International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V73I8P125

Volume 73 Issue 8, 292-302, August 2025

Original Article

Comparative and Analysis of Adaptive Robust Controller and Intelligent Adaptive Iterative Learning Controller Uncertainty Compensation for Industrial Robot

Vo Thu Ha[1], Bui Thi Hong Diem[2], Than Thi Thuong[3], Tran The Anh[4], Trieu Tuan Anh[5]

[1,2,3]*Faculty of Electrical-Automation Engineering, University of Economics - Technology for Industries, Hanoi, Vietnam.*
[4]*Office Admissions & Communication, University of Economics - Technology for Industries, Hanoi, Vietnam.*
[5]*Nhat Minh Technology Company Limited, Hanoi, Vietnam.*

[1]*Corresponding Author : vtha@uneti.edu.vn*

Received: 27 March 2025 Revised: 12 August 2025 Accepted: 27 August 2025 Published: 30 August 2025

Abstract - *The robot system's parameters fluctuate unpredictably or cannot be precisely determined and are further influenced by external disturbances during movement. Various adaptive, robust, and adaptive-robust control methods exist for robot motion control, all requiring an uncertain mathematical model, parameter estimation, or the assumption of constant uncertainty. However, an alternative approach—iterative learning control—does not rely on a mathematical model or assume parameter constancy. Instead, it determines learning function parameters online using an optimization method based on minimizing the sum of squared errors. This article explores and compares control performance for a 2-degree-of-freedom robot, highlighting the effectiveness of the adaptive-robust controller versus the conventional learning controller.*

Keywords - *Adaptive-Robust Control, Interactive Learning Control (ILC), Taylor series estimation.*

1. Introduction

Industrial robots are objects that work in cycles, performing repetitive tasks to ensure accuracy in speed and position. However, industrial robots are highly nonlinear objects with many uncertain parameters, are affected by inter-channel effects between joints, and are subject to external interference, which are the causes of trajectory tracking errors, so research to improve the quality of precise trajectory tracking motion control of industrial robots is always of interest to many domestic and foreign scientists. That is why many control methods for industrial robots have been developed, from traditional to using artificial intelligence, including iterative learning control.

2. Literature Review

Conventional control methods are usually model-based, which are control methods designed from the Euler-Lagrange model of robots (1). According to the document, these control methods are built and classified on the basis of known parameters and external disturbances or uncertain components of the model, precisely the gravity compensation PD control method [1]. This control method requires knowing the exact parameters of the object, while the actual industrial robot object has many constant and uncertain parameters. Therefore, implementing controllers according to this method is challenging in terms of accuracy, or in other words, difficult to implement in practice. The fuzzy control method, according to documents [2, 3], is a new method that has appeared in recent years but has had some applications in practice. The essential advantage of fuzzy control compared to other controls is that it can synthesize the controller without knowing the characteristics of the object accurately in advance. Fuzzy control has transferred biological systems' information processing and control principles to technical systems. Therefore, fuzzy control has successfully solved many complex control problems that were previously unsolved. The disadvantage of the method is that it is necessary to have expert knowledge and operating experience to build a suitable fuzzy controller when synthesizing. There have been many studies in different directions to standardize the design and optimization of fuzzy controllers. The control method uses neural networks (Neural Networks: NN), according to documents [4-6]. This control method has the advantage of parallel processing, so the information processing speed is very high. Due to the ability to "learn," the network promises to apply interference in the field of science and technology, especially controlling complex dynamic systems (systems with strong nonlinearity, systems with unknown parameters, or known but incomplete or inaccurate parameters) with high control accuracy and small control time, good anti-interference, stable and sustainable, capable of controlling objects with the same model with different parameters, even variations. The disadvantage of the method is the complexity of synthesizing the neural controller and the low feasibility of practical applications. According to document [1], the robust control method was born in the 1960s and is used for systems in which one or more parameters are not accurately modeled or cannot be modeled. Therefore, the robust control problem aims to control the system to change within an allowable range or the uncertain disturbance changes continuously without having to model the entire system. The limitation of this method is that the controller requires pre-determining the boundaries of the system parameters, and the appearance of external disturbances cannot ensure the asymptotic stability of the trajectory tracking error.

According to documents [7-9], the adaptive control method was born after 1950 and is used in automatic systems. In essence, the adaptation process measures state parameters such as errors and output signals to evaluate the system's current state. Using calculation tools, the controller will automatically determine the parameters and adjust them accordingly to achieve the control characteristics as desired. The adaptation process occurs according to the change in system parameters or the impact of disturbances. Therefore, the structure of the controller usually consists of two parts: the adaptation part to update and identify data and the control part to determine the control signal. Each of the above control methods contributes to the application in some way.

Although many control methods exist, the tracking quality is not always satisfactory due to insufficient model accuracy or unrepresentative impact disturbances. The reasons may be limitations in theory, modeling understanding, incompatibility with traditional control methods, or unforeseen changes in the system. Although the initial model is accurate, material aging and actuator changes reduce the control quality over time. Redesigning the controller or replacing equipment wastes essential system information and increases maintenance costs. Research on integrating intelligent control methods to calibrate control signals without redesigning or replacing equipment is necessary to optimize maintenance time and improve operational efficiency.

One of the suitable methods is Iterative Learning Control (ILC) [10-22], which relies on experience to improve current and future control quality. With ILC, one does not need to re-adjust the traditional controller or intervene deeply in the existing system. However, ILC cannot be successfully applied to all volatile systems. The quality of the ILC depends significantly on the dynamics and the reasonable selection of the tuning law. Therefore, it is necessary to research and find an intelligent solution to pre-intervene in the system, creating the possibility of applying iterative learning control to it and determining the tuning law applicable to many classes of systems. The main content of the paper is to compare and analyze the performance between traditional controllers, such as adaptive stability controllers and Iterative Learning Controllers (ILC), for 2-degree-of-freedom robots to evaluate the ability to improve control quality, reduce trajectory tracking errors, and ensure system stability. The paper analyzes how the adaptive robust controller handles disturbances and uncertainties in the robot dynamics model compared to the ILC's ability to learn from past data to improve accuracy. From there, the study proposes a suitable approach to optimize control performance and improve robot reliability during operation.

The content of the paper is presented in 5 parts: Part 1 is a general introduction, Part 2 is the adaptive-robust control method, Part 3 is the theory of ILC iterative learning control and its application to control industrial robot systems, and Part 4 is the conclusion and future research directions.

3. Robot Control using an Adaptive Robust Control Method

3.1. Algorithmic Content, [1]

Starting from the robot's dynamic equation written in the form:

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + T_d \quad (1)$$

In this case, T is the nx1 vector describing the unknown external noise.

Suppose the robot's dynamics equations can be rewritten as follows:

$$\omega = M(q)(\ddot{q}_d + \Lambda\dot{e}) + C(q,\dot{q})(\dot{q}_d + \Lambda e) + G(q) + T_d \quad (2)$$

The uncertainty here is the uncertainty in the amount of external disturbance load, the friction coefficient. Therefore, a positive scalar function ρcan be used to limit the uncertainty ρ, which can be given as follows:

$$\rho \geq \|w\| \quad (3)$$

The physical properties that can be used to show that the given robot's dynamic equations can be bounded are as follows:

$$\rho = \delta_0 + \delta_1\|e\| + \delta_2\|e\|^2 \geq \|w\| \quad (4)$$

With:

$$e = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \quad (5)$$

$\delta_1, \delta_2, \delta_3$ Positive limit constants are calculated based on the maximum value of load mass, connecting rod mass, friction coefficient, and external noise. As is known, the robust controller requires the bounding region defined in (4) to have a pre-determined shape. The adaptive robust controller studied here will "learn" the online bounding regions that the adaptive component determines when the robot arm moves.

The proposed adaptive robust controller is as follows:

$$\tau = K_v r + \frac{r\hat{\rho}^2}{\hat{\rho}\|r\|+\varepsilon} \tag{6}$$

In there, $K_v r$ the persistence component $\frac{r\hat{\rho}^2}{\hat{\rho}\|r\|+\varepsilon}$ is the adaptation component

K_v is a positive definite diagonal nxn matrix
V_R is the auxiliary control nx1 vector
$r = \Lambda e + \dot{e}$
$\dot{\varepsilon} = -k_\varepsilon \varepsilon, \varepsilon(0) > 0$
k_ε: is a positive control constant

$$\hat{\rho} = \hat{\delta}_0 + \hat{\delta}_1\|e\| + \hat{\delta}_2\|e\|^2 \tag{7}$$

$\hat{\delta}_1, \hat{\delta}_2, \hat{\delta}_3$ These are the dynamic estimates of the corresponding limiting constants $\delta_1, \delta_2, \delta_3$ above. Those estimates marked with " ^ " are continuously updated by the adaptive component and are written in a different form as follows:

$$\hat{\rho} = S\hat{\theta} \tag{8}$$

In there:

$$S = [1 \quad \|e\| \quad \|e\|^2] \quad \text{and} \quad \hat{\theta} = [\hat{\delta}_1 \quad \hat{\delta}_2 \quad \hat{\delta}_3]^T$$

Therefore, ρ can be written in matrix form as follows:

$$\rho = S\theta \tag{9}$$

In there: $\theta = [\delta_0\delta_1\delta_2]^T$

The updated rule is given as follows:

$$\dot{\hat{\theta}} = \gamma S^T\|r\| \tag{10}$$

In there γ is a positive control constant.

Set:

$$\tilde{\theta} = \theta - \hat{\theta} \tag{11}$$

Therefore:

$$\dot{\tilde{\theta}} = -\gamma S^T\|r\| \tag{12}$$

The proposed adaptive-stable controller above makes the Robot system stable according to the Lyapunov criterion. Choose a positive definite Lyapunov function:

$$V = \frac{1}{2}r^T M(q) r + \frac{1}{2}\tilde{\theta}^T\gamma^{-1}\tilde{\theta} + k_\varepsilon^{-1}\varepsilon \tag{13}$$

$$\Rightarrow \dot{V} = r^T M(q)\dot{r} + \frac{1}{2}r^T\dot{M}(q)r + \tilde{\theta}^T\gamma^{-1}\dot{\tilde{\theta}} + k_\varepsilon^{-1}\dot{\varepsilon} \tag{14}$$

From this, we have:

$$\omega - \tau = M(q)\dot{r} + C(q,\dot{q})r \tag{15}$$

Set $v_R = \frac{r\hat{\rho}^2}{\hat{\rho}\|r\|+\varepsilon}$ then we have:

$$M\dot{r} = -Cr + \omega - K_v r - v_R \tag{16}$$

$$\Rightarrow \dot{V} = r^T(-Cr + \omega - K_v r - v_R) + \frac{1}{2}r^T\dot{M}r + \tilde{\theta}^T\gamma^{-1}\dot{\tilde{\theta}} + k_\varepsilon^{-1}\dot{\varepsilon}$$

$$= -r^T K_v r + r^T(\omega - v_R) - \frac{1}{2}r^T(2C(q,\dot{q}) - \dot{M}(q))r + \tilde{\theta}^T\gamma^{-1}\dot{\tilde{\theta}} + k_\varepsilon^{-1}\dot{\varepsilon} \tag{17}$$

Because $S = 2C - \dot{M}$ it is the skew matrix

So $\frac{1}{2}r^T(2C - \dot{M})r = 0$

And

$$\dot{\tilde{\theta}} = -\gamma S^T\|r\| \Rightarrow \gamma^{-1}.\dot{\tilde{\theta}} = -S^T\|r\|$$

$$\Rightarrow \tilde{\theta}^T\gamma^{-1}\dot{\tilde{\theta}} = -\tilde{\theta}^T S^T\|r\| = -S\tilde{\theta}\|r\| \tag{18}$$

$$\Rightarrow \dot{V} = -r^T K_v r + r^T(\omega - v_R) - S\tilde{\theta}\|r\| + k_\varepsilon^{-1}\dot{\varepsilon} \tag{19}$$

Due to $\rho \geq \|w\|$ and $\rho = S\theta$

$$\Rightarrow \dot{V} \leq -r^T K_v r - S\tilde{\theta}\|r\| + S\theta\|r\| - r^T v_R + k_\varepsilon^{-1}\dot{\varepsilon}$$

$$\Rightarrow \dot{V} \leq -r^T K_v r + S\hat{\theta}\|r\| - \frac{r^T r(S\hat{\theta})^2}{S\hat{\theta}\|r\| + \varepsilon} - \varepsilon$$

$$\Rightarrow \dot{V} \leq -r^T K_v r + S\hat{\theta}\|r\| - \frac{\|r\|^2(S\hat{\theta})^2}{S\hat{\theta}\|r\|+\varepsilon} - \varepsilon$$

$$\Rightarrow \dot{V} \leq -r^T K_v r + \frac{\varepsilon S\hat{\theta}\|r\|}{S\hat{\theta}\|r\|+\varepsilon} - \varepsilon \tag{20}$$

So

$$\frac{\varepsilon S\hat{\theta}\|r\|}{S\hat{\theta}\|r\|+\varepsilon} - \varepsilon \leq 0 \quad \Rightarrow \quad \dot{V} \leq -r^T K_v r$$

$$\Rightarrow \dot{V} \leq 0 \tag{21}$$

Thus, according to the Lyapunov stability theorem, the system is stable with the given control law. In general, the torque controller is designed as follows:

$$\tau = K_v r + \frac{r\hat{\rho}^2}{\hat{\rho}\|r\|+\varepsilon} \tag{23}$$

with

$$\hat{\rho} = S\hat{\theta} = \left[1 \left\|\begin{bmatrix} e \\ \dot{e} \end{bmatrix}\right\| \left\|\begin{bmatrix} e \\ \dot{e} \end{bmatrix}\right\|^2\right] [\delta_0 \delta_1 \delta_2]^T$$

$$r = e + \dot{e} \quad và \quad \dot{\varepsilon} = -k_\varepsilon \varepsilon \quad và \quad K_v = k_v I$$

The update rule estimates the bounds for the parameters:

$$\dot{\hat{\theta}} = \gamma S^T \|r\| \tag{24}$$

The structure diagram of the adaptive robust controller is shown in Figure 1:

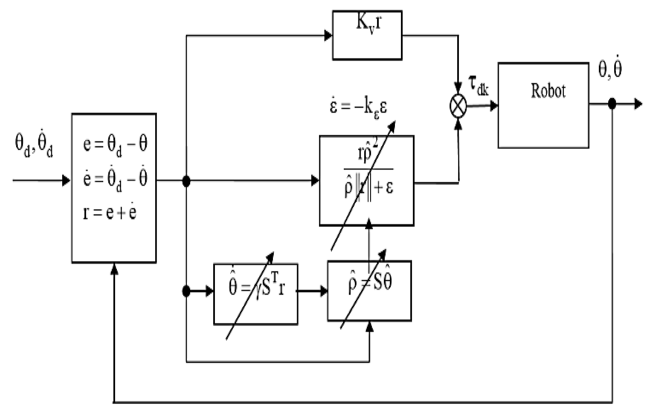


Fig. 1 Schematic diagram of a sustainable-adaptive control structure

3.2. Simulation and Verification via a Robot with 2 Degrees of Freedom using Matlab/Simulink Software

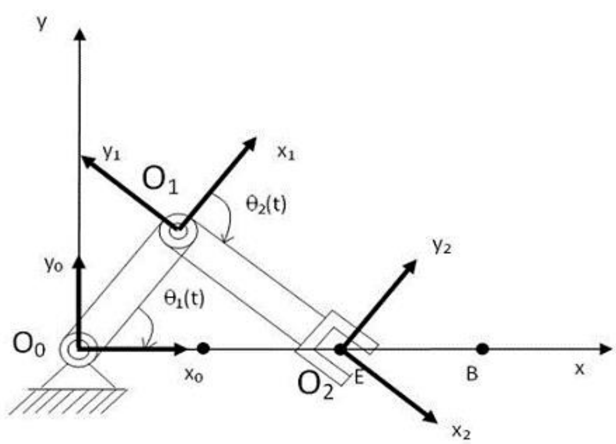


Fig. 2 2-DOF robot

Results of the Motivational Learning Method

$M_{11} = m_1 a_{c1}^2 + I_1 + m_2 a_1^2 + I_2$

$M_{12} = M_{21} = M_{22} = I_2, H_{11} = H_{21} = 0$

$G_{11} = m_1 g a_{c1} c\theta_1 + m_2 g a_1 c\theta_1$

$G_{21} = 0$

From (23), we can determine the input sustainable-adaptive control law of the form:

$$\begin{cases} \tau_{dk_1} = k_v r_1 + r_1 \hat{\rho}^2 \frac{1}{\hat{\rho}\|r\|+\varepsilon} \\ \tau_{dk_2} = k_v r_2 + r_2 \hat{\rho}^2 \end{cases}$$

With:

$$K_v = k_v I; r_1 = \Lambda e_1 + \dot{e}_1, r_1 = \Lambda e_2 + \dot{e}_2, r_2 = \Lambda e_2 + \dot{e}_2$$

$$\dot{\varepsilon} = -k_\varepsilon \varepsilon, \|r\| = \sqrt{r_1^2 + r_2^2}$$

$$\hat{\rho} = S\hat{\theta} = [1 \quad \|e\| \quad \|e\|^2][\delta_1 \quad \delta_2]^T$$

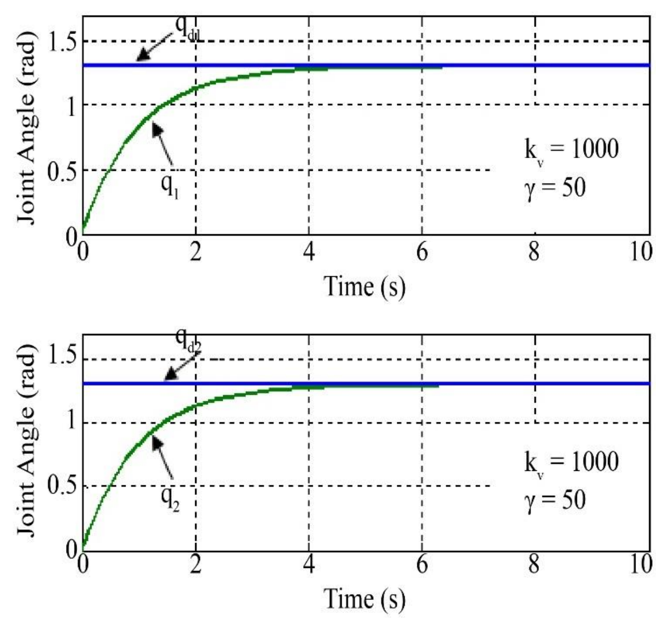
$$\|e\| = \sqrt{e_1^2 + e_2^2 + \dot{e}_1^2 + \dot{e}_2^2}$$

Updated law:

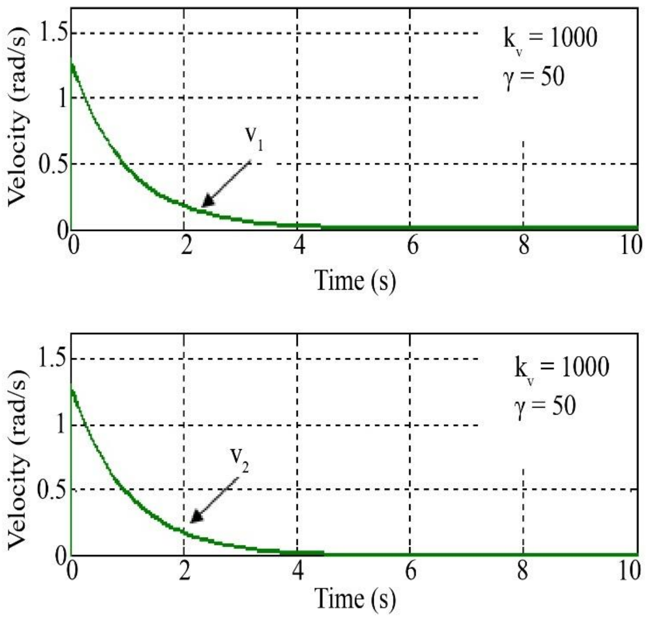
$$\dot{\delta}_0 = \gamma(r), \quad \dot{\delta}_1 = \gamma\|e\|\|r\|, \quad \dot{\delta}_2 = \gamma\|e\|^2\|r\| \tag{25}$$

Table 1. Controller parameters

Symbol	Parameter name	Joint axis parameter values
k_v	Adjustment factor sustainability component	$k_v = 1000$
q_d	The set value of the joint axes	$q_{d_1} = 1.3(rad), q_{d_2} = 1.3(rad)$
γ	Adaptive component correction factor	$\gamma = 50$



(a)



(b)

Fig. 3 Position and velocity deviations of joints

3.2.1. Comments

The robot simulation results show that the joint angle error between the set and actual values is approaching 0, with no oscillation. The speed value also approaches 0.

The sustainable-adaptive control method combines the two methods of sustainable control and adaptive control. Therefore, the sustainable-adaptive controller will include sustainable and adaptive components. The sustainable component will stabilize the system in a specific bounding area, while the adaptive component always updates the change in the bounding area. Therefore, when using the sustainable-adaptive controller, the system is always stable.

Over time, environmental factors like equipment wear gradually deteriorate the original design's quality, making controller reconstruction necessary. In traditional industrial robot motion control design, a fundamental requirement is a clear understanding of the control object, which must be represented by a mathematical model, such as a transfer function or a system of first-order differential equations.

Additionally, unforeseen external influences can reduce system performance, requiring reevaluation of the mathematical model, assessment of objective impacts, and adjustments or redesign of the controller. To address these disadvantages, intelligent control methods can be used, which do not rely on the robot's dynamic model (1), ensuring control quality remains unaffected by components q(t) and d(t). This article focuses on the iterative learning control method, emphasizing the online optimization of standard arithmetic function parameters by minimizing the sum of squared errors and applying it to 2-degree-of-freedom robots.

4. Iterative Learning Control Theory and its Application in Industrial Robot System Control

The paper aims to use an iterative learning controller with a linear learning function to control nonlinear processes. To achieve this goal, there are two main tasks:

First: Determine the principle of reasonable correction of the control signal from experience for the iterative learning controller. In other words, it would be best if the correction of the control signal, including the selection of the convergence parameter for the learning function, does not depend on the mathematical model of the process.

Second: If the deviation of the mathematical model of the process and the deviation of the actuator are considered total disturbances, it is necessary to intelligently estimate this total disturbance component, including the function uncertainty components, without using the mathematical model of the system. This, through the control of total disturbance compensation, will expand the scope of applying the combined control method between ILC and traditional techniques.

4.1. Determine the Optimal Standard Arithmetic Function Parameters Online by Minimizing the Sum of Squared Errors

Consider a linear discrete system in the form:

$$\begin{cases} x_k(i+1) = A_{x_k}(i) + B_{u_k}(i) \\ \qquad y_k(i) = C_{x_k}(i) \end{cases} \quad (26)$$

With: $A \in R^{nxn}, B \in R^{nxm}, C \in R^{mxn}$ are the system matrix, control matrix, and output matrix; $x(t) \in R^n, u(t) \in R^m, y(t) \in R^m$ are vectors of state, input, and output signals.

When using D-style learning functions:

$$u_{k+1}(i) = u_k(i) + Ke_k(i+1), i = 0,1,\dots,N-1 \quad (27)$$

with $e_k(i) = r_k(i) - y_k(i)$

Where: k is the trial index; i is the specific time in each trial, corresponding to time $t = kT + iT_s$. The preset signal is assumed to be periodic with period T.

The convergence condition is given as follows:

$$\|I - CBK\| < 1 \quad (28)$$

4.1.1. Comment

With the assumption $CB \neq 0$, [22] is not only a sufficient condition but also a necessary condition. So from (28), the content of the article proposes to determine the parameter matrix K for the learning function (27) that can change adaptively according to the tracking error through each kth trial, called the online learning function parameter, according to the optimization criterion of minimizing the sum of squared tracking errors, denoted by. The proof of the necessity and sufficiency of condition (28) is shown in the documents [11, 21].

With the necessary and sufficient condition (28) showing that the system (26) has $CB \neq 0$an infinite number of parameters, K will exist that makes the learning and tuning process with the learning function P converge. However, if the system $CB = 0$is a matrix with all elements equal to 0, there will not exist any convergent parameter K. So the content of the article will present the case of creating many convergent parameters, we can determine a parameter with the best convergence speed as well as in the case where there is no convergent parameter K in the sense that $\|\varepsilon_k\| \to 0$we can at least determine a parameter K that makes $\|\varepsilon_k\| \to min$

According to [11, 21], it is proposed to use the standard

$$\|I - CBK\| \to min \quad (29)$$

To determine the parameter K, replace the condition (25). So, the accepted solution will depend on the tracking error vector $\|\underline{\varepsilon}_k\|$, which changes with each test k. So, to be strict, the criterion (29) will be rewritten as:

$$K_k = arg \min_{a \le K \le b} \|\underline{e}_{k+1}\| = arg \min_{a \le K \le b} \|I(-\Phi K)\underline{e}_k\| \quad (30)$$

Where a and b are two appropriately chosen limiting matrices, the matrix comparison $a < K < b$is understood to be performed for each corresponding element of those matrices.

From (30) compared with (28), it can be seen that at each trial k, the online learning coefficient $K_k$will always $\|\varepsilon_{k+1}\|$ have the smallest value compared to the parameter chosen from condition (28). This will increase the convergence for the learning and adjustment process.

So, to facilitate the implementation of (30), we can use another form of the right-hand side:

$$\|(I - \Phi K)\varepsilon_k\|^2 = \varepsilon_k^T (I - \Phi K)^T (I - \Phi K)\varepsilon_k \to min \quad (31)$$

Then the optimization problem (30) becomes:

$$K_k = arg \min_{a \le K \le b} [\varepsilon_k^T (\Phi K)^T (\Phi K)\varepsilon_k - 2\varepsilon_k^T \Phi K \varepsilon_k] \quad (32)$$

With:

$$\Phi = \begin{pmatrix} CB & 0 & \cdots 0 \\ CAB & CB & \cdots 0 \\ \vdots & \vdots & \ddots \vdots \\ -CA^{N-1}B & -CA^{N-2}B & \cdots CB \end{pmatrix}, \varepsilon_k = \begin{pmatrix} e_k(1) \\ e_k(2) \\ \vdots \\ e_k(N) \end{pmatrix}$$

4.2. Stabilization and Linearization without using Mathematical Models

4.2.1. Estimating the Derivative of a Function Vector from Measured Data using Taylor Decomposition

Consider the function vector $x_k(i)$ at time $t_i = kT + iT_s$. If written in the language of iteration $x(t_i) = x_k(i)$, the problem is that from two consecutive measured values of $x_k(i)$ and $x_k(i-1)$, we need to approximate the derivative of $\dot{x}_k(i)$ at that current time.

Use Taylor series analysis to perform the above estimation problem. This formula is as follows [22]. At $t_i = kT + iT_s$ and when $T_s$is very small, simultaneous, and twice differentiable, we have:

$$x_k(i) = D_1^T x_k(i) - D_0^T \dot{x}_k(i-1) + \frac{T_s^2}{2} \ddot{x}_k(\varsigma) \quad (33)$$

with:$D_1^T = -D_0^T = \frac{1}{T_s}$

Are the corresponding Taylor series coefficients and $(i-1)T_s \le \varsigma \le iT_s$. In case $\ddot{x}_k(\varsigma) = 0, \forall \varsigma$the derivative approximation formula (35) becomes exact, i.e., the Taylor series analysis has no errors.

$$\dot{x}_k(i) \approx \frac{x_k(i) - x_k(i-1)}{T_s} \quad (34)$$

This shows that the Taylor series derivative estimation will achieve very high accuracy when the sampling interval T_s is small enough.

4.2.2. Application to the Linearization of Nonlinear System Stability without using A Model

Consider the nonlinear system $q \in R^n$described by:

$$\ddot{q}(t) = f(q, \dot{q}) + u \quad (35)$$

Where$f(q, \dot{q})$ is a vector of uncertain functions, the task is to design a controller to linearize and stabilize the nonlinear system without using the original model (35). To solve the above problem, we apply the derivative approximation formula (34) obtained by Taylor series analysis. First, transform (35) into the form:

$$\ddot{q} = -A_1 q - A_2 \dot{q} + u + d \quad (36)$$

With A_1, A_2 the preference matrix,

$$d = (q, \dot{q}) + A_1 q + A_2 \dot{q} \quad (37)$$

This is the new functional uncertainty component. If we can accurately estimate and compensate for the uncertainty component d, $\hat{d} \approx d$the control signal will have the form:

$$u = v - \hat{d} \quad (38)$$

System (36) becomes:

$$\dot{x} = \begin{pmatrix} 0_n & I_n \\ -A_1 & -A_2 \end{pmatrix} x + \begin{pmatrix} 0_n \\ I_n \end{pmatrix} [\vartheta + \delta_d]$$
$$= Ax + B[\vartheta + \delta_d] \quad (39)$$

With $0_n, I_n$ are zero and identity matrices of dimension nxn, as well as:

$$x = \begin{pmatrix} q \\ \dot{q} \end{pmatrix}, A = \begin{pmatrix} 0_n & I_n \\ -A_1 & -A_2 \end{pmatrix},$$
$$B = \begin{pmatrix} 0_n \\ I_n \end{pmatrix}, \delta_d = d - \hat{d} \approx 0 \quad (40)$$

To make the linear system (39) asymptotically stable, we must choose two matrices, A_1, A_2, so that (40) is a square matrix, all of whose eigenvalues have negative genuine parts. Next, we estimated $d$through $\hat{d}$formula (36), which is rewritten according to the formula of iterative learning, then:

$$\ddot{q}_k(i) = -A_1 q_k(i) - A_2 \dot{q}_k(i) + u_k(i) + d_k(i) \quad (41)$$

or

$$\dot{x}_k(i) = Ax_k(i) + B[u_k(i) + d_k(i)] \quad (42)$$

Replace the left side in (42) with the derivative approximation formula (34) to get:

$$\Leftrightarrow \frac{x_k(i) - x_k(i-1)}{T_s} \approx Ax_k(i) + B[u_k(i) + d_k(i)] \quad (43)$$

Then convert it into the correct formula by replacing $\approx$, $d_k(i)$ with $=, \hat{d}_k(i)$, respectively, we will get:

$$\frac{x_k(i) - x_k(i-1)}{T_s} = Ax_k(i) + B[u_k(i) + \hat{d}_k(i)]$$

$$\Rightarrow \hat{d}_k(i) = (B^T B)^{-1} B^T \left[\frac{x_k(i) - x_k(i-1)}{T_s} - Ax_k(i) - Bu_k(i)\right] = B^T \left[\frac{x_k(i) - x_k(i-1)}{T_s} - Ax_k(i) - Bu_k(i)\right] \quad (44)$$

(45) will be used to estimate $\hat{d}(t)$ in piecewise constant form as follows:

$$\hat{d}_k(t) = \hat{d}_k(i) khikT + iT_s \le t \le kT + (i+1)T_s \quad (45)$$

in there i = 0,1,…,N-1 with $N = \frac{T}{T_s}$. If there is$i \ge N$, then replace$kT + (i+1)T_s$it with $(i+1)T + (i-N)T_s$

To evaluate the quality of the stable linearized controller in formula $u = \vartheta - d$ when using the estimated value $\hat{d}_k(t)$ from formula (45), we have some apparent conclusions as follows:

- If $f(q, \dot{q})$ it is continuous, the smaller the T_s value update, the smaller the δ_d deviation between the actual value and the estimated value
- The estimated formula $\hat{d}_k(i)$ depends only on matrix A and B given by (40), not the original model (35).
- Estimated value $\hat{d}_k(i)$ according to (44) will make the two sides of (43), with the smallest quadratic standard. Indeed, if the symbol of the two sides of (42) is:

$$\varepsilon = Ax_k(i) + B[u_k(i) + d_k(i)] - \frac{x_k(i) - x_k(i-1)}{T_s}$$
$$= Bd_k(i) + \Delta; \Delta = Ax_k(i) + Bu_k(i) - \frac{[x_k(i) - x_k(i-1)]}{T_s}$$

Then, the square root optimization problem

$$d^* = arg \min_{d_k(i)} \|\varepsilon^2\| = arg \min_{d_k(i)} (Bd_k(i) + \Delta)^T B$$
$$d_k(i) + \Delta = arg \min_{d_k(i)} [d_k(i)^T (B^T B) d_k(i) + 2\Delta^T B d_k(i) + \Delta^T \Delta] \quad (46)$$

Will give a single solution:

$$d^* = -(B^T B)^{-1} (\Delta^T B)^T = -B^T \Delta$$

And this solution coincides with (44).

4.3. Control the Structure of Two Loops without using a Robot's Math Model

The structure diagram consists of two loops, as shown in Figure 4. The outer loop is an iterative learning controller to determine the control signal ϑ to make the trajectory of the joint variables q accurately follow the given trajectory R. This iterative learning controller will use a P-type learning function with the learning function parameter K_k adjusted online after each kth trial according to the principle of minimizing the sum of squared tracking errors. The inner circuit is a bright irregular identity identifier E by Umu, thanks to the Taylor banana analysis method to convert the original robot system (1) into a linear system with noise in the input (14) by the compensation control method.

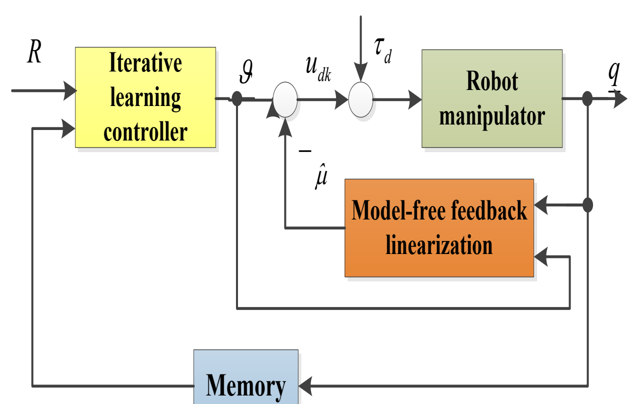


Fig. 4 Structure diagram for robot control

4.3.1. The Inner Ring Control is with an Intelligent Linear Controller, Thanks to a State Feedback

First, to control the inner ring, we need to deploy the formula to estimate the function through the Taylor chain analysis to get the inner ring controller, as shown in Figure 4, helping to transfer the irregular robot system in the form:

$$\dot{x} = Ax + B[\hat{\vartheta} + \mu] \quad (47)$$

with $A = \begin{pmatrix} 0_n & I_n \\ -A_1 & -A_2 \end{pmatrix}, B = \begin{pmatrix} 0_n \\ I_n \end{pmatrix}$ Linear system:

$$\ddot{q} = \vartheta + \mu - \hat{\mu} = \vartheta + \delta \quad (48)$$

The residual uncertainty component is tiny. The task of robust tracking with this residual disturbance will be performed by an iterative learning controller in the outer loop whose learning parameters are determined online. The problem of implementing formula (33) to estimate the derivative of a function vector from its measured values for the industrial robot control problem described by (47), which is then rewritten in the standard form of iterative learning, expressed together with the K-th duty cycle as:

$$\dot{x}_k(\tau) = Ax_k(\tau) + B[\vartheta_k(\tau) + \mu] \quad (49)$$

It has the following characteristics:

First, the derivative value of the function vector $x_k(\tau)$ to be estimated is at the last of the two previous measured values $x_k(\tau)$. Second, the set of all measured values is equidistant, i.e., there is $\tau_i - \tau_{i-1} = T_s$ for all i.

Through these characteristics, the inner loop controller helps to ensure the system's stability by transferring the original system's complex problem to a simpler and more controllable system before transferring it to the ILC iterative learning controller in the outer loop. Based on these two characteristics, the derivative estimation formula of the state vector given in (33) becomes:

$$\dot{x}_k(i) \approx D_1^T x_k(i) - D_0^T x_k(i-1) \quad (50)$$

Substituting (50) into (49), corresponding to $\tau = iT_s$ when $\hat{\mu}$(i-1) it has been complemented from the previous time, we get

$$D_1^T x_k(i) - D_0^T x_k(i-1) \approx Ax_k(i) + B[\vartheta_k(i) + \hat{\mu}(i) - \hat{\mu}(i-1)] \quad (51)$$

From (51), we will use to determine the estimated value directly as follows:

First, we replace $\approx$ and $\mu(i)$ in (51) by = and $\hat{\mu}(i)$

$$D_1^T x_k(i) - D_0^T x_k(i-1) = \\ = Ax_k(i) + B[\vartheta_k(i) + \mu(i) - \hat{\mu}(i-1)] \quad (52)$$

Next, with (52), we can deduce the estimated value of the uncertainty component of the function $\mu(t)$ at the current time $\tau = kT + iT_s$, denoted by $\hat{\mu}(t)$, by the recurrence calculation from $\hat{\mu}(i-1)$ as follows:

$$B\hat{\mu}(i) = D_1^T x_k(i) - D_0^T x_k(i-1) - \\ -Ax_k(i) - B[\vartheta_k(i) - \hat{\mu}(i-1)] \quad (53)$$

or

$$\hat{\mu}(i) = B^T[D_1^T x_k(i) - D_0^T x_k(i-1) - Ax_k(i)] - \\ -\vartheta_k(i) - \hat{\mu}(i-1) \quad (54)$$

Because there is $B^T B = I_n$.

Equation (54) allows us to estimate the uncertainty component $\mu(t)$ accurately from the previous measurement data. The quality of the inner loop controller is influenced by the estimated component $\hat{\mu}(i)$ accuracy, ensuring the linearization system's efficient and stable operation. This helps to reduce the residual noise, improve the ability to track the sample trajectory, and enhance the robustness of the industrial robot system.

4.3.2. Iterative learning controller for outer loop control.

The next task is to design an iterative learning controller for the outer loop, which determines $\vartheta_k(i)$ the controller $x = \begin{pmatrix} q \\ \dot{q} \end{pmatrix}$ *and* $u_{dk} = \vartheta - A_1 q - A_1 \dot{q}$ after compensating $\hat{\mu}(i)$. This is equivalent to constructing the controller:

$$u_k(i) = \vartheta_k(i) - (A_1, A_2)x_k(i) - \hat{\mu} \quad (55)$$

Such that the system $\ddot{q} = u + \mu = \vartheta - \hat{\mu} + \mu = \vartheta + \delta$, when transformed into discrete form with $\vartheta_k(i)$ being a piecewise constant, is described by:

$$\text{“}\begin{cases} x_k(i+1) = \hat{A}x_k(i) + \hat{B}[\vartheta_k(i) + \delta]\text{,,} \\ y_k(i) = \hat{C}x_k(i) \end{cases} \quad (56)$$

Has output $y_k(i)$ following sample trajectory $R(i)$, where:

$$\hat{A} = e^{AT_s}, \hat{B} = \int_0^{T_s} e^{At} B dt, \\ A = \begin{pmatrix} 0_n & I_n \\ -A_1 & -A_2 \end{pmatrix}, B = \begin{pmatrix} 0_n \\ I_n \end{pmatrix}, \hat{C} = (I_n \quad 0_n) \quad (57)$$

In addition, the two optional matrices A_1 và A_2 of the controller (55) also need to make the matrix A defined by (57) Hurwitz (or Schur) to ensure the stability of the system. The iterative learning controller designed here will use the P-type learning function with the parameter K optimally adjusted for each k trial. The update rule is as follows:

$$u_{k+1}(i) = \vartheta_k(i) + K_k e_k(i) \quad (58)$$

with

$$K_k = arg \min_{a \le K \le b} \|(I - \Phi K)\varepsilon_k\| \quad (59)$$

and: $\varepsilon_k(i) = r(i) - y_k(i)$ $\varepsilon_k = \begin{pmatrix} \varepsilon_k(0) \\ \varepsilon_k(1) \\ \vdots \\ \varepsilon_k(N-1) \end{pmatrix}$, $\phi =$

$$\begin{pmatrix} \hat{C}\hat{B} & 0_n & \cdots & 0_n \\ \hat{C}\hat{A}\hat{B} & \hat{C}\hat{B} & \cdots & 0_n \\ \vdots & \vdots & \ddots & \vdots \\ \hat{C}\hat{A}^{N-1}\hat{B} & \hat{C}\hat{A}^{N-2}\hat{B} & \cdots & \hat{C}\hat{B} \end{pmatrix}$$

4.4. Control Algorithm

Algorithm: Model-free optimization of learning parameters in industrial robots' second iterative learning controller.

Choose two matrices A_1,A_2, given in (25), that become Hurwitz.
Determine $\hat{A}, \hat{B}, \hat{C}$ the given in (25) and Φ the given in (28).
Choose $0 < t_s << 1$. Calculate $S = T/t_s$.
Determine $D_1^T = \frac{1}{T_s}$; $D_0^T = -D_1^T$ given in (23)
Choose learning $\hat{\mu}$ and tracking error $\underline{E}_0$.
Allocate $\vartheta(n)$ =R(n), $with$ n = 0,1,...,S-1 and Z = 0.
Choose learning parameter K so that Φ of (28) becomes Schur.
While continuing the control, do
for $n = 0,1,\ldots,S-1$ do
Send $u_{dk} = A_1 q - A_2 \dot{q} - \hat{\mu}$ to the robot for a while t_s
measure $X = vec(E, \dot{E})$, $\underline{Y}(n) = \underline{q}$.
Calculate $\hat{\mu}_k \leftarrow B^T \left[\frac{X-Z}{t_s} - AX\right] - (\vartheta(n) - \hat{\mu})$.
$Set \;\; Z \leftarrow X$
end for
assemble
$E = vec(E(0), \ldots\ldots, E(N-1))$.
calculate $K = \underset{a \le K \le b}{argmin} \|(I - \Phi K)E\|$
and $\vartheta \leftarrow \vartheta + KE$; $E_0 \leftarrow E$;
end while

4.5. Applied to robot control

With t_s=0.02s and $\tau_d(t) = \begin{pmatrix} \theta_{11} \sin \theta_{12} t \\ \theta_{13} \sin \theta_{14} t \end{pmatrix}$ (60)

Where all $\boldsymbol{\theta_n, n = 1 \div 15, T = 10s}$

$$\boldsymbol{R(t)} = \begin{cases} \boldsymbol{2 * sin(3\pi t/T) + 0.5\, sin(6\pi t/T)} \\ \boldsymbol{4 * sin(3\pi t/T) - 1\, sin(6\pi t/T)} \end{cases}$$

$A_1 = \begin{bmatrix} 50 & 0 \\ 0 & 20 \end{bmatrix}$; $A_2 = \begin{bmatrix} 15 & 0 \\ 0 & 8 \end{bmatrix}$; $K = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.3 \end{bmatrix}$; $t_s = 0.02s$

The simulation results validating the above algorithm are presented in Figures 5 and 6.

These visual tracking results confirm the convergence of both joint variables to their desired references. Throughout the working day, the maximum tracking error reaches approximately 280 with:

$$max|E_{280}(1)| \approx 0.028 \text{and} max|E_{280}(2)| \approx 0.024.$$

Moreover, they confirmed that the tracking error decreases as the number of trials increases, thereby fully validating the aforementioned theoretical claims.

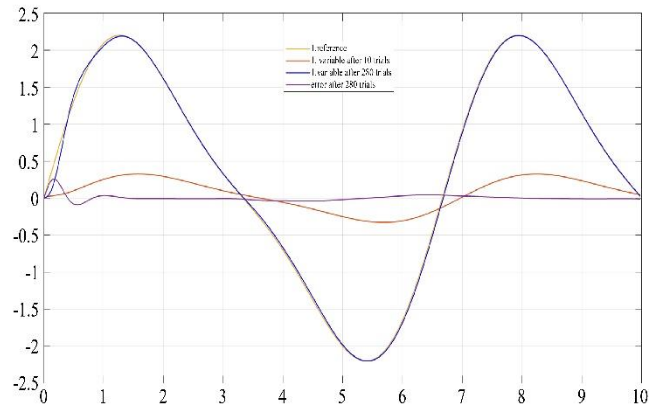


Fig. 5 The output response of the first common variable after trying 10 and 280 times

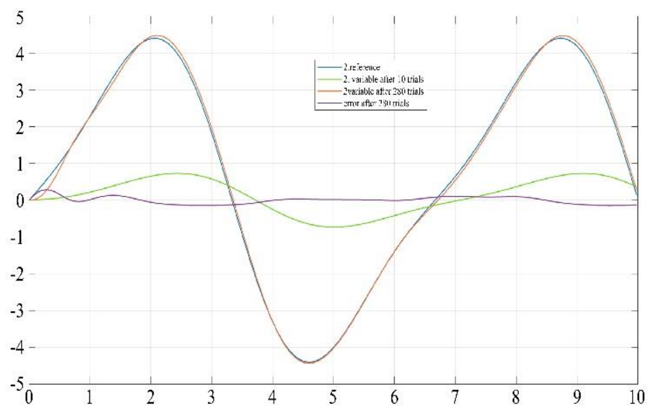


Fig. 6 The output response of the second common variable after trying 10 and 280 times

5. Conclusion

The article proposes two control methods to improve the accuracy of the trajectory for a 2-DOF robot: Sustainable control - adaptation. This is applied to robots with uncertainty parameters and intermingling impact between joints, helping to improve trajectory accuracy. This algorithm is verified as a basis for development for the following joints. Repeat control: Use a two-loop structure, in which the loop in the estimates of the uncertainty component compensates for the robot input by the Taylor chain analysis method, and the outer ring is the repetitive learning controller. This method does not require a math model but only uses data measured from the system. The article focuses on developing conditions for determining the linear function parameter to ensure the convergence process, especially for linear objects with a time model that is not satisfied. In the coming time, you will study the combination of two methods to take advantage of both, helping to improve the accuracy of the trajectory and the system's adaptability. Specifically, sustainable control adaptation will ensure stability and resistance to irregular parameters, while repetitive control will optimize the power based on past data. The study will focus on building an integrated control algorithm, evaluating the actual performance of industrial robotic systems, expanding applications for nonlinear systems, and changing over time.

References

[1] Frank L. Lewis, Darren M. Dawson, and Chaouki T. Abdallah, *Robot Manipulator Control Theory and Practice*, 2nd ed., CRC Press, 2003. [Google Scholar] [Publisher Link]

[2] Halima Medjoubi, Abdessemad Yassine, and Hassam Abdelouahab, "Design and Study of an Adaptive Fuzzy Logic-Based Controller for Wheeled Mobile Robots Implemented in the Leader-Follower Formation Approach," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 6935-6942, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] Boucetta Kasmi, and Abdelouaheb Hassam, "Comparative Study between Fuzzy Logic and Interval Type-2 Fuzzy Logic Controllers for the Trajectory Planning of a Mobile Robot," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 7011-7017, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Wen Yu, and J.A. Heredia, "PD Control of Robot with RBF Networks Compensation," *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, Como, Italy, vol. 5, pp. 329-334, 2000. [CrossRef] [Google Scholar] [Publisher Link]

[5] Lingwei Wu, Qiuzhen Yan, and Jianping Cai, "Neural Network-Based Adaptive Learning Control for Robot Manipulators with Arbitrary Initial Errors," *IEEE Access*, vol. 7, pp. 180194-180204, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] Kevin Gurney, *An Introduction to Neural Networks*, 1st ed., Taylor & Francis Group, CRC Press, 1997. [CrossRef] [Google Scholar] [Publisher Link]

[7] Lorenzo Sciavicco, and Bruno Siciliano, *Modeling and Control of Robot Manipulator*, Springer London, 2000. [CrossRef] [Google Scholar] [Publisher Link]

[8] Bin Yao, "*Nonlinear Adaptive Robust Control*," School of Mechanical Engineering, Purdue University, pp. 1-110, 2002. [Google Scholar] [Publisher Link]

[9] Bin Yao, and Masayoshi Tomizuka, "Adaptive Robust Control of Robot Manipulators: Theory and Comparative Experiments," *Proceedings of the 2nd Chinese World Congress on Intelligent Control and Intelligent Automation*, pp. 1-6, 1997. [Google Scholar]

[10] Douglas A. Bristow, Marina Tharayil, and Andrew G. Alleyne, "A Survey of Iterative Learning Control: A Learning Based Method for High-Performance Tracking Control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96-114, 2006. [Google Scholar]

[11] Mikael Norrlöf, "*Iterative Learning Control: Analysis, Design, and Experiments*," Doctoral Thesis, Linköping University, 2000. [Google Scholar] [Publisher Link]

[12] Duy Hoang et al., "A Model-Free Approach for Output Regulation of Uncertain 4 DOF Serial Robot with Disturbance," *2022 11th International Conference on Control, Automation and Information Sciences (ICCAIS)*, Hanoi, Vietnam, pp. 67-72, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[13] Cao Thanh Trung et al., "Convergence Parameters for D-Type Learning Function," *International Conference on Engineering Research and Applications, ICERA 2020*, Thai Nguyen, Vietnam, vol. 1, pp. 262-269, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Abdelhamid Tayebi, "Adaptive Iterative Learning Control for Robot Manipulators," *Automatica*, vol. 40, no. 7, pp. 1195-1203, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[15] Yoshihiko Miyasato, "Iterative Learning Control of Robotic Manipulators by Hybrid Adaptation Schemes," *42nd IEEE International Conference on Decision and Control*, Maui, HI, USA, vol. 5, pp. 4428-4433, 2003. [CrossRef] [Google Scholar] [Publisher Link]

[16] Richard Lee et al., "Adaptive Iterative Learning Control of Robot Manipulators for Friction Compensation," *IFAC-PapersOnline*, vol. 52, no. 15, pp. 175-180, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[17] Onder Tutsoy, and Duygun Erol Barkana, "Model-Free Adaptive Control of the Under-Actuated Robot Manipulator with the Chaotic Dynamics," *ISA Transactions*, vol. 118, pp. 106-115, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[18] Onder Tutsoy, "Design and Comparison Based Analysis of Adaptive Estimator for Completely Unknown Linear Systems in the Presence of OE Noise and Constant Input Time Delay," *Asian Journal of Control*, vol. 18, no. 3, pp. 1020-1029, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[19] Yu-Hsiu Lee et al., "Industrial Robot Accurate Trajectory Generation by Nested Loop Iterative Learning Control," *Mechatronics*, vol. 74, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] Youqing Wang, Furong Gao, and Francis J. Doyle III, "Survey on Iterative Learning Control, Repetitive Control and Run to Run Control," *Journal of Process Control*, vol. 19, no. 10, pp. 1589-1600, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[21] J.J. Duistermaat, and J.A.C. Kolk, *Distributions: Theory and Applications*, 1st ed., Birkhäuser Boston, MA, 2010. [CrossRef] [Google Scholar] [Publisher Link]
[22] Milton Abramowitz, and Irene A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, 1965. [Google Scholar] [Publisher Link]