

Article

Not peer-reviewed version

Hybrid Learning-based Control of Closed-Kinematic Chain Mechanism Robot Manipulators

[Charles C. Nguyen](#)*, [Tuan M. Nguyen](#), [Ha T.T. Ngo](#), Tri T. Nguyen, [Tu T. C. Duong](#)

Posted Date: 26 February 2026

doi: 10.20944/preprints202602.1612.v1

Keywords: learning control; iterative algorithm; closed-kinematic chain mechanism; parallel robots; Stewart platform; PID control; feedback control



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

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.

Article

Hybrid Learning-Based Control of Closed-Kinematic Chain Mechanism Robot Manipulators

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

¹ Department of Electrical and Computer Engineering, College of Engineering, Physics and Computing, The Catholic University of America, Washington, DC 20064, USA

² Department of Electrical Engineering, Danang University of Science and Technology, Danang City, 50000, Vietnam

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

* Correspondence: nguyen@cua.edu

Abstract

This paper presents a novel hybrid learning-based control scheme for position control of robot manipulators whose structure is based on a closed-kinematic-chain mechanism (CKCM). The developed control scheme integrates two complementary control components: The Feedback Controller and the Learning Controller. The Feedback Controller is designed using linearization about a desired trajectory and a PID control law whose gains are selected by a tuning algorithm to guarantee semi-global stability of the closed-loop system. The Learning Controller incorporates PID-type iterative learning strategy to generate additional control inputs to compensate for modeling uncertainties and unmodeled dynamics. By updating the control input iteratively from trial to trial, the Learning Controller progressively improves the overall control performance. The effectiveness of the developed control scheme is demonstrated through computer simulations conducted on a six-degree-of-freedom CKCM robot manipulator. Simulation results are presented and discussed to evaluate the tracking accuracy and robustness of the developed approach.

Keywords: learning control; iterative algorithm; closed-kinematic chain mechanism; parallel robots; Stewart platform; PID control; feedback control

1. Introduction

In many industrial applications, robot manipulators are required to execute identical tasks repeatedly. From an engineering standpoint, exploiting information from previous executions to improve tracking accuracy in subsequent trials is highly desirable. However, classical control approaches offer limited capability for achieving high-precision tracking in such repetitive operations. Consequently, significant research efforts have focused on developing control strategies that explicitly utilize data from prior trials, commonly referred to as iterative learning control (ILC) [1–3]

ILC enhances the performance of conventional feedback control by iteratively updating the control input based on historical tracking errors and control signals. Compared with many advanced control schemes, ILC possesses a relatively simple structure and is well suited for real-time implementation, as it requires low computational complexity and minimal a priori knowledge of system dynamics. Since the seminal work in [4], which first introduced ILC for robotic manipulator motion tracking, numerous ILC algorithms and applications have been extensively investigated [5–8].

Most existing studies share a common framework, with primary differences arising in the analysis of stability, convergence, and performance. Among various learning schemes, PD-type and D-type ILC algorithms are the most widely adopted in practical robotic applications, largely due to

the fact that many advanced ILC formulations remain highly theoretical and difficult to implement in real-world systems [9–12].

Closed-kinematic-chain mechanism (CKCM) manipulators, which feature parallel mechanical structures, have attracted considerable attention due to their ability to achieve high-precision positioning and their inherent avoidance of error accumulation associated with serial joint configurations [13–15]. CKCM manipulators have been applied in a wide range of fields, including space-based telerobotic assembly, aircraft simulators for pilot training, and the monitoring of human arm motion for stroke rehabilitation [16]. Despite extensive research on the mechanism design and dynamic modeling of CKCM manipulators, relatively limited effort has been devoted to addressing their motion control problems [17–23].

In this paper, a novel hybrid learning control strategy combining PID feedback control with an iterative learning control (ILC) scheme is developed for position control of CKCM robot manipulators. The learning controller adopts a PID-type iterative update law for a class of linear time-varying dynamic systems. Uniform convergence of the learning process is established using an exponentially weighted supremum-norm analysis and a contraction mapping framework. The effectiveness of the proposed hybrid learning control scheme is evaluated through computer simulations for the position control of a six-degree-of-freedom (DOF) CKCM manipulator.

This paper is organized as follows. Section 2 introduces the hybrid learning control scheme which integrates a Feedback Controller with a Learning Controller. Section 3 presents the design of the Feedback Controller, while Section 4 describes the development of the Learning Controller. An algorithm providing a design procedure for the learning control scheme is given in Section 5. Section 6 presents and discusses the results of computer simulation studies conducted to evaluate the effectiveness of the hybrid control scheme in tracking several paths. Finally, Section 7 concludes the paper with a summary of the main findings and outlines directions for future research.

2. The Hybrid Learning Control Scheme

Figure 1 presents the hybrid learning control scheme (HLCS) developed to control the motion of CKCM robot manipulators. As shown in the figure, the HLCS mainly consists of two controllers: a Feedback Controller and a Learning Controller. In general, the Feedback Controller ensures the overall stability of the system through appropriately selected control gains, while the Learning Controller aims to improve both the transient and steady-state responses of the manipulator end-effector position after each trial.

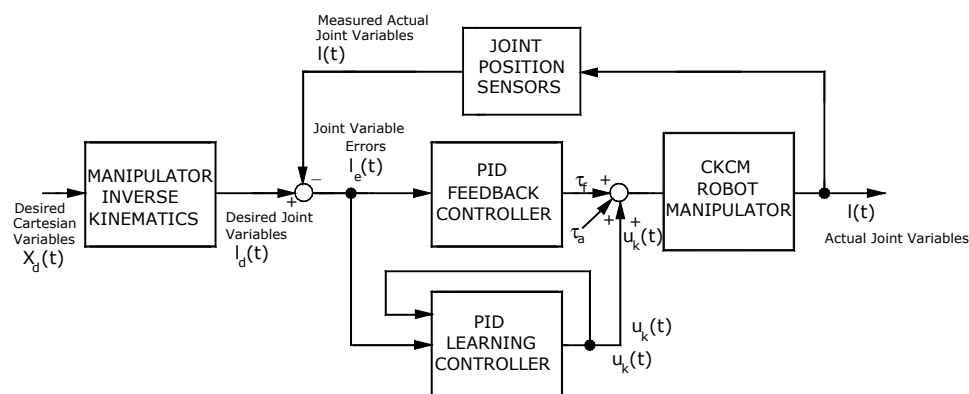


Figure 1. The hybrid learning control scheme for CKCM manipulators.

Robot kinematics and control deal with two types of variables, Cartesian variables and joint variables. Cartesian variables include positional variables x , y , and z , expressing the position of the

end-effector with respect to a reference frame, and three orientational variables, such as Euler angles, expressing the orientation of the end-effector with respect to the same reference frame. For CKCM robot manipulators possessing 6 DOFs, the joint variables are the lengths of the six actuators, l_1 , l_2 , l_3 , l_4 , l_5 , and l_6 .

The Feedback Controller takes advantage of the existence of a closed-form solution for inverse kinematics of CKCM robot manipulators and uses it to convert the desired Cartesian variables expressed in vector $\mathbf{x}_d(t)$ to the desired joint variables contained in vector $\mathbf{l}_d(t)$. The actual joint variables expressed by vector $\mathbf{l}(t)$ are measured by joint position sensors, and then the measured actual joint variables are subtracted from the desired joint variables contained in vector $\mathbf{l}_d(t)$ to produce the joint variable errors expressed by vector $\mathbf{l}_e(t)$. The joint variable errors are then sent to the Feedback Controller, which is a proportional–derivative–integral (PID) controller employing the PID control law to produce an input $\boldsymbol{\tau}_f(t)$ for the manipulator. The controller gains could be set at certain fixed values using standard control techniques, including linearization and pole placement or tuning guidelines to stabilize the closed-loop control system.

To improve the performance of CKCM robot manipulators in repetitive tasks, the Learning Controller utilizes an ILC algorithm based on errors from previous trials to generate a corrective input \mathbf{u}_k , which complements the input $\boldsymbol{\tau}_f(t)$ produced by the Feedback Controller. The total input to the CKCM robot manipulator consists of the feedback input $\boldsymbol{\tau}_f(t)$, the learning input \mathbf{u}_k , and an auxiliary input $\boldsymbol{\tau}_a(t)$, which compensates for the manipulator dynamics, as will be discussed later in the analysis of the Feedback Controller.

3. The Feedback Controller

This section will address the design of the Feedback Controller. The dynamical equations of a CKCM robot manipulator possessing 6 DOFs are given by

$$\mathbf{M}(\mathbf{l})\ddot{\mathbf{l}}(t) + \mathbf{C}(\mathbf{l}, \dot{\mathbf{l}})\dot{\mathbf{l}}(t) + \mathbf{G}(\mathbf{l})\mathbf{l}(t) = \boldsymbol{\tau}(t) \quad (1)$$

where $\mathbf{l}(t)$ denotes the (6x1) joint variable vector containing the lengths of the 6 actuators of the manipulator, $\mathbf{M}(\mathbf{l})$, $\mathbf{C}(\mathbf{l}, \dot{\mathbf{l}})$ and $\mathbf{G}(\mathbf{l})$ represent the (6x6) manipulator mass matrix, the (6x1) centrifugal and Coriolis force vector and the (6x1) gravitational force vector, respectively. In addition, $\boldsymbol{\tau}(t)$ denotes the (6x1) control input vector to the manipulator.

Linearizing (1) about the desired joint variable vector \mathbf{l}_d by using Taylor Series expansion and neglecting higher order terms, we obtain

$$\mathbf{M}(t)\ddot{\mathbf{z}}(t) + \mathbf{C}(t)\dot{\mathbf{z}}(t) + \mathbf{G}(t)\mathbf{z}(t) + \boldsymbol{\tau}_a(t) = \boldsymbol{\tau}(t) \quad (2)$$

where

$$\mathbf{z}(t) = \mathbf{l}(t) - \mathbf{l}_d(t) \quad (3)$$

$$\mathbf{M}(t) = \mathbf{M}(\mathbf{l}_d)_{\mathbf{l}_d, \mathbf{l}_d} \quad (4)$$

$$\mathbf{C}(t) = \frac{\partial}{\partial \dot{\mathbf{q}}} [\mathbf{C}(\mathbf{l}, \dot{\mathbf{l}})]_{\mathbf{l}_d, \dot{\mathbf{l}}_d} \quad (5)$$

$$\mathbf{G}(t) = \frac{\partial}{\partial \mathbf{q}} [\mathbf{M}(\mathbf{l})\ddot{\mathbf{l}}_d + \mathbf{C}(\mathbf{l}, \dot{\mathbf{l}})\dot{\mathbf{l}} + \mathbf{G}(\mathbf{l})]_{\mathbf{l}_d, \mathbf{l}_d} \quad (6)$$

and

$$\boldsymbol{\tau}_a(t) = \mathbf{M}(\mathbf{l}_d)\ddot{\mathbf{l}}_d + \mathbf{C}(\mathbf{l}_d, \dot{\mathbf{l}}_d)\dot{\mathbf{l}}_d + \mathbf{G}(\mathbf{l}_d) \quad (7)$$

From Figure 1, we get

$$\boldsymbol{\tau}(t) = \boldsymbol{\tau}_f(t) + \mathbf{u}_k(t) + \boldsymbol{\tau}_a(t) \quad (8)$$

where $\boldsymbol{\tau}_f(t)$, the output of the Feedback Controller employing the PID control law is given by

$$\boldsymbol{\tau}_f(t) = \mathbf{K}_p \mathbf{l}_e(t) + \mathbf{K}_D \dot{\mathbf{l}}_e(t) + \mathbf{K}_I \int \mathbf{l}_e(t) dt \quad (9)$$

with

$$\mathbf{l}_e(t) = \mathbf{l}_d(t) - \mathbf{l}(t) \quad (10)$$

and \mathbf{K}_p , \mathbf{K}_i and \mathbf{K}_d are the controller gain matrices of the PID control law, respectively.

Now substituting (8)-(9) into (2) yields

$$\mathbf{M}(t)\ddot{\mathbf{z}}(t) + [\mathbf{C}(t) + \mathbf{K}_d]\dot{\mathbf{z}}(t) + [\mathbf{G}(t) + \mathbf{K}_p]\mathbf{z}(t) + \mathbf{K}_i \int \mathbf{z}(t) dt = \mathbf{u}_k(t) \quad (11)$$

where we let

$$\boldsymbol{\tau}_a(t) = \boldsymbol{\tau}_d(t) \quad (12)$$

We also note that

$$\mathbf{l}_e(t) = -\mathbf{z}(t) \quad (13)$$

The linearized dynamics model (34) can be semi-globally stabilized by properly selecting the gain matrices \mathbf{K}_p , \mathbf{K}_i and \mathbf{K}_d of the PID control law by employing tuning guidelines given in [24].

4. The Learning Controller

Figure 2 illustrates the structure of the Learning Controller consisting of a PID-type learning control law and a large-scale integrated random-access memory (LSI RAM). The learning process is described by

$$\mathbf{u}_{k+1}(t) = \mathbf{u}_k(t) + \{\boldsymbol{\Gamma} \frac{d}{dt} + \boldsymbol{\Phi} + \boldsymbol{\Psi} \int dt\} \mathbf{l}_{ek}(t) \quad (14)$$

where $\mathbf{u}_k(t)$ denotes the output of the learning controller during the k^{th} trial, $\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$ and $\boldsymbol{\Gamma}$ are the gain matrices of the PID learning control law, respectively. $\mathbf{l}_{ek}(t)$ is the joint error as given in (10) during the k^{th} trial. During the k^{th} trial, the information of \mathbf{u}_{k+1} is computed using (14) and stored in the lower part of the RAM as a set of densely sampled digital data. After the k^{th} trial, the stored data will be loaded to the upper part of the memory and will be sent to the actuators of the CKCM robot manipulator during the $(k + 1)^{th}$ trial. The lower part of the memory is now empty and ready to store new data.

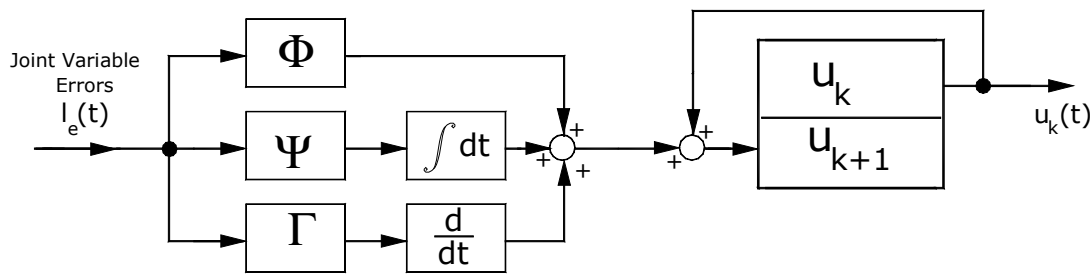


Figure 2. The PID-Learning Controller.

In general, a learning controller deals with a class of linear time-varying systems described by

$$\begin{cases} \mathbf{R}(t)\ddot{\mathbf{x}}(t) + \mathbf{Q}(t)\dot{\mathbf{x}}(t) + \mathbf{P}(t)\mathbf{x}(t) = \mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{x}(t) \end{cases} \quad (15)$$

where $\mathbf{y}(t), \mathbf{x}(t)$ and $\mathbf{u}(t)$ all $\in R^n$ denote the output vector, the state vector and input vector, respectively. The system matrices $\mathbf{R}(t), \mathbf{Q}(t)$, and $\mathbf{P}(t) \in R^{n \times n}$ are assumed to be continuously differentiable and satisfy

$$\begin{cases} \mathbf{R}(t) = \mathbf{R}^T(t) \geq \mathbf{R}_0 > 0 \\ \mathbf{Q}(t) \geq \mathbf{Q}_0 \\ \mathbf{P}(t) \geq \mathbf{P}_0 \end{cases} \quad (16)$$

where $\mathbf{R}_0, \mathbf{Q}_0$, and \mathbf{P}_0 are the lower limit of $\mathbf{R}(t), \mathbf{Q}(t)$, and $\mathbf{P}(t)$, respectively. Given a desired output $\mathbf{y}_d(t) \in R^n$ that is continuously differentiable on $[0, T]$, and an input $\mathbf{u}_k(t) \in R^n$ that is continuous on $[0, T]$, we consider a PID-type ILC law described by

$$\begin{cases} \mathbf{R}(t)\ddot{\mathbf{x}}_k(t) + \mathbf{Q}(t)\dot{\mathbf{x}}_k(t) + \mathbf{P}(t)\mathbf{x}_k(t) = \mathbf{u}_k(t) \\ \mathbf{y}_k(t) = \mathbf{x}_k(t) \\ \mathbf{e}_k(t) = \mathbf{y}_d(t) - \mathbf{y}_k(t) \\ \mathbf{u}_{k+1}(t) = \mathbf{u}_k(t) + \left\{ \Gamma \frac{d}{dt} + \Phi + \Psi \int dt \right\} \mathbf{e}_k \end{cases} \quad (17)$$

In applying the PID learning algorithm defined in (17), three standard assumptions commonly adopted in ILC are considered:

- The repeatability of the initial conditions is ensured; that is, the initial state $\mathbf{x}_k(0)$ of the system can be reset to the same value at the beginning of each operation as follows:

$$\mathbf{x}_k(0) = \mathbf{x}^0 \text{ for } k = 1, 2, \dots \quad (18)$$

- Each operation ends at a fixed finite time $T > 0$.
- The system dynamics represented by $\mathbf{R}(t), \mathbf{Q}(t)$ and $\mathbf{P}(t)$ is assumed to remain invariant throughout the repeated training process.

Now an operator Δ for $\mathbf{x}(t)$ is defined as

$$\Delta(\mathbf{x}(t)) = \Gamma \frac{d\mathbf{x}(t)}{dt} + \Phi \mathbf{x}(t) + \Psi \int_0^t \mathbf{x}(\tau) d\tau \quad (19)$$

and we recall that a ρ -vector norm of the state vector $\mathbf{x}(t)$ with element $x_i, i = 1, 2, 3, \dots, n$ is defined by

$$|\mathbf{x}|_\rho = (\sum_{i=1}^n |x_i|^\rho)^{1/\rho} \quad (20)$$

which is a positive scalar.

In order to relate the elements of the state vector $\mathbf{x}(t)$ to the gain matrices Γ, Φ , and Ψ of the ILC law, we force the ρ -vector norm of $\mathbf{x}(t)$ with an initial condition $\mathbf{x}(0) = 0$ to be as follows

$$\begin{aligned} |\mathbf{x}(t)|_\rho &= \int_0^t e^{-\rho\tau} \{\Delta \mathbf{x}(\tau)\}^T \Phi^{-1} \{\Delta \mathbf{x}(\tau)\} d\tau \\ &= \int_0^t e^{-\rho\tau} \{\Phi \mathbf{x} + \Gamma \dot{\mathbf{x}} + \Phi \mathbf{y}\}^T \Phi^{-1} \{\Phi \mathbf{x} + \Gamma \dot{\mathbf{x}} + \Phi \mathbf{y}\} d\tau \\ &= \int_0^t e^{-\rho\tau} \{\mathbf{x}^T \Phi \mathbf{x} + \mathbf{x}^T \Gamma \Phi^{-1} \Gamma \dot{\mathbf{x}} + \mathbf{y}^T \Psi \Phi^{-1} \Psi \mathbf{y}\} d\tau + \int_0^t e^{-\rho\tau} \{\mathbf{x}^T \Gamma \dot{\mathbf{x}} + \mathbf{x}^T \Psi \mathbf{y} + \mathbf{x}^T \Gamma \Phi^{-1} \Psi \mathbf{y}\} d\tau \end{aligned} \quad (21)$$

where the mathematical operator \mathbf{X} denotes $\frac{d}{dt}\mathbf{x}$, and mathematical operator \mathbf{Y} denotes $\int_0^t \mathbf{x}$. By applying integration by parts to (21), this equation can be rewritten as

$$|\mathbf{x}(t)|_\rho = \int_0^t e^{-\rho\tau} \{ \mathbf{x}^T (\mathbf{\Phi} + \rho\mathbf{\Gamma}) \mathbf{x} + \mathbf{X}^T \mathbf{\Gamma} \mathbf{\Phi}^{-1} \mathbf{\Gamma} \mathbf{x} + \mathbf{Y}^T (\mathbf{\Psi} \mathbf{\Phi}^{-1} \mathbf{\Psi} + \rho\mathbf{\Psi}) \mathbf{Y} \} d\tau + \int_0^t e^{-\rho\tau} \{ 2\mathbf{X}^T \mathbf{\Gamma} \mathbf{\Phi}^{-1} \mathbf{\Psi} \mathbf{Y} \} d\tau + e^{-\rho t} \{ \mathbf{Y}^T \mathbf{\Psi} \mathbf{Y} + \mathbf{x}^T \mathbf{\Gamma} \mathbf{x} \} \quad (22)$$

We also have from (17)

$$\mathbf{R}(t)\ddot{\mathbf{d}}_k + \mathbf{Q}(t)\dot{\mathbf{d}}_k + \mathbf{P}(t)\mathbf{d}_k = \mathbf{\Delta}(e_k(t)) \quad (23)$$

where we define

$$\dot{\mathbf{d}}_k = \mathbf{x}_{k+1}(t) - \mathbf{x}_k(t) \rightarrow \dot{\mathbf{d}}_k = \mathbf{e}_k(t) - \mathbf{e}_{k+1}(t) \quad (24)$$

Now, from (24) let us consider

$$|\mathbf{e}_{k+1}(t)|_\rho = |\mathbf{e}_k(t) - \dot{\mathbf{d}}_k(t)|_\rho \quad (25)$$

$$= |\mathbf{e}_k(t)|_\rho + |\dot{\mathbf{d}}_k(t)|_\rho - 2 \int_0^t e^{-\rho\tau} \{ \Delta \dot{\mathbf{d}}_k(\tau) \}^T \mathbf{\Phi}^{-1} \{ \Delta \mathbf{e}_k(\tau) \} d\tau \quad (26)$$

$$= |\mathbf{e}_k(t)|_\rho + |\dot{\mathbf{d}}_k(t)|_\rho - 2 \int_0^t e^{-\rho\tau} \{ \ddot{\mathbf{d}}_k(\tau)^T \mathbf{\Gamma} + \dot{\mathbf{d}}_k(\tau)^T \mathbf{\Phi} + \mathbf{d}_k^T \mathbf{\Psi} \} \mathbf{\Phi}^{-1} \{ \mathbf{R} \ddot{\mathbf{d}}_k + \mathbf{Q} \dot{\mathbf{d}}_k + \mathbf{P} \mathbf{d}_k \} d\tau \quad (27)$$

Substituting $|\dot{\mathbf{d}}_k(t)|_\rho$ from (26) into (27) and simplifying the integral term yield

$$|\mathbf{e}_{k+1}(t)|_\rho = |\mathbf{e}_k(t)|_\rho - \int_0^t e^{-\rho\tau} \{ \dot{\mathbf{d}}_k^T \mathbf{S}_2 \ddot{\mathbf{d}}_k + \dot{\mathbf{d}}_k^T \mathbf{S}_1 \dot{\mathbf{d}}_k + \mathbf{d}_k^T \mathbf{S}_0 \mathbf{d}_k \} d\tau - e^{-\rho t} \{ \dot{\mathbf{d}}_k^T \mathbf{K}_2 \ddot{\mathbf{d}}_k + \dot{\mathbf{d}}_k^T \mathbf{K}_1 \dot{\mathbf{d}}_k + \mathbf{d}_k^T \mathbf{K}_0 \mathbf{d}_k \} + e^{-\rho t} \{ \mathbf{d}_k^T \mathbf{L}_0 \ddot{\mathbf{d}}_k + \dot{\mathbf{d}}_k^T \mathbf{V}_0 \dot{\mathbf{d}}_k + \mathbf{d}_k^T \mathbf{U}_0 \mathbf{d}_k \} + \int_0^t e^{-\rho\tau} \{ \mathbf{d}_k^T \mathbf{L}_1 \ddot{\mathbf{d}}_k + \dot{\mathbf{d}}_k^T \mathbf{V}_1 \dot{\mathbf{d}}_k + \mathbf{d}_k^T \mathbf{U}_1 \mathbf{d}_k \} d\tau \quad (28)$$

where

$$\mathbf{S}_2 = \rho \mathbf{\Gamma} \mathbf{\Phi}^{-1} \mathbf{R} - \mathbf{\Gamma} \mathbf{\Phi}^{-1} \mathbf{R} - \mathbf{\Gamma} \mathbf{\Phi}^{-1} \dot{\mathbf{R}} - \mathbf{\Gamma} \mathbf{\Phi}^{-1} \mathbf{\Gamma} - 2\mathbf{R}$$

$$\mathbf{S}_1 = \rho(\mathbf{Q} + \mathbf{P}^T \mathbf{\Phi}^{-1} \mathbf{\Gamma} - \mathbf{\Gamma}) - \mathbf{\Phi} - \dot{\mathbf{Q}} - \dot{\mathbf{P}}^T \mathbf{\Phi}^{-1} \mathbf{\Gamma} - 2\mathbf{\Psi} \mathbf{\Phi}^{-1} \mathbf{Q} + 2\mathbf{\Gamma} \mathbf{\Phi}^{-1} \mathbf{\Psi}$$

$$\mathbf{S}_0 = \rho(\mathbf{\Psi} \mathbf{\Phi}^{-1} \mathbf{P} - \mathbf{\Psi}) - \mathbf{\Psi} \mathbf{\Phi}^{-1} \mathbf{\Psi} - \mathbf{\Psi} \mathbf{\Phi}^{-1} \dot{\mathbf{P}}$$

$$\mathbf{K}_2 = \mathbf{\Gamma} \mathbf{\Phi}^{-1} \mathbf{R}$$

$$\mathbf{K}_1 = \mathbf{Q} + \mathbf{P}^T \mathbf{\Phi}^{-1} \mathbf{\Gamma} - \mathbf{\Gamma}$$

$$\begin{aligned}
K_0 &= \Psi\Phi^{-1}P - \Psi \\
L_1 &= \rho(-2\Psi\Phi^{-1}R) + 2\Psi\Phi^{-1}\dot{R} \\
V_1 &= \rho(-2R) + 2\dot{R} + 2\Psi\Phi^{-1}R \\
U_1 &= \rho(-2\Psi\Phi^{-1}Q + 2\Psi\Phi^{-1}\Gamma) + 2\Psi\Phi^{-1}\dot{Q} \\
&\quad - 2\Psi\Phi^{-1}\dot{\Gamma} \\
L_0 &= -2\Psi\Phi^{-1}R \\
V_0 &= -2R \\
U_0 &= -2\Psi\Phi^{-1}Q + 2\Psi\Phi^{-1}\Gamma
\end{aligned}$$

We have now successfully established a relationship between the performance errors $e_k(t)$ and $e_{k+1}(t)$ of the k^{th} and k^{th} trials, respectively, in terms of the ILC gain matrices and the system matrices of the linear time-varying system, as shown on the right-hand side of (28). Equation (28) can be regarded as a discrete-time difference equation that relates $e_k(t)$ to $e_{k+1}(t)$ when the system is controlled by the PID-ILCS described in (17). Furthermore, the system described by (28) may be regarded as a discrete closed-loop feedback system in the k -domain—rather than the time domain—in which the PID-ILCS functions as a discrete feedback controller.

Based on (28), the central theorem of the proposed PID learning law can now be stated as follows: **Theorem 1:** Assume that each component of $\mathbf{u}_0(t)$ is continuous and each component of $\dot{\mathbf{y}}_d(t)$ is continuously differentiable, namely

$$\mathbf{u}_0(t) \in \mathbb{C}[0, T] \text{ and } \mathbf{y}_d(t) \in \mathbb{C}^2[0, T] \quad (29)$$

and the gain matrices Γ , Φ , and Ψ are symmetric positive definite and satisfy

$$\begin{cases}
\left(\lambda_2 - \frac{1}{2}\lambda_1 - \frac{1}{2}\right) \mathbf{R} > 0 \\
\left(1 + \frac{1}{2}\lambda_1\right) \mathbf{Q} + \lambda_2 \mathbf{P}^T - 2\mathbf{R} - \left(\frac{1}{2}\lambda_1 + 1\right) \Gamma > 0 \\
\lambda_1(-2\mathbf{Q} - 2\mathbf{R} + \mathbf{P} + 2\Gamma) - \Psi > 0
\end{cases} \quad (30)$$

where $\|A\|$ denotes the spectral norm of matrix A and the scalars λ_1 and λ_2 are defined as

$$\begin{cases}
\lambda_1 = \|\Phi^{-1}\Psi\| \\
\lambda_2 = \|\Phi^{-1}\Gamma\|
\end{cases} \quad (31)$$

then the PID-type learning control described by (17) with the same initial position and velocity, is convergent in the sense that $\dot{\mathbf{x}}_k(t) \rightarrow \dot{\mathbf{x}}_d(t)$ pointwise in $t \in [0, T]$ and $\mathbf{x}_k(t) \rightarrow \mathbf{x}_d(t)$ uniformly in $t \in [0, T]$ as $k \rightarrow \infty$.

Proof: Applying Schwarz's inequality to the last six terms of (28), it can be seen that

$$\begin{aligned}
|e_{k+1}(t)|_\rho \leq |e_k(t)|_\rho - \int_0^t e^{-\rho\tau} \{ \ddot{\mathbf{d}}_k^T \mathbf{A}_2 \ddot{\mathbf{d}}_k + \dot{\mathbf{d}}_k^T \mathbf{A}_1 \dot{\mathbf{d}}_k + \mathbf{d}_k^T \mathbf{A}_0 \mathbf{d}_k \} d\tau - e^{-\rho t} \{ \ddot{\mathbf{d}}_k^T \mathbf{B}_2 \ddot{\mathbf{d}}_k + \\
\dot{\mathbf{d}}_k^T \mathbf{B}_1 \dot{\mathbf{d}}_k + \mathbf{d}_k^T \mathbf{B}_0 \mathbf{d}_k \} \quad (32)
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{A}_2 &= \rho\Gamma\Phi^{-1}R - \gamma\rho(\Psi\Phi^{-1}R + R) + Z \\
\mathbf{A}_1 &= \rho(Q + P^T\Phi^{-1}\Gamma - \Gamma) \\
&\quad - \rho(\gamma^{-1}R \\
&\quad + \gamma(\Psi\Phi^{-1}Q - \Psi\Phi^{-1}\Gamma)) + Z
\end{aligned}$$

$$\begin{aligned}
\mathbf{A}_0 &= \rho(\Psi\Phi^{-1}\mathbf{P} - \Psi) \\
&\quad - \rho\gamma^{-1}(\Psi\Phi^{-1}\mathbf{R} + \Psi\Phi^{-1}\mathbf{Q}) \\
&\quad - \Psi\Phi^{-1}\Gamma) + \mathbf{Z} \\
\mathbf{B}_2 &= \Gamma\Phi^{-1}\mathbf{R} - \gamma(\Psi\Phi^{-1}\mathbf{R} + \mathbf{R}) \\
\mathbf{B}_1 &= (\mathbf{Q} + \mathbf{P}^T\Phi^{-1}\Gamma - \Gamma) \\
&\quad - (\gamma^{-1}\mathbf{R} \\
&\quad + \gamma(\Psi\Phi^{-1}\mathbf{Q} - \Psi\Phi^{-1}\Gamma)) \\
\mathbf{B}_0 &= \Psi\Phi^{-1}\mathbf{P} - \Psi - \gamma^{-1}(\Psi\Phi^{-1}\mathbf{R} + \Psi\Phi^{-1}\mathbf{Q}) \\
&\quad - \Psi\Phi^{-1}\Gamma)
\end{aligned}$$

where $0 < \gamma < 1$ is a constant and \mathbf{Z} contains the remaining term of matrices that are independent of ρ . If ρ is chosen sufficiently large and $\gamma = 1/2$, we can make all terms of the right-hand side of (32) become positive definite using the conditions given in (30). With that, it can be concluded that

$$|\mathbf{e}_{k+1}(t)|_\rho \leq |\mathbf{e}_k(t)|_\rho \quad (33)$$

which in light of (26) also implies that

$$\ddot{\mathbf{d}}_k \rightarrow 0, \dot{\mathbf{d}}_k \rightarrow 0 \text{ as } k \rightarrow \infty \quad (34)$$

Note that according to the condition (29), $\mathbf{x}_0(t) \in \mathbb{C}^2[0, T]$, which implies $\Delta(\mathbf{e}_0(t)) \in \mathbb{C}^1[0, T]$ and also $\mathbf{d}_0(t) \in \mathbb{C}^4[0, T]$. Hence, with the help of a similar argument in the derivation of (23), we obtain

$$\frac{d}{dt}(\mathbf{R}(t)\ddot{\mathbf{d}}_k + \mathbf{Q}(t)\dot{\mathbf{d}}_k + \mathbf{P}(t)\mathbf{d}_k) = \Delta(\dot{\mathbf{e}}_k(t)) \quad (35)$$

Similarly, we have

$$\begin{aligned}
|\dot{\mathbf{e}}_{k+1}(t)|_\rho \leq |\dot{\mathbf{e}}_k(t)|_\rho - \int_0^t e^{-\rho\tau} \{ \ddot{\mathbf{d}}_k^T \mathbf{A}_2 \ddot{\mathbf{d}}_k + \ddot{\mathbf{d}}_k^T \mathbf{A}_1 \ddot{\mathbf{d}}_k + \dot{\mathbf{d}}_k^T \mathbf{A}_0 \dot{\mathbf{d}}_k + \mathbf{d}_k^T \mathbf{Z} \mathbf{d}_k \} d\tau - \\
e^{-\rho t} \{ \ddot{\mathbf{d}}_k^T \mathbf{B}_2 \ddot{\mathbf{d}}_k + \ddot{\mathbf{d}}_k^T \mathbf{B}_1 \ddot{\mathbf{d}}_k + \dot{\mathbf{d}}_k^T \mathbf{B}_0 \dot{\mathbf{d}}_k \} (36)
\end{aligned}$$

Applying the same argument as in the derivation of (34), it can be concluded that

$$\ddot{\mathbf{d}}_k \rightarrow 0, \dot{\mathbf{d}}_k \rightarrow 0, \mathbf{d}_k \rightarrow 0 \text{ as } k \rightarrow \infty \quad (37)$$

Substituting this result into (23) yields

$$\Delta(\mathbf{e}_k(t)) \rightarrow 0 \text{ as } k \rightarrow \infty \quad (38)$$

From the expression for $\Delta(\mathbf{e}_k(t))$ and from the definition as in (31), we obtain the integral inequality

$$\|\Phi \mathbf{e}_k\| \leq \|\Delta \mathbf{e}_k\| + \frac{d}{dt} \lambda_2 \|\Phi \mathbf{e}_k\| + \int_0^t \lambda_1 \|\Phi \mathbf{e}_k\| d\tau \quad (39)$$

With the assumption that $\|\Phi \mathbf{e}_k\| \leq \|\Phi \mathbf{e}_k(0)\| e^{\lambda_2 t}$, we apply the Bellman-Gronwall Inequalities (Appendix A) to obtain

$$\|\Phi \mathbf{e}_k\|(1 - \lambda_2) \leq \|\Delta \mathbf{e}_k\| + \lambda_1 \int_0^t \|\Delta \mathbf{e}_k\| e^{\lambda_1 \tau} d\tau \quad (40)$$

Then, for any fixed $t \in [0, T]$, equation (26) implies that

$$\mathbf{e}_k(t) \rightarrow 0 \text{ as } k \rightarrow \infty \quad (41)$$

Finally, the Ascoli-Arzela's Theorem (Appendix B) can be invoked in the following way. Observing that $\dot{\mathbf{x}}_d(t) - \dot{\mathbf{x}}_k(t)$ is uniformly bounded, which implies the uniform boundedness and equicontinuity of $\mathbf{x}_d(t) - \mathbf{x}_k(t)$. In addition, since $\mathbf{x}_d(t) - \mathbf{x}_k(t)$ is pointwise convergent to zero, it follows from Ascoli-Arzela's Theorem that $\mathbf{x}_k(t)$ converges to $\mathbf{x}_d(t)$ uniformly in $t \in [0, T]$. Thus, this completes the proof of Theorem 1.

5. Design Procedure for the Hybrid Learning Control Scheme

This section combines developments of the Feedback Controller in Section 3 and the Learning Controller in Section 4 to outline a design procedure for the HLCS.

First, for the Feedback Controller, we recall that a general 6 DOF CKCM robot manipulator with the dynamics given in (1) can be represented by a linearized model about a desired joint variable vector \mathbf{l}_d given in (2). After applying a PID feedback control law, the closed-loop system can be represented by (11), which can be semi-globally stabilized by employing tuning guidelines given in [24].

For the Learning Controller, comparing (15) and (2), we recognize that $\mathbf{R}(t)$, $\mathbf{Q}(t)$ and $\mathbf{P}(t)$ in (15) could represent $\mathbf{M}(t)$, $\mathbf{C}(t)$ and $\mathbf{G}(t)$, respectively, and $\mathbf{x}(t)$ in (15) represents $\mathbf{l}(t)$. If Theorem 1 is employed to determine the gain matrices $\mathbf{\Gamma}$, $\mathbf{\Phi}$, and $\mathbf{\Psi}$ of the ILC law so that $\mathbf{x}_k(t) \rightarrow \mathbf{x}_d(t)$, then $\mathbf{l}_k(t) \rightarrow \mathbf{l}_d(t)$.

Based on the above discussion, we obtain a design procedure for the HLCS as follows:

Algorithm 1. Design Procedure for HLCS

Step 1. Linearize the dynamics of the CKCM robot manipulator (1) about a desired joint variable vector $\mathbf{l}_d(t)$ to obtain $\mathbf{M}(t)$, $\mathbf{C}(t)$ and $\mathbf{G}(t)$ of the linearized model.

Step 2. Use the tuning procedure in [24] to obtain \mathbf{K}_p , \mathbf{K}_i and \mathbf{K}_d of the PID control law of the Feedback Controller to globally stabilize the closed-loop feedback system (11).

Step 3. Use Theorem 1 to obtain the gain matrices $\mathbf{\Gamma}$, $\mathbf{\Phi}$, and $\mathbf{\Psi}$ of the ILC law so that $\mathbf{x}_k(t) \rightarrow \mathbf{x}_d(t)$ resulting in $\mathbf{l}_k(t) \rightarrow \mathbf{l}_d(t)$.

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

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

6.1. Inverse Kinematics and Dynamics of the 6-DOF CKCM Manipulator

Figure 3 shows the CKCM manipulator that possesses six DOFs and comprises a stationary base platform and a moving platform coupled together through six 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. A robotic end-effector can be attached to the moving platform to carry out tasks. The displacements of the six linear actuators result in the motion of the moving platform and its end-effector.



Figure 3. The 6 DOF CKCM manipulator.

The manipulator inverse kinematics is defined as the determination of the lengths of the six actuators to produce a desired position and orientation of the moving platform. Appendix C containing a complete kinematic analysis of the above manipulator provides a closed-form solution of the inverse kinematics from (A.21) as

$$l_i^2 = x^2 + y^2 + z^2 + r_A^2 + r_B^2 + 2(r_{11}a_{ix} + r_{12}a_{iy})(x - b_{ix}) + 2(r_{21}a_{ix} + r_{22}a_{iy})(y - b_{iy}) + 2(r_{31}a_{ix} + r_{32}a_{iy})z - 2(xb_{ix} + yb_{iy}) \quad (42)$$

where l_i for $i = 1, 2, \dots, 6$ are the lengths of the six actuators, and r_A and r_B denote the radii of the moving and base platforms, respectively. Furthermore, r_{ij} are the elements of the orientation matrix ${}^B_A\mathbf{R}$ and x, y, z denote the desired position of the moving platform with respect to the base platform. We note that Equation (42) indeed presents the closed-form solution of the inverse kinematics, in the sense that the actuator lengths l_i for $i = 1, 2, \dots, 6$ can be determined from the desired Cartesian configuration of the moving platform, specified by $x, y, z, \alpha, \beta, \gamma$.

Appendix D contains a dynamical analysis of the above CKCM manipulator. It employs the Lagrangian formulation to derive the dynamics of the manipulator in terms of the kinematic energy and the potential energy of the manipulator. In particular, Equation (A.31) represents the equations of motion of the manipulator expressing the relationship between the Cartesian force/torque F_j applied to the moving platform and the Cartesian position and orientation of the moving platform ϕ_j , for $j = 1, 2, \dots, 6$. We also see that (A.31) contains the three terms, the Inertial Term, the Centrifugal and Coriolis Term and the Gravity Term.

6.2. Computer Simulation Results

Table 1 presents the parameters of the 6-DOF CKCM manipulator employed in the computer simulation study. The simulations are conducted using SimMechanics™ (Version 2019), a toolbox of MATLAB/Simulink (Version 2019). This toolbox enables efficient modeling and visualization of real mechanical systems composed of multiple rigid bodies. Moreover, by leveraging the Simulink environment, SimMechanics™ facilitates the simulation of the physical behaviors and motions of the modeled systems.

The SimMechanics library represents physical components as interconnected blocks, explicitly capturing the geometric and kinematic relationships among the robot components. Consequently, this modeling approach does not require sophisticated mathematical formulations of the system dynamics, while providing a convenient and accurate means of deriving the dynamic model of the system. Figure 4 illustrates the SimMechanics™ model of the manipulator. The integration of the plant with the HLCS is illustrated in Figure 5. The orange blocks represent memory units that store

commands from previous iterations, the gray blocks correspond to the feedback and learning controllers, and the cyan block denotes the robot manipulator (plant).

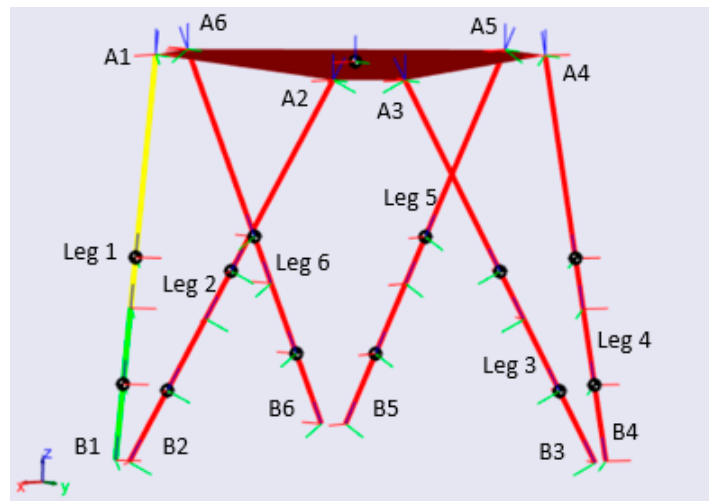


Figure 4. Simulink model of the 6 DOF CKCM manipulator.

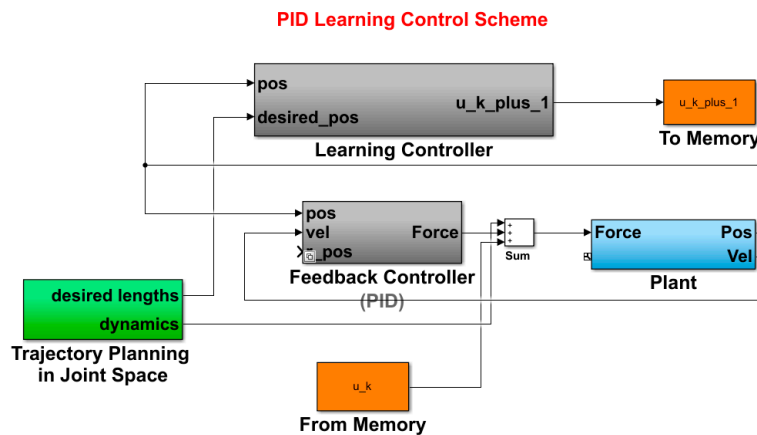


Figure 5. Simulink block diagram of HLCS for the 6 DOF CKCM manipulator.

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

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} ($\text{kg}\cdot\text{m}^2$)	0.09
Inertia coefficient of the platform, I_{yy} ($\text{kg}\cdot\text{m}^2$)	0.09
Inertia coefficient of the platform, I_{zz} ($\text{kg}\cdot\text{m}^2$)	0.18
PID Feedback Controller	Control Gain
K_p	50000
K_d	850
K_i	10

PID-type ILC	Control gains
Φ	47728
Γ	847
Ψ	7.5

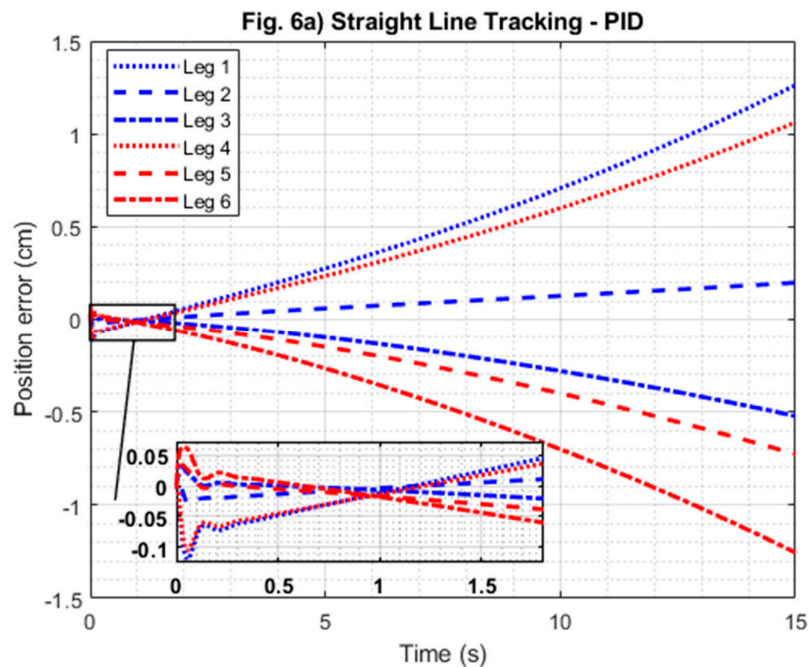
In the simulation program, the sampling period is fixed at 1 ms, and the numerical solver employed is the ODE8 (Dormand–Prince) method. The plant parameters and controller gains are listed in Table 1.

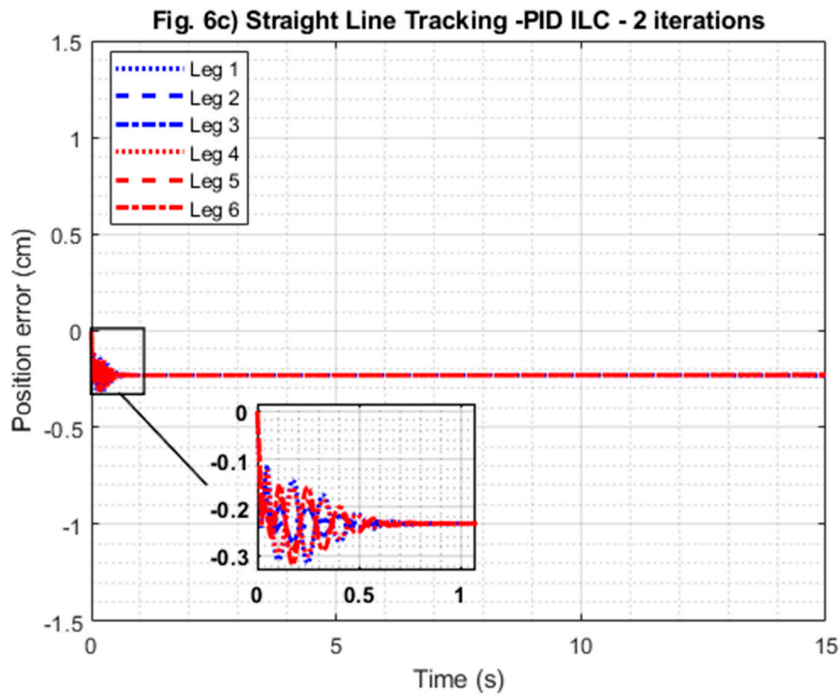
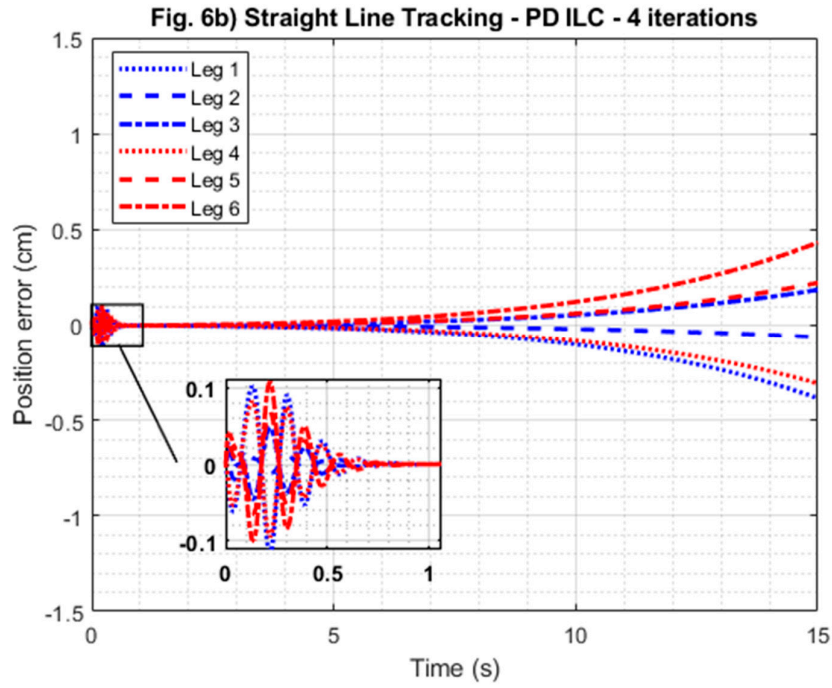
To evaluate the tracking performance of the HLCS, its simulation model is obtained and employed to track the manipulator end-effector along two different planar trajectories. The obtained results are then compared with those achieved using a PID feedback controller alone and with those achieved by a hybrid control scheme combining PID feedback with PD-type ILC.

Case 1: Tracking a Straight Line Motion

The straight-line trajectory to be tracked by the end-effector is defined by $x + y - 2z + 1 = 0$ (in meters) over a duration of 15 s. The simulation results for this case are presented in Figure 6(a)–6(d). Figure 6(a)–6(c) show the time responses of the leg tracking errors obtained using the PID controller, the PD-type ILC, and the developed PID-HLCS, respectively. Figure 6(d) compares the desired and actual motions achieved by the three control schemes.

As observed from the results, the HLCS yields substantial improvements in both transient and steady-state performance after only two learning trials, whereas the PD-type ILC requires four trials to reach a comparable level of performance. In Figure 6(c), the steady-state tracking error of the HLCS converges to approximately -0.22 cm within 0.6 s, while the responses of the other two controllers fail to converge after the transient phase.





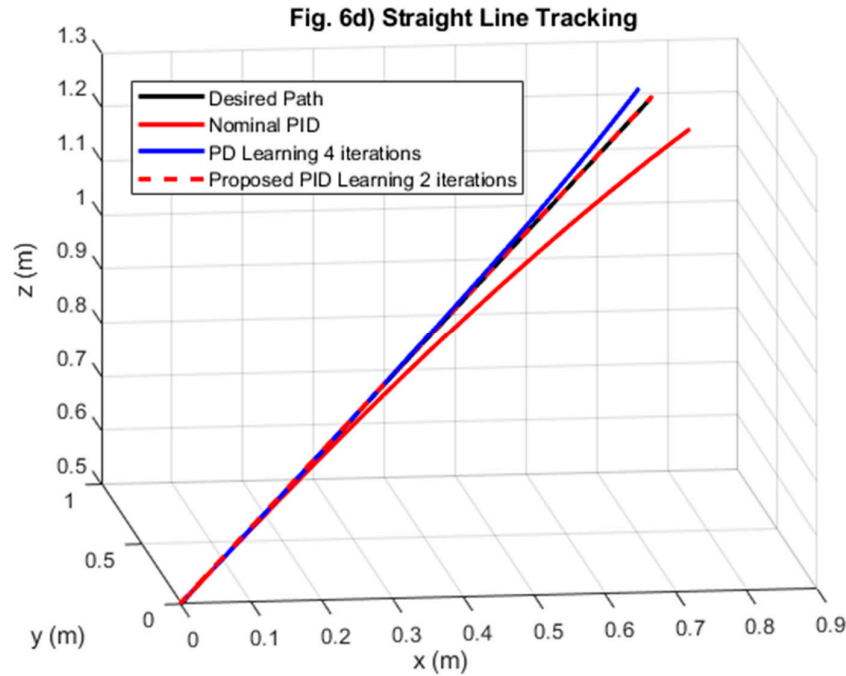
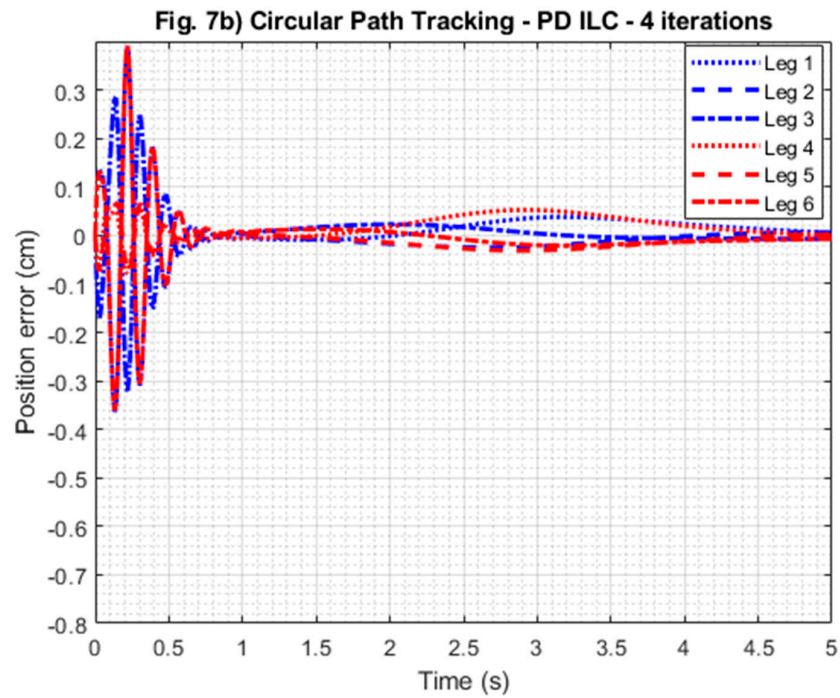
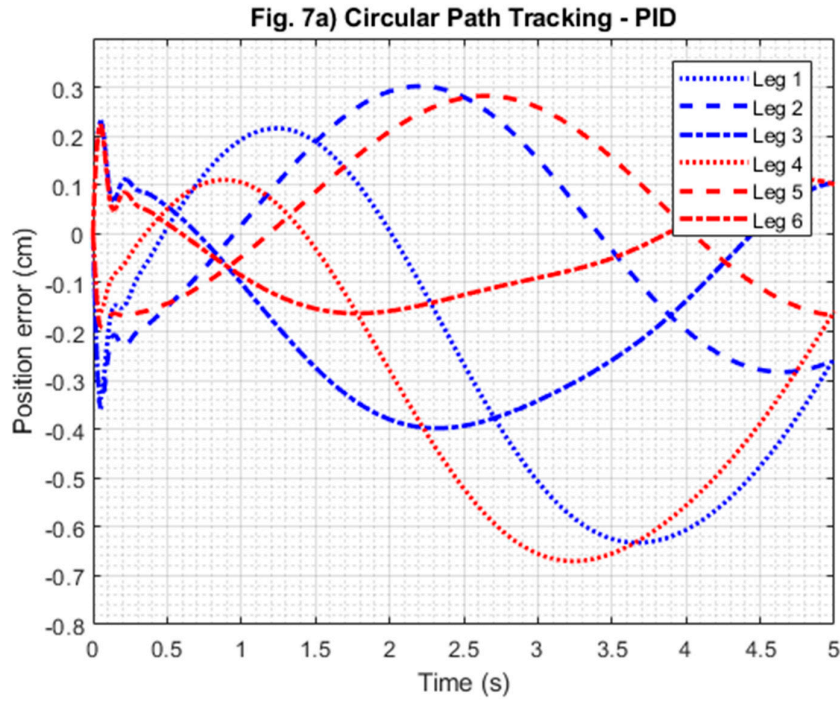


Figure 6. Simulation results of tracking a straight line.

Case 2: Tracking a Circular Motion

The circular trajectory to be tracked by the manipulator end-effector is defined by $x^2 + (y - 0.2)^2 + (z + 0.5)^2 = 0.2^2$ (in meters) over a duration of 5 s. Similarly, Figure 7(a)–7(c) illustrate the time responses of the leg tracking errors obtained using the PID controller, the PD-type ILC, and the proposed PID-type ILC, respectively. Figure 7(d) compares the desired and actual end-effector motions achieved by the three control schemes.

The circular path to be followed by the manipulator end-effector is described by $x^2 + (y - 0.2)^2 + (z + 0.5)^2 = 0.2^2$ [in m] in 5s. Similarly, Figure 7a-c represent the time responses of the leg errors from the PID, PD ILC, and the developed HLCS, respectively. Figure 7d compares the actual and desired motions of all three controllers.



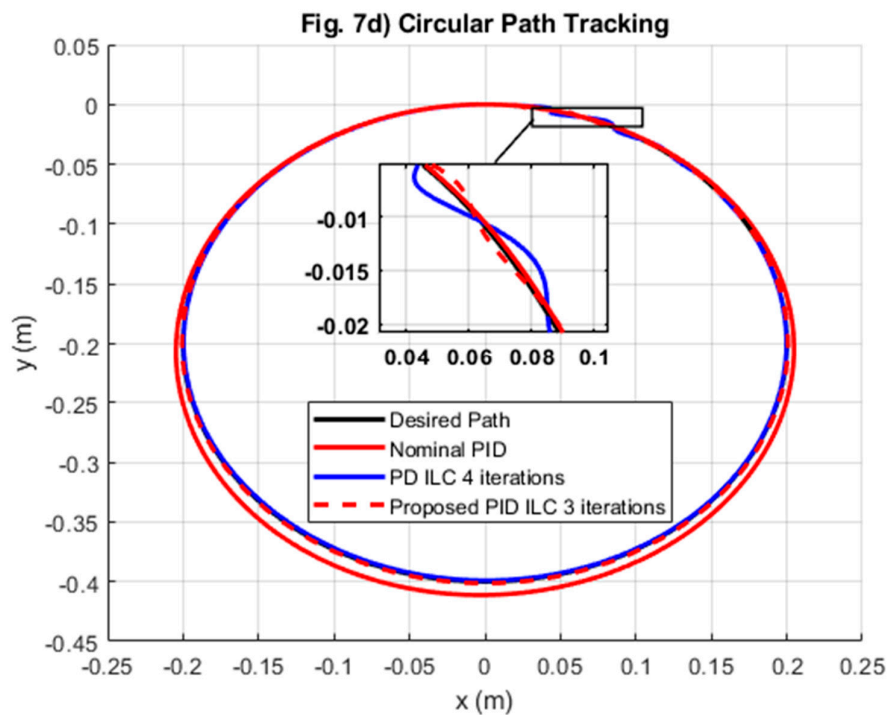
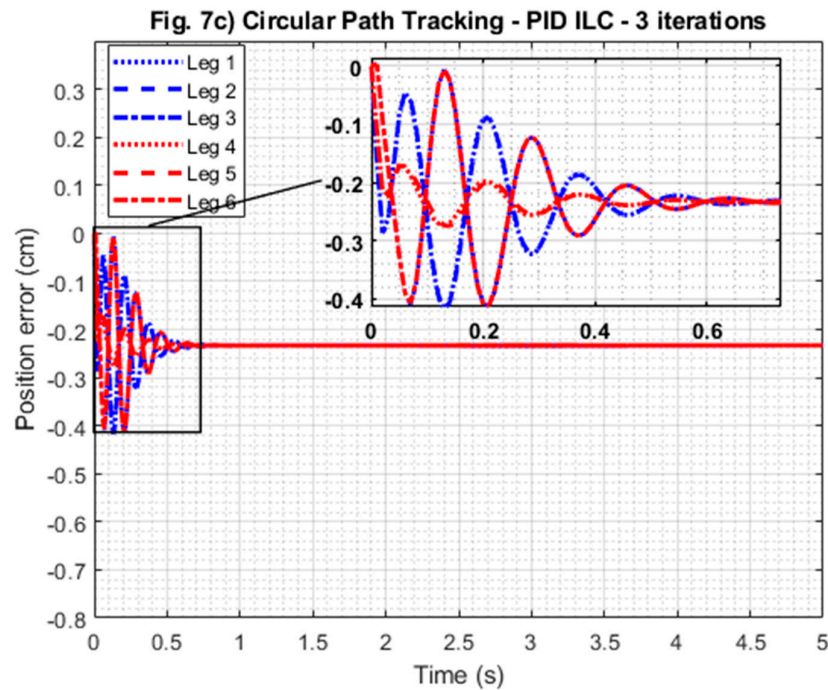


Figure 7. Simulation results of tracking a circular path.

Figure 7(b) illustrates the improvement in tracking error achieved by the PD-type ILC; however, its transient response exhibits the largest degradation, as evidenced in the zoomed-in view of Figure 7(d). In contrast, Figure 7(c) demonstrates that the HLCS achieves satisfactory tracking performance after only three learning iterations.

Figure 8 depicts the error convergence across iterations, represented by the q -norm of the tracking error as a function of the iteration number. The results clearly indicate that the developed

HLCS attains a faster convergence rate than the conventional PD-type ILC for both straight-line and circular trajectory tracking tasks.

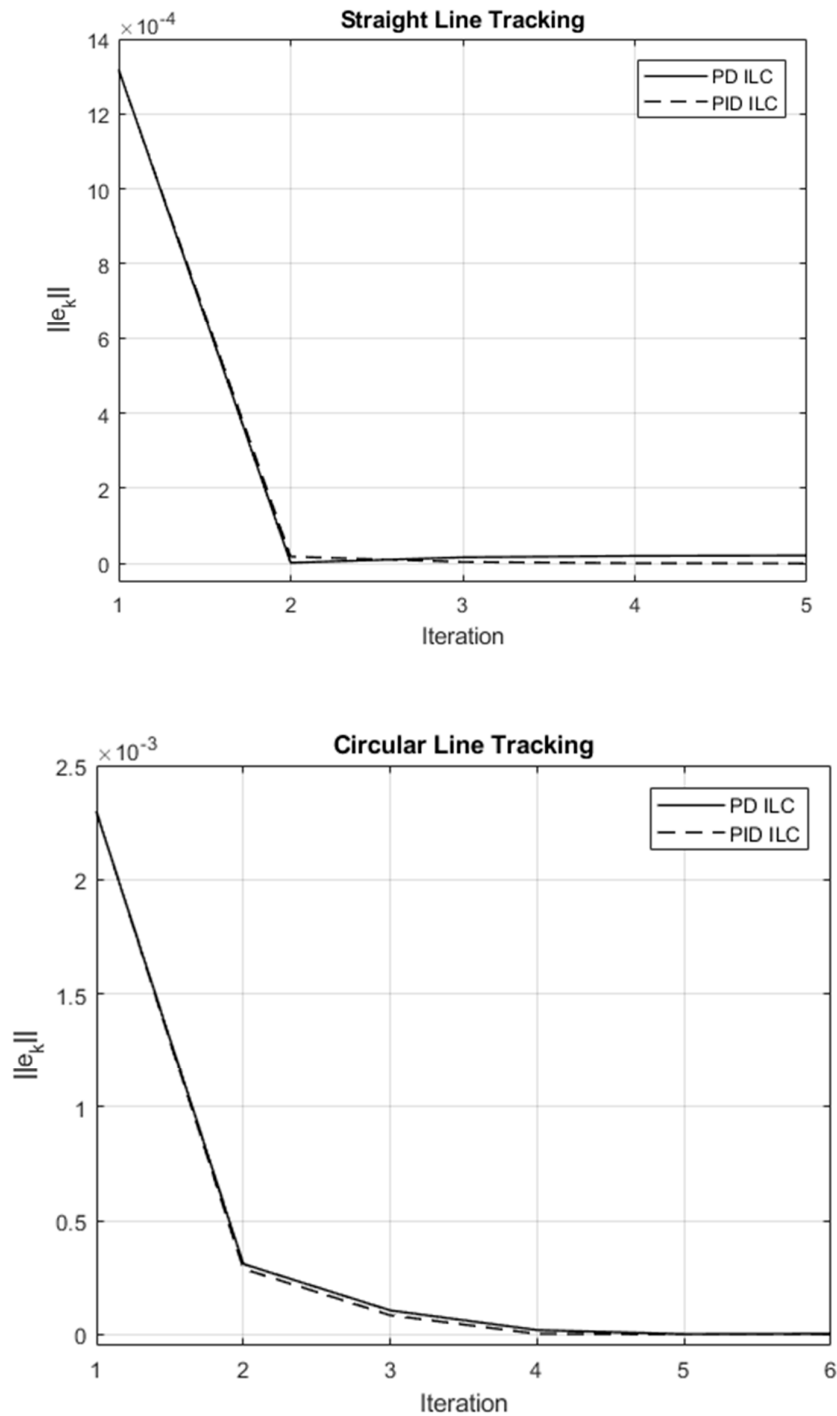


Figure 8. ρ -norm of the tracking error with respect to the number of iterations.

7. Conclusions

This paper presents the development of the HLCS, a novel hybrid learning control strategy that integrates feedback control with learning control for the motion control of robot manipulators. The hybrid control scheme is designed for robot manipulators based on the CKCM architecture, which

offers several attractive features, including the existence of inverse kinematics solutions, high-precision positioning capability, and inherent avoidance of joint error accumulation commonly associated with manipulators designed based on open kinematic chain mechanisms.

The HLCS mainly consists of two components: a Feedback Controller and a Learning Controller, each serving a distinct control function. The Feedback Controller employs a PID control law whose gains are selected using a linearization method to globally stabilize the closed-loop system. The Learning Controller stores joint error information and the corresponding control inputs during a trial and utilizes this information to improve both transient and steady-state responses of the manipulator motion in subsequent trials. Uniform convergence of the learning process is established through exponentially weighted supremum-norm analysis and a contraction mapping framework. Based on the development of the Feedback and Learning Controllers, a systematic design procedure for the HLCS is proposed.

Computer simulations are conducted to evaluate the control performance of the HLCS in comparison with a conventional PID control scheme without learning capability and a PD control scheme combined with iterative learning control (ILC). All control schemes are implemented to track the same straight and circular paths. Simulation results demonstrate that the HLCS outperforms the other two control schemes in terms of convergence rate and tracking accuracy.

Future work on this novel learning control scheme is recommended as follows:

- Experimental evaluation of the developed HLCS on real CKCM manipulators and comparison with other conventional learning control schemes. [25–29]
- Application of the HLCS to serial robot manipulators, with performance evaluation through both simulation and experimental validation.
- Extension of the hybrid concept proposed in this paper to integrate a feedback controller with intelligent control techniques, such as fuzzy control, neuro-fuzzy control, or neural networks, to enhance real-time control performance of robot manipulators.
- Investigation of the robustness and resilience of the developed HLCS.

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

Funding: This research received no external funding.

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

Acknowledgments: The authors have reviewed and edited the output and take full responsibility for the content of this publication.

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

Abbreviations

The following abbreviations were used in this manuscript:

CKCM	Closed-Kinematic Chain Mechanism
PID	Proportional-Integral-Derivative
HLCS	Hybrid Learning Control Scheme
ILC	Iterative Learning Control

Appendix A. Bellman-Gronwall Theorem

Bellman-Gronwall Theorem in Differential Form. Let I be an interval of real values $[a,b]$ with $a < b$. Let β and u be real-values continuous functions defined on I . If u is differentiable in the interior I^0 (the interval I without the end points a or b) and satisfies the differential inequality as follows

$$\dot{u}(t) \leq \beta(t)u(t), t \in I^0 \quad (\text{A.1})$$

Then u is bounded by the solution of the corresponding differential equation $\dot{v}(t) = \beta(t)v(t)$

$$u(t) \leq u(a)\exp\left(\int_a^t \beta(s)ds\right), t \in I \quad (\text{A.2})$$

Bellman-Gronwall Theorem in Integral Form. Let I be an interval of real values $[a,b]$ with $a < b$. Let x, β and u be real-values continuous functions defined on I . If β is non-negative and if u satisfies the differential inequality as follows

$$u(t) \leq x(t) + \int_a^t \beta(s)u(s)ds, t \in I \quad (\text{A.3})$$

then

$$u(t) \leq x(t) + \int_a^t x(s)\beta(s)\exp\left(\int_s^t \beta(r)dr\right)ds, t \in I \quad (\text{A.4})$$

If, in addition, the function x is non-decreasing, then

$$u(t) \leq x(t)\exp\left(\int_a^t \beta(s)ds\right), t \in I \quad (\text{A.5})$$

Appendix B. Arzela-Ascoli Theorem

The Arzala-Ascoli theorem is a fundamental result of mathematical analysis giving necessary and sufficient conditions to decide whether every sequence of a given family of real-valued continuous functions defined on a closed and bounded interval has a uniformly convergent subsequence whose main condition is the equicontinuity of the family of functions. By definition, a sequence $\{f_n(x)\}_{n \in \mathbb{N}}$ of continuous functions on an interval $I \in [a, b]$ is uniformly bounded if there is a number M such that $|f_n(x)| \leq M$ for all functions f_n in the sequence and $x \in [a, b]$. The sequence is said to be equicontinuous if, for all $\varepsilon > 0$ and x , there exists a $\delta > 0$ such that $|f_n(x) - f_n(y)| \leq \varepsilon$ whenever $|x - y| < \delta$ for all functions f_n in the sequence. Thus, the theorem can be state as follows

Arzela-Ascoli Theorem: Consider a sequence of real-valued continuous functions $\{f_n(x)\}_{n \in \mathbb{N}}$ defined on a closed and bounded interval $[a, b]$ of the real line. If this sequence is uniformly bounded and equicontinuous, then there exists a subsequence $\{f_{n_k}(x)\}_{k \in \mathbb{N}}$. The converse is also true, which is if all subsequence of $\{f_n(x)\}_{n \in \mathbb{N}}$ itself has a uniformly convergent subsequence, then $\{f_n(x)\}_{n \in \mathbb{N}}$ is uniformly bounded and equicontinuous.

Appendix C. Kinematic Analysis of the 6 DOF CKCM Manipulator

This appendix presents a detailed kinematic analysis of the 6 DOF CKCM robot manipulator and obtains a closed-form solution of the manipulator inverse kinematics. Figure A1 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 degrees.

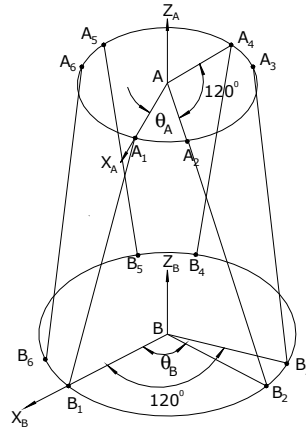


Figure A1. Frame assignment for the CKCM manipulator.

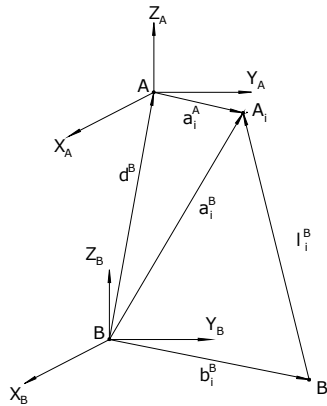


Figure A2. Vector diagram for the i^{th} actuator.

Referring to Figure A2, 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 A3, 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:

$$\mathbf{a}_i^A = [r_A \cos(\lambda_i) \ r_A \sin(\lambda_i) \ 0]^T \quad (\text{A.6})$$

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 $\mathbf{b}_i^A = (b_{ix} \ b_{iy} \ b_{iz})^T$ is given by the following:

$$\mathbf{b}_i^B = [r_B \cos(\Lambda_i) \ r_B \sin(\Lambda_i) \ 0]^T \quad (\text{A.7})$$

The angles λ_i and Λ_i are determined by the following:

$$\Lambda_i = 60(i - 1) \text{ deg}; \ \lambda_i = 60(i - 1) \text{ deg}, \quad \text{for } i = 1, 3, 5 \quad (\text{A.8})$$

$$\Lambda_i = \Lambda_{i-1} + \theta_B \text{ deg}; \lambda_i = \lambda_{i-1} + \theta_A \text{ deg}, \text{ for } i = 2,4,6 \quad (\text{A.9})$$

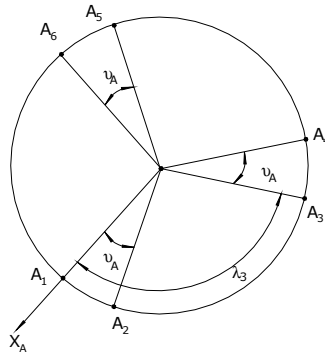


Figure A3. Position of the ball joints A_i with respect to the X_A -axis.

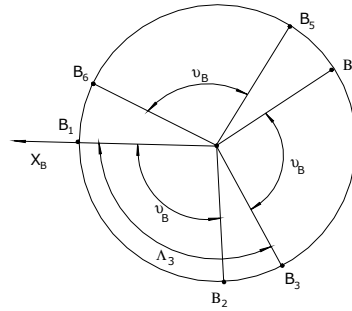


Figure A4. Position of the ball joints B_i with respect to the X_B -axis.

The kinematic equations can be derived from the vector diagram for each actuator in Figure A4. The length vector $\mathbf{l}_i^B = (l_{ix} \ l_{iy} \ l_{iz})^T$, expressing the vector $B_i A_i$ with respect to frame {B}, can be computed as follows:

$$\mathbf{l}_i^B = \mathbf{a}_i^B - \mathbf{d}_B \quad (\text{A.10})$$

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:

$$\mathbf{d}_B = (x \ y \ z)^T \quad (\text{A.11})$$

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:

$${}^B_A \mathbf{R} = \mathbf{R}_{zyx, \alpha \beta \gamma} = \begin{bmatrix} C\alpha C\beta & C\alpha S\beta S\gamma - S\alpha C\gamma & C\alpha S\beta C\gamma + S\alpha S\gamma \\ S\alpha C\beta & S\alpha S\beta S\gamma + C\alpha C\gamma & S\alpha S\beta C\gamma - C\alpha S\gamma \\ -S\beta & C\beta S\gamma & C\beta C\gamma \end{bmatrix} \quad (\text{A.12})$$

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

$$\mathbf{a}_i^B = {}^B_A \mathbf{R} \mathbf{a}_i^A + \mathbf{d}_B \quad (\text{A.13})$$

Substituting (C.8) into (C.5) yields the following:

$$\mathbf{l}_i^B = {}^B_A \mathbf{R} \mathbf{a}_i^A + \mathbf{d}_B - \mathbf{b}_i^B \quad (\text{A.14})$$

Using (A9), vector \mathbf{l}_i^B can be solved for a desired Cartesian configuration of the moving platform, specified by $x, y, z, \alpha, \beta, \gamma$ as follows:

$$\mathbf{l}_i^B = \begin{bmatrix} l_{ix} \\ l_{iy} \\ l_{iz} \end{bmatrix} = \begin{bmatrix} r_{11}a_{ix} + r_{12}a_{iy} + r_{13}a_{iz} + x - b_{ix} \\ r_{21}a_{ix} + r_{22}a_{iy} + r_{23}a_{iz} + y - b_{iy} \\ r_{31}a_{ix} + r_{32}a_{iy} + r_{33}a_{iz} + z - b_{iz} \end{bmatrix} \quad (\text{A.15})$$

where r_{ij} are the elements of the orientation matrix ${}^B_A \mathbf{R}$ given by (C.7). After solving for vector \mathbf{l}_i^B , the square of the required corresponding length of the i th actuator, q_i can be computed by the following:

$$l_i^2 = (l_{ix}^2 + l_{iy}^2 + l_{iz}^2) \quad \text{for } i = 1, 2, \dots, 6 \quad (\text{A.16})$$

We proceed to substitute (A.15) into (A.16) to obtain the following:

$$\begin{aligned} l_i^2 = & x^2 + y^2 + z^2 + a_{ix}^2 + a_{iy}^2 + a_{iz}^2 + b_{ix}^2 + b_{iy}^2 + b_{iz}^2 \\ & + 2(r_{11}a_{ix} + r_{12}a_{iy} + r_{13}a_{iz})(x - b_{ix}) \\ & + 2(r_{21}a_{ix} + r_{22}a_{iy} + r_{23}a_{iz})(y - b_{iy}) \\ & + 2(r_{31}a_{ix} + r_{32}a_{iy} + r_{33}a_{iz})(z - b_{iz}) - 2(xb_{ix} + yb_{iy} + zb_{iz}) \end{aligned} \quad (\text{A.17})$$

In Figure A2, since \mathbf{a}_i^A and \mathbf{b}_i^B lie on the $x_A - y_A$ plane and $x_B - y_B$ plane, respectively, we have the following:

$$a_{iz} = b_{iz} = 0 \quad (\text{A.18})$$

$$a_{ix}^2 + a_{iy}^2 + a_{iz}^2 = r_A^2 \quad (\text{A.19})$$

$$b_{ix}^2 + b_{iy}^2 + b_{iz}^2 = r_B^2 \quad (\text{A.20})$$

Now in light of (A.18)-(A.20), (A.17) becomes the following:

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

Equation (A.21) presents the closed-form solution of the inverse kinematics problem, in the sense that the actuator lengths l_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, (A.21) 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 [30].

Appendix D. Dynamical Analysis of the 6 DOF CKCM Manipulator

A detailed dynamical analysis of the 6 DOF CKCM robot manipulator discussed in Appendix C can be found in [17]. We present below only results pertinent to the computer simulation study. First to compactly represent the dynamical equations presented later, we introduce the following convention:

- Cartesian Coordinates Vector

$$\Phi = (\phi_1 \phi_2 \phi_3 \phi_4 \phi_5 \phi_6)^T = (x_1 \ x_2 \ x_3 \ \eta_1 \ \eta_2 \ \eta_3)^T = (x \ y \ z \ \alpha \ \beta \ \gamma)^T \quad (\text{A.22})$$

Note: $\alpha \beta \gamma$ are the Z-Y-X Euler Angles

- Cartesian Coordinates Velocity Vector

$$\dot{\Phi} = (\dot{\phi}_1 \dot{\phi}_2 \dot{\phi}_3 \dot{\phi}_4 \dot{\phi}_5 \dot{\phi}_6)^T = (\dot{x}_1 \dot{x}_2 \dot{x}_3 \dot{\eta}_1 \dot{\eta}_2 \dot{\eta}_3)^T = (\dot{x} \dot{y} \dot{z} \dot{\alpha} \dot{\beta} \dot{\gamma})^T \quad (\text{A.23})$$

- Cartesian Translational Velocity Vector

$$v_B = (\dot{x}_1 \dot{x}_2 \dot{x}_3)^T = (\dot{x} \dot{y} \dot{z})^T \quad (\text{A.24})$$

- Angular Velocity Vector

$$\Omega_B = (\omega_1 \omega_2 \omega_3)^T = (\omega_x \omega_y \omega_z)^T \quad (\text{A.25})$$

- Moment of Inertia

$I_1 = I_x$ = moment of inertia around the x-axis

$I_2 = I_y$ = moment of inertia around the y-axis

$I_3 = I_z$ = moment of inertia around the z-axis

- Cartesian Force/Torque Vector

$$(\tau_1 \tau_2 \tau_3 \tau_4 \tau_5 \tau_6)^T = (f_x f_y f_z M_\alpha M_\beta M_\gamma)^T \quad (\text{A.26})$$

The Lagrangian formulation describes the behavior of the dynamic system in terms of work and energy stored in the system rather than in terms of forces and moments of the individual members involved. Since no closed-form solutions exist for the forward kinematics of the manipulator, we cannot express Cartesian position and orientation of the moving platform in terms of the lengths of the links. Consequently, the generalized coordinates are selected to be the Cartesian coordinates ϕ_j for $j = 1, 2, \dots, 6$. The general form of Lagrangian equations of motion the 6 DOF CKCM is presented by:

$$F_j = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} \quad (\text{A.27})$$

where the Lagrangian L is computed by

$$L = K - P \quad (\text{A.28})$$

q_j and F_j are the generalized coordinates and the generalized force/torque, respectively, K and P denote the kinetic energy and the potential energy of the manipulator end-effector, respectively.

The total kinetic energy K of the end-effector consists of the kinetic energy created from the translational and rotational motion of the moving platform with respect to the base platform and the kinetic energy produced by the rotational motion of the links about the ball joints and the translational motion of the links along the prismatic joints. Thus K can be computed by:

$$K = \frac{1}{2} \sum_{i=1}^6 (m l_{ci}^2 \dot{\theta}_i^2 + m l_{ci}^2 \dot{\theta}_i^2) + \frac{1}{2} \sum_{k=1}^3 M \dot{x}_k^2 + \frac{1}{2} \sum_{k=1}^3 I_k \omega_k^2 \quad (\text{A.29})$$

where l_{ci} is the distance between the center of gravity of the i th link and the attachment point B_i , m is the total mass of the i th link and M is the mass of the moving platform. Furthermore, θ_i is the angle formed between the i th link and its perpendicular projection on the base platform surface as shown in Figure A5 below.

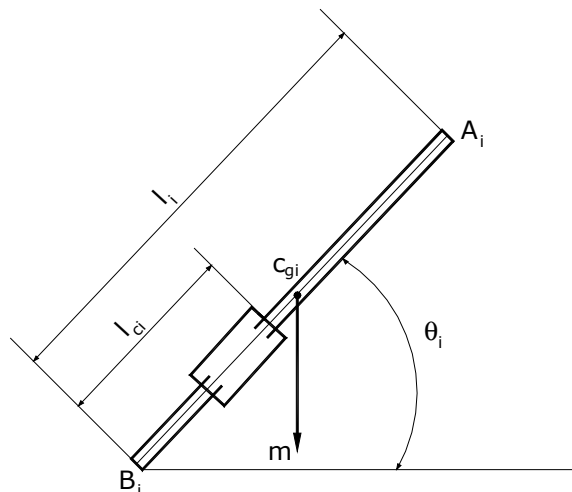


Figure A5. Free body diagram of the i^{th} link.

Similarly, the total potential energy P consisting of the potential energy of the moving platform and the potential energy of the links is computed by

$$P = Mgz + mg \sum_{i=1}^6 l_{ci} \sin \theta_i \quad (\text{A.30})$$

Using (D.7-D.9) in (D.6) and after some mathematical manipulations we obtain the following equations of motion for the manipulator:

$$F_j = \left\{ \sum_{i=1}^6 \left[ml_{ci}^2 \frac{\partial \dot{\theta}_i}{\partial \phi_j} \ddot{\theta}_i + m \frac{\partial l_{ci}}{\partial \phi_j} \ddot{l}_{ci} \right] + \sum_{k=1}^3 \left[M \frac{\partial \ddot{x}_k}{\partial \phi_j} \ddot{x}_k + I_k \frac{\partial \dot{\omega}_k}{\partial \phi_j} \dot{\omega}_k \right] \right\}$$

Inertial Term

$$\begin{aligned} & + \left\{ \sum_{i=1}^6 \left[ml_{ci}^2 \frac{d}{dt} \left(\frac{\partial \dot{\theta}_i}{\partial \phi_j} \right) \dot{\theta}_i + ml_{ci} \frac{d}{dt} \left(\frac{\partial \dot{l}_{ci}}{\partial \phi_j} \right) + 2ml_{ci} \dot{\theta}_i \dot{l}_{ci} \frac{\partial \dot{\theta}_i}{\partial \phi_j} \right. \right. \\ & \quad \left. \left. - ml_{ci} \dot{\theta}_i^2 \frac{\partial l_{ci}}{\partial \phi_j} - ml_{ci}^2 \dot{\theta}_i \frac{\partial \dot{\theta}_i}{\partial \phi_j} - ml_{ci} \frac{\partial \dot{l}_{ci}}{\partial \phi_j} \right] \right. \\ & \quad \left. + \sum_{k=1}^3 \left[I_k \omega_k \frac{d}{dt} \left(\frac{\partial \omega_k}{\partial \phi_j} \right) - I_k \omega_k \frac{\partial \omega_k}{\partial \phi_j} \right] \right\} \end{aligned}$$

Coriolis and Centrifugal Term

$$+ \left\{ \sum_{i=1}^6 \left[mg \frac{\partial l_{ci}}{\partial \phi_j} \sin \theta_i + mgl_{ci} \cos \theta_i \frac{\partial \theta_i}{\partial \phi_j} \right] + Mg \frac{\partial z}{\partial \theta_j} \right\} \quad (\text{A.31})$$

Gravity Term

for $j = 1, 2, \dots, 6$. Equation (A.31) represents the relationship between the Cartesian force/torque F_j applied to the moving platform and the Cartesian position and orientation of the moving platform ϕ_j , for $j = 1, 2, \dots, 6$.

References

1. Ahn, H.S.; Chen, Y.; Moore, K.L. Iterative learning control: brief survey and categorization. *IEEE Trans. Syst., Man, Cybern.* **2007**, *6*, 1099–1121.
2. Bristow, D.A.; Tharayil, M.; Alleyne, A.G. A Survey of Iterative Learning Control. *IEEE Control Syst.* **2006**, *26*, 96–114, doi:10.1109/MCS.2006.1636313.

3. Wang, Y.; Gao, F.; Doyle, F.J. Survey on Iterative Learning Control, Repetitive Control, and Run-to-Run Control. *Journal of Process Control* **2009**, *19*, 1589–1600, doi:10.1016/j.jprocont.2009.09.006.
4. Arimoto, S.; Kawamura, S.; Miyazaki, F. Bettering Operation of Robots by Learning. *J. Robotic Syst.* **1984**, *1*, 123–140, doi:10.1002/rob.4620010203.
5. Arimoto, S. Robustness of Learning Control for Robot Manipulators. In Proceedings of the Proceedings., IEEE International Conference on Robotics and Automation; IEEE Comput. Soc. Press: Cincinnati, OH, USA, 1990; pp. 1528–1533.
6. Xu, J.-X. A Survey on Iterative Learning Control for Nonlinear Systems. *International Journal of Control* **2011**, *84*, 1275–1294, doi:10.1080/00207179.2011.574236.
7. Tsai, C.S.; Chen, W.; Yun, D.; Tomizuka, M. Iterative Learning Control for Vibration Reduction in Industrial Robots with Link Flexibility. In Proceedings of the 2013 American Control Conference; IEEE: Washington, DC, June 2013; pp. 5195–5200.
8. Zhang, D.; Wei, B. On the Development of Learning Control for Robotic Manipulators. *Robotics* **2017**, *6*, 23, doi:10.3390/robotics6040023
9. Ernesto, H.; Pedro, J.O. Iterative Learning Control with Desired Gravity Compensation under Saturation for a Robotic Machining Manipulator. *Mathematical Problems in Engineering* **2015**, 1–13, doi:10.1155/2015/187948.
10. Ichijo, Y.; Murao, T.; Kawai, H.; Fujita, M. Passivity-Based Iterative Learning Control for 2DOF Robot Manipulators with Antagonistic Bi-Articular Muscles. In Proceedings of the 2014 IEEE Conference on Control Applications (CCA); IEEE: Juan Les Antibes, France, October 2014; pp. 234–239.
11. Jia, B.; Liu, S.; Liu, Y. Visual Trajectory Tracking of Industrial Manipulator with Iterative Learning Control. *Industrial Robot: An International Journal* **2015**, *42*, 54–63, doi:10.1108/IR-09-2014-0392.
12. Su, Y.; Zheng, C. A Simple Repetitive Learning Control for Asymptotic Tracking of Robot Manipulators with Actuator Saturation. *IFAC Proceedings Volumes* **2011**, *44*, 6886–6891, doi:10.3182/20110828-6-IT-1002.00493.
13. Lee, T.Y.; Shim, J.K. Forward kinematics of the general 6–6 Stewart platform using algebraic elimination. *Mechanism and Machine Theory* **2001**, *36*, 1073–1085.
14. Lee, S.H.; Song, J.B.; Choi, W.C.; Hong, D. Position control of a Stewart platform using inverse dynamics control with approximate dynamics. *Mechatronics* **2003**, *13*, 605–619.
15. Su, Y.X.; Duan, B.Y.; Zheng, C.H.; Zhang, Y.F.; Chen, G.D.; Mi, J.W. Disturbance-Rejection High-Precision Motion Control of a Stewart Platform. *IEEE Trans. Contr. Syst. Technol.* **2004**, *12*, 364–374, doi:10.1109/TCST.2004.824315.
16. Reynolds, K.L. Adaptive Compliance Control of a Parallel Manipulator for Upper Extremity Rehabilitation. Master Thesis, The Catholic University of America, Washington, D.C., 2012.
17. Nguyen, C.C.; Pooran, F.J. Dynamic Analysis of a 6 DOF CKCM Robot End-Effector for Dual-Arm Telerobot Systems. *Robotics and Autonomous Systems* **1989**, *5*, 377–394, doi:10.1016/0921-8890(89)90022-5.
18. 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.
19. 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.
20. Bouakrif, F.; Boukhetala, D.; Boudjema, F. Velocity observer-based iterative learning control for robot manipulators. *Int. J. Syst. Sci.* **2013**, *44*, 214–222. [[Google Scholar](#)] [[CrossRef](#)]
21. Nguyen, C.C.; Nguyen, T.T.; Duong, T.C.; Nguyen, M.T.; Ngo, T.T.H.; Lu, S. Real-time experiments for decentralized adaptive synchronized motion control of a closed-kinematic chain mechanism robot manipulator. *Machines* **2025**, *13*, 652.
22. Queen, M.P.; Kuma, M.; Aurtherson, P. Repetitive Learning Controller for Six Degree of Freedom Robot Manipulator. *Int. Rev. Autom. Control* **2013**, *6*, 286–293. [[Google Scholar](#)]
23. Bristow, D.A.; Singler, J.R. Analysis of transient growth in iterative learning control using Pseudospectra. In Proceedings of the Symposium on Learning Control at IEEE CDC 2009, Shanghai, China, 14–15 December 2009; pp. 1–6. [[Google Scholar](#)]

24. Cervantes, I.; Alvarez-Ramirez, J. On the PID Tracking Control of Robot Manipulators. *Systems & Control Letters* **2001**, *42*, 37–46, doi:10.1016/S0167-6911(00)00077-3.
25. Hazem, Z.B.; Saidi, F.; Guler, N.; Altaif, A.H. A Hybrid Reinforcement Learning Framework Combining TD3 and PID Control for Robust Trajectory Tracking of a 5-DOF Robotic Arm. *Automation* **2025**, *6*, 56. <https://doi.org/10.3390/automation6040056>
26. Ngo, T.Q.; Nguyen, M.; Wang, Y.; Ge, J. An adaptive iterative learning control for robot manipulator in task space. *Int. J. Comput. Commun. Control* **2012**, *7*, 518–529. [[Google Scholar](#)] [[CrossRef](#)]
27. Bukkems, B.; Kostic, D.; Jager, B.; Steinbuch, M. Learning-Based Identification and Iterative Learning Control of Direct-Drive Robots. *IEEE Trans. Control Syst. Technol.* **2005**, *13*, 537–549. [[Google Scholar](#)] [[CrossRef](#)]
28. Armin, S.; Goele, P.; Jan, S. Optimization-based iterative learning control for robotic manipulators. In Proceedings of the Benelux Meeting on Systems and Control, Soesterberg, The Netherlands, 22–24 March 2016. [[Google Scholar](#)]
29. Marino, R.; Tomei, P.; Verrelli, C.M. Robust adaptive learning control for nonlinear systems with extended matching unstructured uncertainties. *Int. J. Robust Nonlinear Control* **2012**, *22*, 645–675. [[Google Scholar](#)] [[CrossRef](#)]
30. Merlet, J.P. Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis. *Int. J. Robot. Res.* **2004**, *23*, 221–235.

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.