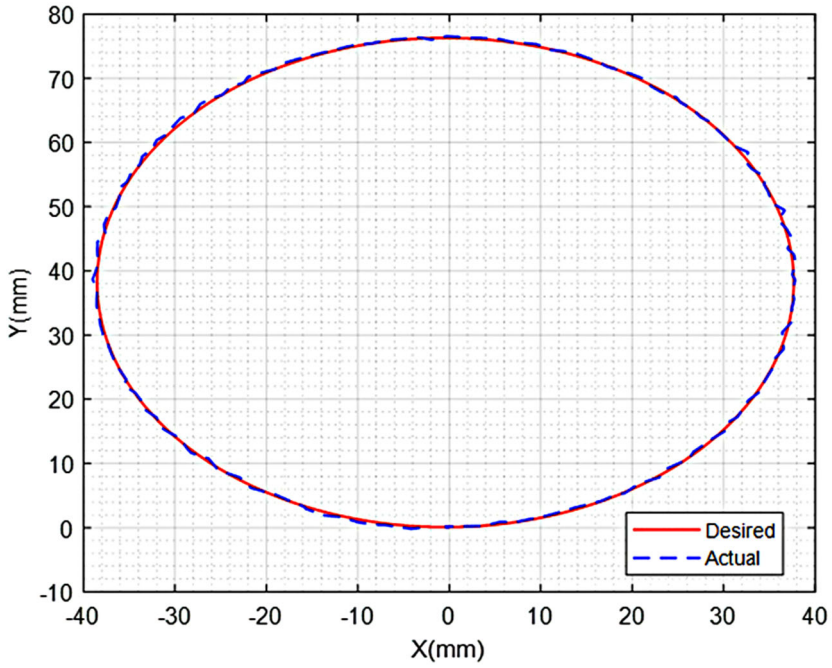


Figure 11. Cartesian circular motions under the DASCS.

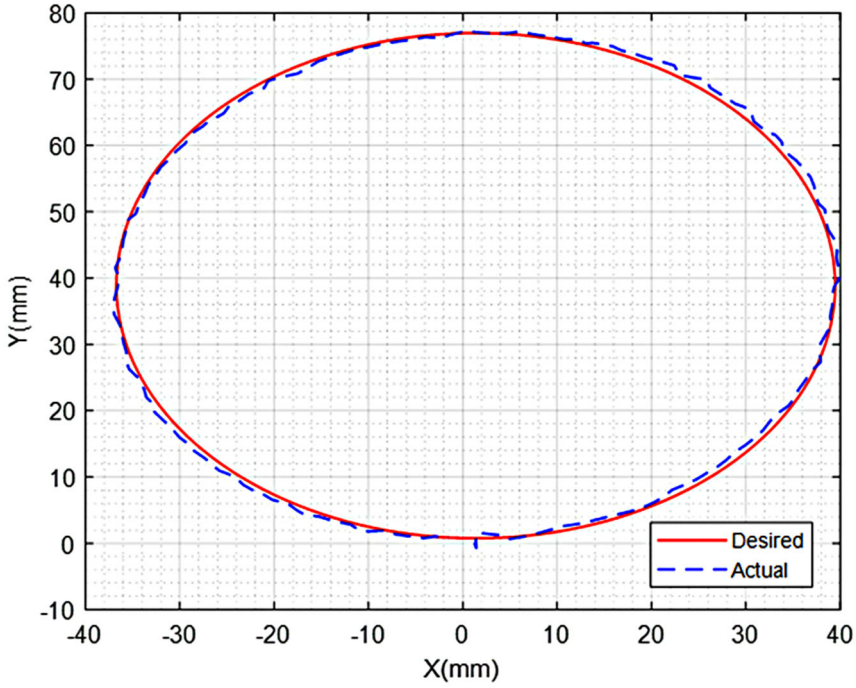


Figure 12. Cartesian circular motions under the SMRACS.

Table 4 shows that the absolute maximum and average position and synchronization errors achieved in the case of the DASCS are significantly smaller than those of the SMRACS.

Table 4. Absolute maximum and average errors of circular motion.

	Maximum Error		**Average Error**	
	SMRAC	**DASCS**	**SMRAC**	**DASCS**
x_e (mm)	3.8236	2.9166	1.9068	1.3485
y_e (mm)	2.5952	2.4054	1.3353	1.3148
q_{e1} (mm)	1.4396	1.0612	0.5228	0.4536
q_{e2} (mm)	1.6191	1.1050	0.7461	0.4497
ε (mm)	2.8422	1.1823	1.2515	0.3736

In light of the obtained results of the two performed experiments presented above, it is appropriate to analyze the results and provide resulting findings. We recall that the main difference between the DASCS and SMRACS is that the adaptation law of the DASCS as expressed in Equations (13) and (14) acts on the generalized errors $r_1(t)$ and $r_2(t)$ and the cross coupling errors ${e_1}^*(t)$ and ${e_2}^*(t)$, while the SMRACS acts on the conventional joint errors $q_{e1}(t)$ and $q_{e2}(t)$. Due to the inclusion of the cross coupling error control in the DASCS, we expect it to have better tracking performance than the SMRACS. Quantitatively, according to Figure 5 for the case of straight-line motion, the synchronized error control of the DASCS enabled it to "synchronize" the time trajectories of joint variable errors $q_{e1}(t)$ and $q_{e2}(t)$, and as a result, they followed each other very closely. On the other hand, without the synchronized error control in the case of the SMRACS, the time trajectories of the joint variable errors were farther apart, as seen in Figure 6, as compared to those of the DASCS. In addition, a qualitative analysis of the time trajectories of joint variable errors in the case of circular motion based on Figures 9 and 10 leads to the same conclusion. Particularly, Figure 10 shows that time trajectories of the joint variable errors under control of SMARCS were totally out of synchronization. Quantitatively, the results tabulated in

Tables 3 and 4 show that the absolute maximum and average errors in Cartesian space (x_e and y_e) and joint space (q_{e1} and q_{e2}) and the synchronization errors (ε_1 and ε_2) of the SMRACS were consistently and significantly higher than those of the DASCS when both control schemes were applied to track the straight-line and circular motions.

5. Conclusions

This paper focused on the experimental evaluation of the performance of a developed DASCS for motion control of CKCM manipulators. In our study, we considered 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. To comparatively evaluate the control performance of the developed DASCS, we conducted real-time experiments in which the DASCS and the SMRACS were applied to control a 2-DOF CKCM of a control scheme testbed to track two typical Cartesian motions, namely a straight line and a circular motion. Experimental results confirmed qualitatively and quantitatively that the DASCS had better tracking performance than the SMRACS. We concluded that the main reason causing the SMRACS to perform worse than the DASCS in the motion control of CKCM manipulators was its lack of synchronized error control. To continue the research in this paper, we recommend conducting experimental investigation of the performance of the DASCS on real CKCM manipulators with more DOFs in comparison with other conventional adaptive control schemes. Furthermore, the DASCS should be applied to OCKM robot manipulators, and its performance should be evaluated using computer simulation and experimentation. Additionally, experiments should be conducted to highlight the particular advantages of the DASCS, such as effective management of the interactions between active joints in closed-loop mechanisms, by including tests to simulate such challenges such as intentionally inducing disturbances in a selected joint and then analyzing the robustness of controllers under this circumstance. Finally, future research can be devoted to combining the DASCS with other intelligent control methods, including fuzzy control, neuro fuzzy control, and neural network control, to potentially improve its control performance.

Author Contributions: Conceptualization, T.T.N. and C.C.N.; methodology, C.C.N. and T.T.N.; software, T.T.N., C.C.N., and T.M.N.; validation, T.T.N., C.C.N. and T.T.C.D.; results analysis, T.T.N., C.C.N. and T.T.C.D.; investigation, T.T.N., C.C.N. and T.M.N.; 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, C.C.N., T.T.N., T.T.C.D. and L.S.; 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.

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

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

References

1. Tsai, L.W. *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*; John Wiley & Sons: Hoboken, NJ, USA, 1999.
2. Osamu, T.; Uchiyama, M.; Pierrot, F. Parallel Robot. U.S. Patent No. 5,333,514, 2 August 1994.
3. Fichter, E.F. A Stewart platform-based manipulator: General theory and practical construction. *Int. J. Robot. Res.* **1986**, *5*, 157–182. [CrossRef]
4. Lin, S. Dynamics of the manipulator with closed chains. *IEEE Trans. Robot. Autom.* **1990**, *6*, 496–501. [CrossRef]

5. Patel, Y.D.; George, P.M. Parallel manipulators applications—A survey. *Mod. Mech. Eng.* **2012**, *2*, 57. [CrossRef]
6. Zhang, D. *Parallel Robotic Machine Tools*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009.
7. Seraji, H. Decentralized adaptive control of manipulators: Theory, simulation, and experimentation. *IEEE Trans. Robot. Autom.* **1989**, *5*, 183–201. [CrossRef]
8. Luh, J.; Walker, M.; Paul, R. Resolved-acceleration control of mechanical manipulators. *IEEE Trans. Autom. Control* **1980**, *25*, 468–474. [CrossRef]
9. Dubowsky, S.; DesForges, D. The application of model-referenced adaptive control to robotic manipulators. *J. Dyn. Syst. Meas. Control.* **1979**, *101*, 193–200. [CrossRef]
10. Astrom, K. *Adaptive Control*, 2nd ed.; Addison-Wesley: Boston, MA, USA, 1995.
11. Zhang, D.; Wei, B.S. A review on model reference adaptive control of robotic manipulators. *Annu. Rev. Control* **2017**, *43*, 188–198. [CrossRef]
12. Nicosia, S.; Tomei, P. Model reference adaptive control algorithms for industrial robots. *Automatica* **1984**, *20*, 635–644. [CrossRef]
13. Nguyen, C.C.; Antrazi, S.S.; Zhou, Z.L.; Campbell, C.E. Adaptive control of a Stewart platform-based manipulator. *J. Robot. Syst.* **1993**, *10*, 657–687. [CrossRef]
14. Li, Y.; Wang, Y. Trajectory tracking control of a redundantly actuated parallel robot using diagonal recurrent neural network. In Proceedings of the International Conference on Natural Computation (ICNC'09), Tianjin, China, 14–16 August 2009; Volume 2, pp. 292–296.
15. Koren, Y. Cross-coupled biaxial computer control for manufacturing systems. *J. Dyn. Syst. Meas. Control* **1980**, *102*, 265–272.17. [CrossRef]
16. Kulkarni, P.K.; Srinivasan, K. Cross-coupled control of biaxial feed drive servo mechanisms. *ASME J. Dyn. Syst. Meas. Control.* **1990**, *112*, 225–232.
17. Tomizuka, M.; Hu, J.S.; Chiu, T.C. Synchronization of two motion control axes under adaptive feedforward control. *ASME J. Dyn. Syst. Meas. Control.* **1992**, *114*, 196–203. [CrossRef]
18. Koren, Y.; Lo, C.C. Advanced controllers for feed drives. *Ann. CIRP* **1992**, *41*, 689–698. [CrossRef]
19. 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.
20. 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. [CrossRef]
21. 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.
22. Nguyen, T.T.; Nguyen, C.C.; Nguyen, M.T.; Duong, T.C.T.; Ngo, T.T.H.; Lu, S. Decentralized adaptive control of closed-kinematic chain mechanism manipulator. *Machines* **2025**, *13*, 331. [CrossRef]
23. Duong, T.T.C.; Nguyen, C.C.; Tran, T.D. Synchronization Sliding Mode Control of Closed-Kinematic Chain Robot Manipulators with Time-Delay Estimation. *Appl. Sci.* **2022**, *12*, 5527. [CrossRef]
24. Nguyen, T.T. Intelligent Control of Closed-Kinematic Chain Robot Manipulators. Ph.D. Thesis, The Catholic University of America, Washington, DC, USA, 2020.

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.