

Article

Not peer-reviewed version

Improve the quality of Industrial Robot Control Using an Iterative Learning Method with Online Optimal Learning and Intelligent Online Learning Function Parameters

[Vo Thu Ha](#)*, [Than Thi Thuong](#), [Vo Quang Vinh](#)

Posted Date: 11 January 2024

doi: 10.20944/preprints202401.0851.v1

Keywords: Interactive learning control (ILC); estimated according to the Taylor series; intelligent feedback



Preprints.org is a free multidiscipline 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 is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Improve the Quality of Industrial Robot Control Using an Iterative Learning Method with Online Optimal Learning and Intelligent Online Learning Function Parameters

Vo Thu Ha^{1*}, Than Thi Thuong¹ and Vo Quang Vinh²

¹ Department of Control and Automation, Faculty of Electrical Engineering, University of Economics—Technology for Industries, Hanoi 11622, Vietnam; ttthuong.dien@uneti.edu.vn (T.T.T.).

² Faculty Control and Automation, Electric Power University, Hanoi 11917, Vietnam; vinhvq@epu.edu.vn

* Correspondence: vtha@uneti.edu.vn; Tel.: +84-0913024989

Abstract: It is unavoidable that the characteristics of the robot system alter inexactly or cannot be correctly determined during movement and are impacted by external disturbances. There are many adaptive control methods, such as sliding control or neural control, to improve the quality of trajectory tracking for robot motion systems. Still, those methods require a large amount of computation related to solving the problem of constrained nonlinear optimization problem. The article presents some techniques for determining an intelligent controller's online learning function parameters, including two circuits: the inner circuit is a state feedback linearized controller, and the outer course is an Interactive learning controller and does not use the robot's mathematical mode with function parameters online optimal learning and intelligent online learning closely following the joint model to stabilize the robot system. Finally, the 2-DOF planar robot simulation is performed on Matlab software R2022b. The findings indicate that outstanding tracking performance is achievable.

Keywords: Interactive learning control (ILC); estimated according to the Taylor series; intelligent feedback

1. Introduction

Robots have been widely used in many industrial fields, so countless robot control methods have been researched and applied, helping robots achieve accurate trajectory tracking. However, these methods are all built based on the mathematical model describing the robot, and they are classified depending on the level of accuracy of that mathematical model. The Euler - Lagrange model is a form of mathematical model often used for developing and synthesizing controllers, and it is impacted by external input noise $\underline{d}(t)$ as follows:

$$\underline{\ddot{q}} + \underline{d} = M(\underline{q}, \underline{\theta}) \underline{\ddot{q}} + C(\underline{q}, \underline{\dot{q}}, \underline{\theta}) \underline{\dot{q}} + G(\underline{q}, \underline{\theta}) + F(\underline{\dot{q}}, \underline{\theta}) \quad (1)$$

In which \underline{q} is the vector of joint variables; $\underline{\theta}$ is the vector of uncertain parameters; $M(\underline{q}, \underline{\theta})$ is the inertia matrix that is always positive definite symmetry; $C(\underline{q}, \underline{\dot{q}}, \underline{\theta})$ is the vector of Coriolis and centrifugal components; $F(\underline{\dot{q}}, \underline{\theta})$ describes the effect of friction; $G(\underline{q}, \underline{\theta})$ is the vector gravity; $\underline{d}(t)$ is the noise vector in the actuator and you is the control signal.

With the matching variables equal to the number of input signals, such a robot system has enough actuators. The robot control task is to build a feedback controller so that the output is the joint variables \underline{q} that follow the desired trajectory $\underline{r}(t)$ and the tracking quality does not depend on uncertain parameters and components and external noise $\underline{d}(t)$. Over time, due to many objective factors, such as equipment wear and tear from environmental impacts, the initially set design quality is no longer guaranteed. In this case, it is common to rebuild the controller. As we know, to design

an industrial robot motion control system using the traditional method, one thing that cannot be changed is to always clearly understand the control object, which means the control object must be controlled and expressed in modeling in the form of some mathematical model that accurately describes the object, which here can be a transfer function and can also be a state model in the form of a system of differential equations-highest classification. At the same time, when building and designing a motion control system, it is necessary to anticipate objective factors that will not impact the system as expected, such as external interference, etc., leading to damage to the system. The control is no longer as effective as before, and one must redefine the mathematical model of the control object, re-evaluate the rules of unwanted effects, bring objectivity into the system, Rebuild the controller, or at least redefine the parameters for the controller. There have been many control methods to solve the above problem, for example, the exact linearization method [1]. If the external noise component \underline{q} passes, but the indeterminate constant parameter still exists, we have the inverse control method of the model [2]. In cases where both external noise components and uncertain constant parameters exist, there is a sliding control method [3]. However, the disadvantage of the sliding control method is the phenomenon of chattering when the system slides on the sliding surface at a high frequency. To improve this vibration phenomenon, there is also a high-order sliding control method [4], but it still requires an estimated value of the maximum standard deviation of the model caused by \underline{q} and $\underline{d}(t)$ and cannot eliminate it. Due to vibration, there is still a risk of premature failure of robot mechanical equipment components. To overcome the above disadvantages, intelligent control methods can be used, such as the control trend of not using the robot's dynamic model (1), so the control quality is not affected. by components $\underline{q}(t)$ and $\underline{d}(t)$. The intelligent control method mainly applied to robots mentioned in the article is the iterative learning method [5]. This integrated control method with cyclic working systems requires the same state. At the same time, the parameters $\underline{q}(t)$ and $\underline{d}(t)$ are also required to be periodic and have the exact change period as the working cycle of industrial robots. The primary iterative learning control method is only sometimes applied to meet the desired control quality. There have been many improvements to iterative learning to improve control quality and expand the scope of practical applications. The first improvement is the improvement of combining iterative learning with traditional control methods, often called indirect iterative learning or direct transmission iterative learning. These improvements include [6] when the friction component alone cannot be determined, [7] when that is not possible, or [8] when the learning function parameters cannot be found, as well as when the learning function parameters need to be changed. Change during work cycles. A control trend for a class of robotic systems is model matching control, including [9]. This control method requires the mathematical model (1), which will encounter problems and disadvantages of previous traditional control methods. The next improvement is to use model-based control but almost entirely not use the mathematical model (1) of the robot, applicable to both cases where $\underline{q}(t)$ and $\underline{d}(t)$ are dependent. It depends on time and does not require periodicity, so it can be considered an intelligent model-matching control method. This method is based on theoretical results on optimal control of each segment on the time axis available in [10] and applying disturbance estimation to control self-balancing two-wheeled vehicles [11].

The main content of the article is to analyze and evaluate additional improvements to essential iterative learning to improve the quality of motion control of robot systems and expand the scope of practical applications, specifically, Intelligent control based on model matching (model matching) almost wholly does not need to use the mathematical model of the robot system, applicable to the case where $\underline{q}(t)$ and $\underline{d}(t)$ depends on time, no need for circulation for the robot system. The content of the article includes six parts. Part 1 is the problem of researching iterative learning for robot systems. Part 2 presents the results of the 2-DOF robot dynamic equation; Part 3 presents two robot control structures using iterative learning; Part 4: The first structure controls the robot using iterative learning; Part 5 and part 6: The second structure controls the robot using iterative learning; Part 7 is conclusions and future research directions.

2. Robot planar dynamic model

A 2-DOF planar robot arm is shown in Figure 1, where θ_1, l_1 and m_1 are respectively.

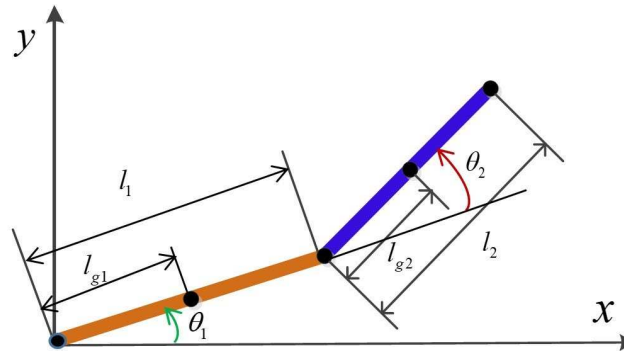


Figure 1. Two -DOF planar robot arm.

The dynamic equations for this system can be found using the Lagrange Euler formulation [1]. The dynamic model of the two-DOF:

$$\underline{u} = M(\underline{q}, \underline{\theta}) \ddot{\underline{q}} + H(\underline{q}, \dot{\underline{q}}, \underline{\theta}) + G(\underline{q}, \underline{\theta}) \quad (2)$$

with the following results:

$$\begin{cases} M_{11} = m_1 l_{c1}^2 + I_1 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos \theta_2) + I_2 \\ M_{12} = m_2 (l_{c2}^2 + l_1 l_{c2} \cos \theta_2) + I_2 \\ M_{21} = m_2 (l_{c2}^2 + l_1 l_{c2} \cos \theta_2) + I_2 \\ M_{22} = m_2 l_{c2}^2 + I_2 \\ H_{11} = -m_2 l_1 l_{c2} \sin \theta_2 (2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) \\ H_{12} = m_2 l_1 l_{c2} (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \sin \theta_2 - m_2 l_1 l_{c2} \sin \theta_2 \dot{\theta}_1 \dot{\theta}_2 \\ G_{11} = m_2 g (l_1 \cos \theta_1 + l_{c2} \cos(\theta_1 + \theta_2)) + m_1 l_{c1} \cos \theta_1 \\ G_{12} = m_2 g l_{c2} \cos(\theta_1 + \theta_2) \end{cases} \quad (3)$$

Problems that arise in working in real-time, factors that cannot be ignored include the impact of interference on the system and unwanted changes to model parameters. The control problem is the signal tracking control for the industrial robot motion system. The most comprehensive solution would be the method of using iterative learning control. This method can be added to a traditional control system to compensate for system errors, or it can also be applied immediately to robots without a conventional controller. The content of the article proposes 02 solutions to control industrial robots by iterative learning using innovative online learning parameters presented explicitly in sections 4, 5, and section 6.

3. Two structure diagrams for robot control using iterative learning method

The content of the article proposes 02 solutions to control industrial robots by iterative learning using innovative online learning parameters Figure 2 and Figure 3.

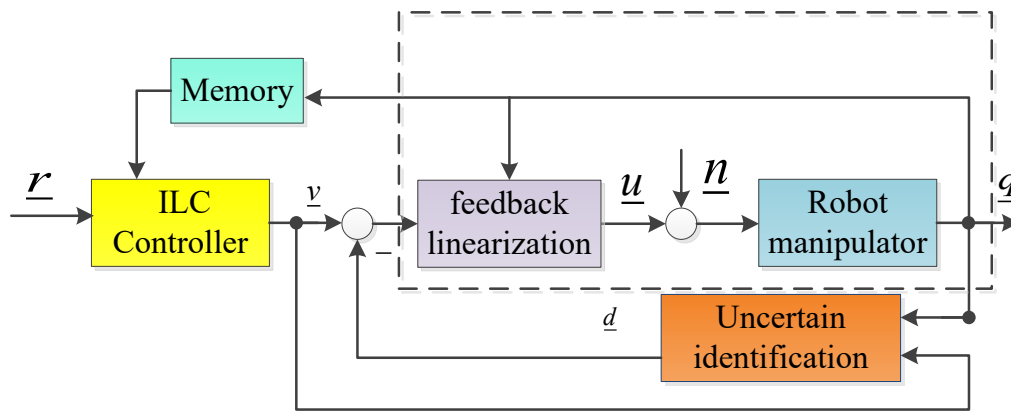


Figure 2. Structure the first diagram for robot control using the iterative learning method.

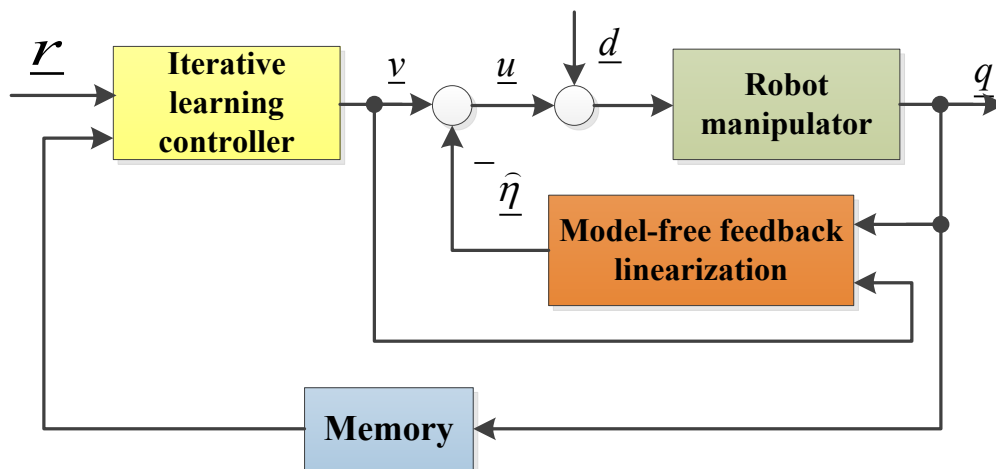


Figure 3. Structure the second diagram for robot control using the iterative learning method.

These two solutions are different because we need to add an intermediate controller in the inner loop with the first solution, but we do not with the second solution. The task of this medium inner-loop controller is to make the industrial robot system adapt to the iterative learning control principle later because the iterative learning controller often requires the control object to be stable (possibly no need for asymptotic stabilization). We need the mathematical model (1) of the industrial robot system to design an intermediate controller as a stable system. In the second solution, the uncertainty estimation block, in addition to the pure task of estimating the function uncertainty components to perform input compensation, must also carry out the additional function of making a stable system. If we apply the "smart" estimation principle introduced in part 5 to design the uncertainty estimation block, then with the second solution, we do not need the industrial robot system's mathematical model (1). However, the price for this is that the processing speed of the second solution will be much slower than the first solution. According to the simulation results obtained, it's about 140 times slower. In addition, since both of these solutions use an uncertainty estimation block for compensation control, they will be applicable for adaptive control of industrial robot systems with additional noise in the input, i.e., applying the method described by (4), (15). This is also a difference from the original model (1) because interference in the actuators is unavoidable. Furthermore, it is possible to assume that this input disturbance can contain matched model uncertainties. In that case, both control solutions are adaptive to the trouble and robustness of model bias.

4. Structure the first diagram for robot control the iterative learning method

4.1. Algorithm content according to the first control diagram

The mathematical model (1) and assume that $C(\underline{q}, \underline{\dot{q}}), G(\underline{q})$ is undetermined. Then we have:

$$M(\underline{q})\ddot{\underline{q}} = \underline{u} + \underline{d} \quad (4)$$

With the sum of all the functional uncertainty components of the model:

$$\underline{d} = \underline{n}(\underline{q}, \underline{\dot{q}}, \underline{\ddot{q}}, t) - C(\underline{q}, \underline{\dot{q}})\underline{\dot{q}} - G(\underline{q}) \quad (5)$$

It can be seen immediately that when used:

$$\underline{u} = M(\underline{q})[\underline{v} - g_1\underline{q} - g_2\underline{\dot{q}}] \quad (6)$$

With g_1, g_2 being two properly chosen matrices, the system becomes:

$$\ddot{\underline{q}} = \underline{v} - g_1\underline{q} - g_2\underline{\dot{q}} + \underline{d} \Leftrightarrow \begin{pmatrix} \underline{\dot{q}} \\ \ddot{\underline{q}} \end{pmatrix} = \begin{pmatrix} 0_n & I_n \\ -g_1 & -g_2 \end{pmatrix} \begin{pmatrix} \underline{q} \\ \underline{\dot{q}} \end{pmatrix} + \begin{pmatrix} 0_n \\ I_n \end{pmatrix} [\underline{v} + \underline{d}] \quad (7)$$

where $\underline{d} = M(\underline{q})^{-1}\underline{d}$ is an uncertain component of system (7).

$$\underline{x} = \begin{pmatrix} \underline{q} \\ \underline{\dot{q}} \end{pmatrix}, G = \begin{pmatrix} 0_n & I_n \\ -g_1 & -g_2 \end{pmatrix}, B = \begin{pmatrix} 0_n \\ I_n \end{pmatrix} \quad (8)$$

The model (7) will have the canonical form:

$$\dot{\underline{x}} = G\underline{x} + B[\underline{v} + \underline{d}] \quad (9)$$

In addition, for system (9) to be stable, that is, to have a Hurwitz matrix \mathbf{A} , we must choose two matrices so that the eigenvalues of \mathbf{A} given in (7) lie to the left of the imaginary axis. The simplest way is to choose them according to the method of assigning pole points using Matlab's *place(.)* command, and these pole points should be selected as accurate (negative) numbers to eliminate fluctuations during transitions.

And then the system output:

$$\underline{y} = \underline{q} = (I_n, 0_n)\underline{x} \quad (10)$$

Follow the given periodic $\underline{r}(t)$ pattern trajectory

Now we use notation: $\left|_{t=iT_s}, \underline{y}_k(i) = \underline{y}_k(t) \right|_{t=iT_s}; \underline{v}_k(i) = \underline{v}_k(t) \left|_{t=iT_s} \right|$

Give the input and output of systems (9) (10), respectively iT_s at the time in the k test, with T_s being the sample extraction cycle. Then the **P-type** learning function, standard to all update time i at the k time, will be:

$$\underline{v}_{k+1} = \underline{v}_k + K\underline{e}_k \quad (11)$$

Where:

$$\begin{aligned} \underline{v}_k &= \text{col}(\underline{v}_k(0), \underline{v}_k(1), \dots, \underline{v}_k(s-1)); \underline{e}_k = \underline{g} - \underline{\omega}_k; \underline{\omega}_k = \text{col}(\underline{r}_k(0), \underline{r}_k(1), \dots, \underline{r}_k(s-1)), \\ \underline{v}_k &= \text{col}(\underline{r}_k(0), \underline{r}_k(1), \dots, \underline{r}_k(s-1)). \end{aligned} \quad (12)$$

$$K = \begin{pmatrix} K & 0_{n \times n} \\ \vdots & \vdots \\ 0_{n \times n} & K \end{pmatrix} \in \mathbb{R}^{nS \times nS} \text{ and } K \in \mathbb{R}^{n \times n}$$

K is the matrix of learning functions that must be chosen so that the learning function converges.

The uncertainty component of the system, in general, can be estimated and explicitly compensated as follows: First, we must approximate model (9) at time i in the k time, then when paid by $\hat{\underline{d}}_k(i-1)$, becomes:

$$\frac{\underline{x}_k(i) - \underline{x}_k(i-1)}{T_s} \approx G\underline{x}_k(i) + B[\underline{v}_k(i) - \hat{\underline{d}}_k(i) + \underline{d}_k(i)]$$

Hence, $G\underline{x}_k(i) \approx \underline{x}_k(i-1) + B[\underline{v}_k(i) - \hat{\underline{d}}_k(i) + \underline{d}_k(i)] \Leftrightarrow \hat{B}\underline{d}_k(i) \approx \underline{h}_k(i)$

In which: $\hat{G} = (I_{2n} - T_s G)$, $\hat{B} = T_s B$, $\underline{h}_k(i) = \hat{G}\underline{x}_k(i) - \underline{x}_k(i-1) - B[\underline{v}_k(i) - \hat{\underline{d}}_k(i-1)]$

So from $\hat{B}^T \hat{B} = T_s^2 I_n$, we have the estimated value $\hat{\underline{d}}_k(i) \approx \underline{d}_k(i)$ for the next offset at time $i+1$ as:

$$\begin{aligned} \underline{d}_k(i) &= (\hat{B}^T \hat{B})^{-1} \hat{B}^T \underline{h}_k(i) = \frac{1}{T_s^2} \hat{B}^T \underline{h}_k(i) = \frac{1}{T_s} B^T \underline{h}_k(i) \\ &= \frac{1}{T_s} (0_n, I_n) \left\{ \hat{G}\underline{x}_k(i) - \underline{x}_k(i-1) - B[\underline{v}_k(i) - \underline{d}_k(i-1)] \right\} \\ &= \frac{1}{T_s} B^T \left\{ [\hat{G}\underline{x}_k(i) - \underline{x}_k(i-1)] - \underline{v}_k(i) + \underline{d}_k(i-1) \right\} \end{aligned} \quad (13)$$

The structure of the intelligent control system for industrial robots is shown in Figure 2.

Below is the algorithm describing the steps to calculate an iterative learning controller, combined with identifying and compensating uncertain components (no need repetitive) for industrial robots after accurate linearization; several tries k is continuous along with the control process and only ends when the control stops.

Algorithm 1: structure the first for robot control the iterative learning method is with function parameters smart online learning

```

Assign  $t = 0, \hat{\underline{d}} = 0$ . Choose  $T_s > 0$  ;
1      Calculate  $\hat{G}, S$  . Choose  $K$ ;
      Assign small value  $\underline{x}_0 = 0, \hat{\underline{d}} = 0, \underline{v} = \text{col}(\underline{v}(0), \dots, \underline{v}(s-1)), \underline{r} = \text{col}(\underline{r}(0), \dots, \underline{r}(s-1))$  .
2      while continue the control do
3          for  $i = 0, 1, \dots, M-1$  do
4              Send  $\underline{u} = \underline{v}(i) - \hat{\underline{d}}$  to into uncertain control (9) and determine  $\underline{x} = \underline{x}(t + iT_s)$  .
5              Calculate  $\hat{\underline{d}} \leftarrow B^T (\hat{G}\underline{x} - \underline{x}_0) / T_s - \underline{v}_h(i) + \hat{\underline{d}}$  .
6              Establish  $\underline{y}(i) = (I_n, 0_n)\underline{x}$  and calculate  $\underline{e}(i) = \underline{r}(i) - \underline{y}(i)$  .
7              end for
8              Set up the sum vector.  $\underline{\varepsilon} \leftarrow \underline{e}(i)$ ,  $i=0, 1, \dots, S-1$  theo (11)
9      Update vector  $\underline{v}$  from its existing value according to (10) that is, calculate the values.  $\underline{v}(i)$ ,  $i = 0, 1, \dots, S-1$ 
10     Set  $t \leftarrow t + T$ 
11     end while

```

4.2. Applied to robot control

For the simulation, there are assigned:

$$G_1 = \begin{bmatrix} 30 & 0 \\ 0 & 12 \end{bmatrix}, G_2 = \begin{bmatrix} 11 & 0 \\ 0 & 7 \end{bmatrix}, K = \begin{bmatrix} 2.9 & 0 \\ 0 & 1.8 \end{bmatrix}, T_s = 0.05s, T = 20s$$

The orbits are set as two periodic functions of period T as follows:

$$r_1(t) = \begin{cases} \min \{T_s t / 2, 1.5\} & \text{when } t \leq T/2 \\ r_2(T-t) & \text{when } T/2 < t \leq T \end{cases}; r_2(t) = \sin(2\pi t/T)$$

We obtain the simulation results in Figure 4, Figure 5, Figure 6 and Figure 7.

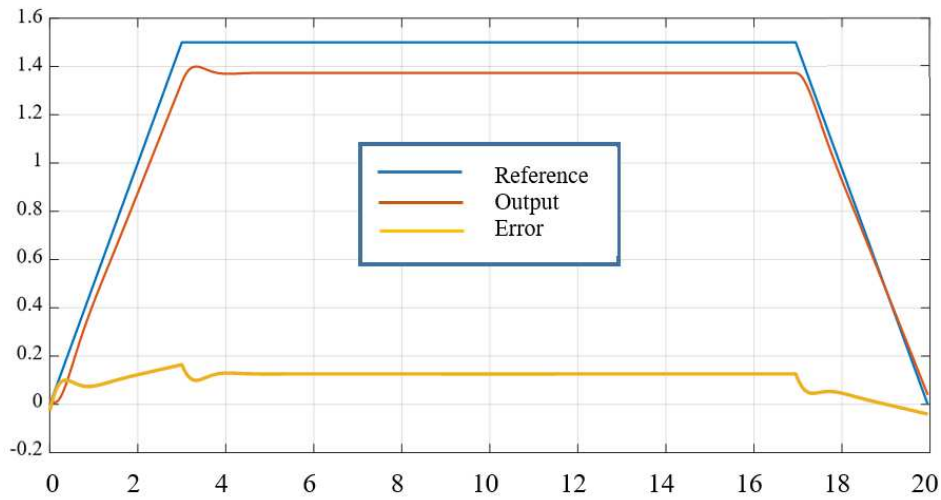


Figure 4. Output tracking results after 20 trials of the first joint.

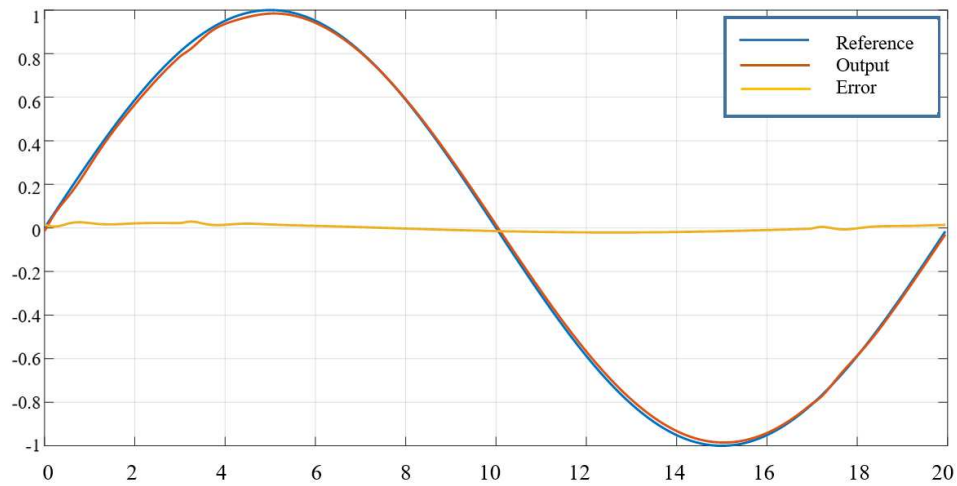


Figure 5. Output tracking results after 20 trials of the second joint.

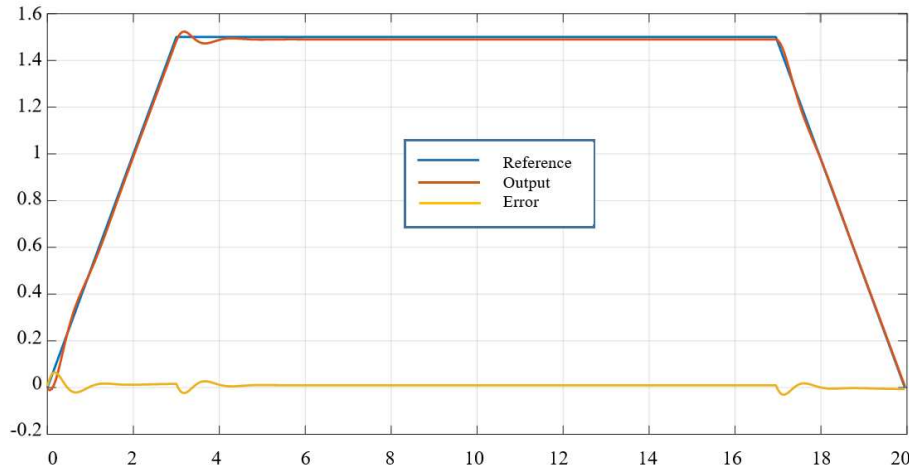


Figure 6. Output tracking results after 60 trials of the first joint.

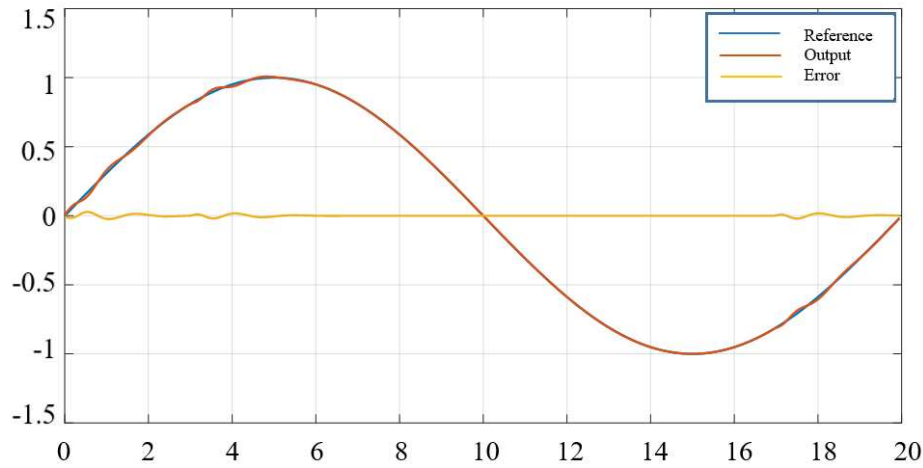


Figure 7. Output tracking results after 60 trials of the second joint.

The simulation results in Figure 4 and Figure 5 show that, when combined with the identification and noise compensation stages of the traditional controller (13), the iterative learning controller (11) has achieved good signal tracking quality. The preset error with tracking error after 60 trials is small enough and acceptable: $-0.03 \leq e_1(t) \leq 0.065$ and $-0.017 \leq e_2(t) \leq 0.02$. The residual error $e_1(t)$ in the simulation results is due to the non-smooth trapezoidal signal set $r_1(t)$ at $t_0 = 0; t_1 = 1.5T_s; t_2 = T - 1.5T_s; t_3 = T$. It also affects the tracking error $e_2(t)$ due to channel interleaving in the system.

5. The structure of the second iterative learning controller with model-free determination of optimal learning parameters for industrial robot

To stabilize the system without using traditional control methods. This makes it possible to apply iterative learning control to even unstable systems without using a mathematical model of the system.

Tool: Estimate the derivative using Taylor series analysis.

5.1. Algorithm content according to the first control diagram

$$\ddot{\underline{q}} = -A_1 \underline{q} - A_2 \dot{\underline{q}} + \underline{u} + \underline{\eta} \quad (14)$$

where A_1, A_2 are two arbitrarily chosen matrices and

$$\underline{\eta} = \underline{d} + \left[I_n - M(\underline{q}, \underline{\theta}) \right] \ddot{\underline{q}} - \left[C(\underline{q}, \dot{\underline{q}}, \underline{\theta}) - A_2 \right] \dot{\underline{q}} - \left[G(\underline{q}, \underline{\theta}) - A_1 \underline{q} \right] \quad (15)$$

From (15) we see that the route cost calculation of model (1) is contained in this vector $\underline{\eta}$. Based on the equivalent model (14), the proposed intelligent controller for the robot is as follows:

First, the estimation errors of the new unknown function vector $\underline{\eta}$ will be adjusted, as shown below.

$$-\hat{\underline{\eta}} + \underline{v} = \underline{u} \quad (16)$$

With the estimation of bias $\underline{\delta}$ as (17) system (14) will become linear:

$$\dot{\underline{x}} = A\underline{x} + B[\underline{v} + \underline{\delta}], \underline{y} = \underline{q} = (I_n, 0_n) \underline{x} \quad (17)$$

$$\text{With } \dot{\underline{x}} = \begin{pmatrix} \underline{q} \\ \dot{\underline{q}} \end{pmatrix}, A = \begin{pmatrix} 0_n & I_n \\ -A_1 & -A_2 \end{pmatrix}, B = \begin{pmatrix} 0_n \\ I_n \end{pmatrix}, \underline{\delta} = \underline{\eta} - \hat{\underline{\eta}} \quad (18)$$

and $0_n, I_n$ are $n \times n$ square matrices with zero

Secondly, control the compensation system in equation (17) so that its output $\underline{y}(t)$ will converge to a desired reference value $\underline{x}(t)$.

5.2. Feedback linearization for internal loop control using model-free disturbance compensation

Returning to the compensator \underline{v}_k in equation (16) and designating as the present working period i.e. with $t = kT + \tau, 0 \leq \tau < T$ and $\underline{v}_k = \underline{v}_k(\tau), \underline{x}_k = \underline{x}_k(\tau)$, then (17) is rewritten as:

$$\dot{\underline{x}}_k(\tau) = A\underline{x}_k(\tau) + B[\underline{v}_k(\tau) - \hat{\underline{\eta}}_k(\tau) + \underline{\eta}_k(\tau)] \quad (19)$$

The automatic mode – free feedback linearization block (Figure 2) aims to estimate $\hat{\underline{\eta}}$ from the measured values of $\underline{x}_k(\tau)$ at two times $\tau = iT_s$ and $\tau - T_s = (i-1)T_s$ for each corresponding $\underline{x}_k(iT_s), \underline{x}_k((i-1)T_s)$ value. With $0 < T_s \ll 1$ being an arbitrarily chosen constant small value, then we expand the Taylor function of the function $\underline{x}_k(\tau)$ around $(i-1)T_s$ as follows,

$$\underline{x}_k((i-1)T_s) = \underline{x}_k(iT_s) - T_s \dot{\underline{x}}_k(iT_s) + \frac{T_s^2}{2} \ddot{\underline{x}}_k(\varsigma) \quad (20)$$

Where $(i-1)T_s \leq \varsigma \leq iT_s$, or

$$\dot{\underline{x}}_k(iT_s) \approx \frac{\underline{x}_k(iT_s) - \underline{x}_k((i-1)T_s)}{T_s} \quad (21)$$

Assuming the last term of expression (20) is very small and can be ignored, the expression $\hat{\underline{\eta}}_k(iT_s)$ will be used to approximate equation (19) by replacing the signs " \approx " and $\underline{\eta}_k(iT_s)$ in (10) with "=" and $\hat{\underline{\eta}}_k(iT_s)$

$$\frac{\underline{x}_k(iT_s) - \underline{x}_k((i-1)T_s)}{T_s} \approx A\underline{x}_k(iT_s) + B[\underline{v}_k(iT_s) - \hat{\underline{\eta}}_k((i-1)T_s) + \underline{\eta}_k(iT_s)] \quad (22)$$

Then, calculate

$$B\hat{\underline{\eta}}_k(iT_s) = \frac{\underline{x}_k(iT_s) - \underline{x}_k((i-1)T_s)}{T_s} - A\underline{x}_k(iT_s) - B[\underline{v}_k(iT_s) - \hat{\underline{\eta}}_k((i-1)T_s)]$$

which yields

$$\hat{\eta}_k(iT_s) = B^T \left[\frac{\underline{x}_k(iT_s) - \underline{x}_k((i-1)T_s)}{T_s} - A\underline{x}_k(iT_s) \right] - \left[\underline{v}_k(iT_s) - \hat{\eta}_k((i-1)T_s) \right] \quad (23)$$

Theorem 1: The value $\hat{\eta}_k(iT_s)$ calculated in equation (23) causes the approximation error of expression (22) to decrease.

Proof: Error in expression (22) $\underline{\varepsilon}$ will be calculated:

$$\begin{aligned} \underline{\varepsilon} &= A\underline{x}_k(iT_s) + B \left[\underline{v}_k(iT_s) - \hat{\eta}_k((i-1)T_s) + \underline{\eta}_k(iT_s) \right] - \frac{\underline{x}_k(iT_s) - \underline{x}_k((i-1)T_s)}{T_s} \\ &= B\underline{\eta}_k(iT_s) + \underline{\lambda} \end{aligned}$$

where

$$\underline{\lambda} = A\underline{x}_k(iT_s) + B \left[\underline{v}_k(iT_s) - \hat{\eta}_k((i-1)T_s) \right] - \frac{\underline{x}_k(iT_s) - \underline{x}_k((i-1)T_s)}{T_s}$$

And the estimated value will also be calculated according to expression (22) as follows:

$$\frac{\underline{x}_k(iT_s) - \underline{x}_k((i-1)T_s)}{T_s} = A\underline{x}_k(iT_s) + B \left[\underline{v}_k(iT_s) - \hat{\eta}_k((i-1)T_s) + \underline{\eta}_k(iT_s) \right]$$

The optimization problem would then be written:

$$\begin{aligned} \eta^* &= \arg \min_{\underline{\eta}_k} \|\underline{\varepsilon}_k\|^2 = \arg \min_{\underline{\eta}_k} \|B\underline{\eta}_k + \underline{\lambda}\|^2 = \arg \min_{\underline{\eta}_k} \|B\underline{\eta}_k + \underline{\lambda}\|^T \|B\underline{\eta}_k + \underline{\lambda}\| \\ &= \arg \min_{\underline{\eta}_k} \left\| \underline{\eta}_k^T \underline{\eta}_k + \underline{\lambda}^T B\underline{\eta}_k + \underline{\lambda}^T \underline{\lambda} \right\|^T \end{aligned}$$

has a unique solution

$$\eta^* = -B^T \underline{\lambda}$$

which coincides with $\hat{\eta}_k(iT_s)$ given in (23).

From (23) we see that the estimation tool created to compensate for noise and uncertainty in the vector model $\underline{\eta}$ does not use the mathematical model (1). Therefore, the noise compensator (16) with $\hat{\eta}$ obtained from expression (23) will be model-free.

5.3. Outer loop control is by iterative learning controller design.

Expression (17) describes the system (14) that becomes LTI by compensating the $\underline{\eta}$ synthetic disturbances in (15) by the compensator (16), and it is rewritten in ILC language for iterative learning systems as after:

$$\left\{ \begin{aligned} \underline{x}_k(i+1) &= \hat{A} \underline{x}_k(i) + \hat{B} [\underline{v}_k(i) + \underline{\delta}_k(i)] \\ \underline{y}_k(i) &= \hat{C} \underline{x}_k(i) \end{aligned} \right\} \quad (24)$$

where $i = 0, 1, \dots, N = T / T_s$, $\underline{x}_k(N) = \underline{x}_{k+1}(0)$, and

$$\hat{A} = \exp(AT_s), \hat{B} = \int_0^{T_s} e^{At} B dt \text{ and } \hat{C} = (I_n, O_n) \quad (25)$$

The control objective at this point is to choose the proper learning parameter K for the update rule P-Type, assuming we have identified two matrices that make the matrix A provided in equation (6).

$$\underline{v}_{k+1}(i) = \underline{v}_k(i) + K \underline{e}_k, \text{ with } \underline{e}_k(i) = \underline{r}_k(i) + \underline{y}_k(i) \quad (26)$$

In order to satisfy the required convergence $\|e_k(i)\| \rightarrow 0$ for all i , or at least as close as possible to the origin.

The value v_k given by ILC (26) is partially constant, so the discrete time model in equation (24) obtained is equivalent to the continuous time model in equation (17). In expression (24), no information about the system model (1) is used either. Therefore (24) can be applied to all robot controllers. From (24), it is obtained with the assumption $\delta_k(i) = 0$

$$y_{k+1}(i) = C\hat{A}^i x_{k+1}(0) + \sum_{j=0}^{i-1} C\hat{A}^{i-1-j} \hat{B}v_{k+1}(j),$$

This results from (24) and the clear demonstration of repeatable ability $x_k(0) = x_{k+1}(0), \forall k$

$$\begin{aligned} e_{k+1}(i) &= r(i) - y_{k+1}(i) = r(i) - \left[C\hat{A}^i x_{k+1}(0) + \sum_{j=0}^{i-1} C\hat{A}^{i-1-j} + (v_k(j) + K e_k(j)) \right] \\ &= r(i) - y_k(i) - \sum_{j=0}^{i-1} C\hat{A}^{i-1-j} \hat{B}K e_k(j) = (I - C\hat{B}K) e_k(i) - \sum_{j=0}^{i-2} C\hat{A}^{i-1-j} \hat{B}K e_k(j) \end{aligned}$$

Hence,

$$e_{k+1} = \Phi e_k \quad (27)$$

Where

$$e_k = \begin{pmatrix} e_k(0) \\ e_k(1) \\ \vdots \\ e_k(N-1) \end{pmatrix} \text{ and } \Phi = \begin{pmatrix} I - C\hat{B}K & 0 & \cdots & 0 \\ -C\hat{A}\hat{B}K & I - C\hat{B}K & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C\hat{A}^{N-2}\hat{B}K & -C\hat{A}^{N-2}\hat{B}K & \cdots & I - C\hat{B}K \end{pmatrix} \quad (28)$$

Theorem 2: For the scenario $\delta_k(i) = 0$, all requirements $\|e_k(j)\| \rightarrow 0$ for all $i = 0, 1, \dots, N-1$ will only be satisfied if and when the P-Type learning parameter K is selected in a way that makes the value supplied in (16) Schur.

Proof: For the autonomous system (27) the proof is comprehensive as it is clear that it is stable if and only if it is Schur.

5.4. Control algorithm and performance of closed – loop system

The following algorithm is established to implement the proposed model-free controller (16) where the v and η values are obtained from expressions (26) and (13), respectively. In this control algorithm, the working time T of the robots is repeatedly tested by each While loop.

In fig. 2 illustrates the output tracking performance of the closed-loop system with the output scenario $\delta_k(i) \neq 0$.

Theorem 3: The proposed mode-free control framework in Fig. 1 (which includes the feedback linearization block via disturbance compensator (16), (23) and the ILC block (26), drives the robot manipulators' (1) output tracking error $|e_k(\tau)|$ to a T_s dependent neighborhood O of origin if d is limited and continuous. If a

smaller T_s is selected, a smaller O will be.

Proof: Because noise \underline{d} is continuous and bounded, the total noise $\underline{\eta}$ and component $\underline{\delta}$ are also continuous and bounded. The upper limit of O is Δ and this limit can be reduced arbitrarily by reducing T_s according to theorem 1

The validity of Theorem 2 was confirmed by

$$\dot{\underline{w}} = A\underline{w} + B\underline{v} \text{ with } \underline{w} = \text{vec}(\underline{r}, \dot{\underline{r}}) \quad (29)$$

The subtraction (29) from (17) yields

$$\dot{\underline{\varepsilon}} = A\underline{\varepsilon} - B\underline{\delta} \text{ where } \underline{\varepsilon} = \text{vec}(\underline{e}, \dot{\underline{e}}) \quad (30)$$

Lyapunov equation $A^T P + PA = -Q$ with optional positive definite matrix Q we always have positive definite P because A is Hurwitz. From the positive definite function $V(\underline{\varepsilon}) = \underline{\varepsilon}^T P \underline{\varepsilon}$ we have the result:

$$\begin{aligned} \dot{V} &= (A\underline{\varepsilon} - B\underline{\delta})^T + \underline{\varepsilon}^T P (A\underline{\varepsilon} - B\underline{\delta}) \\ &= \underline{\varepsilon}^T (A^T P + PA) \underline{\varepsilon} - 2\underline{\varepsilon}^T PB\underline{\delta} = -\underline{\varepsilon}^T Q \underline{\varepsilon} - 2\underline{\varepsilon}^T PB\underline{\delta} \\ &= -\lambda_{\max}(Q) \|\underline{\varepsilon}\|^2 + 2\|PB\|\Delta \|\underline{\varepsilon}\| = \|\underline{\varepsilon}\| [-\lambda_{\max}(Q) \|\underline{\varepsilon}\| + 2\|PB\|\Delta] \end{aligned} \quad (31)$$

with $\lambda_{\max}(Q)$ is the maximum value of the Q matrix.

Since $\dot{V}(\underline{\varepsilon}) < 0$ as long as

$$2\|PB\|\Delta < \lambda_{\max}(Q) \|\underline{\varepsilon}\| \text{ or } \frac{2\|PB\|\Delta}{\lambda_{\max}(Q)} < \|\underline{\varepsilon}\| \quad (32)$$

The orbital error vector tends to the origin

$$O = \left\{ \underline{\varepsilon} \in R^{2n} \left| \|\underline{\varepsilon}\| \leq \frac{2\|PB\|\Delta}{\lambda_{\max}(Q)} \right. \right\} \quad (33)$$

This control algorithm contains all the necessary calculations and has the following structure:

Algorithm 2: The structure of the second iterative learning controller with model-free determination of optimal learning parameters for industrial robot

Choose two matrices A_1, A_2 , given in (25) become Hurwitz.

Determine $\hat{A}, \hat{B}, \hat{C}$ given in (25) and Φ given in (28).

Choose $0 < T_s \ll 1$. Calculate $S = T / T_s$.

1 Determine $D_1^T = 1/T_s; D_0^T = -D_1^T$ given in (23)

Choose learning $\hat{\eta}$ and tracking error e_0 .

Assign $\underline{v}(i) = \underline{r}(i)$, With $i = 0, 1, \dots, S-1$ and $\underline{z} = \underline{0}$. Choose learning parameter K so that Φ of (16) becomes Schur.

2 **while** continue the control **do**

3 **for** $i = 0, 1, \dots, S-1$ **do**

4 Send $\underline{u} = A_1 \underline{q} - A_2 \dot{\underline{q}} - \hat{\eta}$ to robot for a while of T_s .

measure $\underline{x} = \text{vec}(\underline{e}, \dot{\underline{e}})$, $\underline{y}(i) = \underline{q}$.

5 calculate $\hat{\eta}_k \leftarrow B^T \left[\frac{\underline{x} - \underline{z}}{T_s} - A \underline{x} \right] - (\underline{v}(i) - \hat{\eta})$. Set $\underline{z} \leftarrow \underline{x}$.

```

6   end for
7   establish  $\underline{v} = \text{vec}(\underline{v}(0), \dots, \underline{v}(N-1)), \underline{e} = \text{vec}(\underline{e}(0), \dots, \underline{e}(N-1))$ .
8   calculate  $K = \arg \min_{a \leq K \leq b} \|(I - \Phi K)e\|$  and  $\underline{v} \leftarrow \underline{v} + K\underline{e}; \underline{e}_0 \leftarrow e;$ 
9   end while

```

Thus, each while - do loop represents one working cycle (one attempt) in this algorithm. In addition, it can be seen that the above control algorithm does not use model (1) of the robot at all, so it is also an intelligent algorithm.

5.5. Applied to robot control

The robot is repetitive with $T = 0.02s$ and disturbed by:

$$\underline{d}(t) = \begin{pmatrix} \theta_{11} \sin \theta_{12} t \\ \theta_{13} \sin \theta_{14} t \end{pmatrix} \quad (34)$$

The parameters $\theta_i, i = 1 \div 14$ are random. The robot is assumed to have an period cycle: $T = 10s$; The control signal update time is selected: $T_s = 0.02s$. A robot with 2 degrees of freedom has the following parameters (Fig 1): $l_1 = 0.4, l_2 = 0.6, g = 9.81$

Assigned:

$$r_1(t) = \begin{cases} \min\{T_s t / 2, 1.5\} & \text{when } t \leq T/2 \\ r_2(T-t) & \text{when } T/2 < t \leq T \end{cases}; r_2(t) = \sin(2\pi t / T)$$

$$A_1 = \begin{bmatrix} 31 & 0 \\ 0 & 12 \end{bmatrix}, A_2 = \begin{bmatrix} 12 & 0 \\ 0 & 8 \end{bmatrix}, K = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.3 \end{bmatrix}, T_s = 0.02$$

Simulation results to verify the above algorithm are shown in Figure 8 and Figure 9.

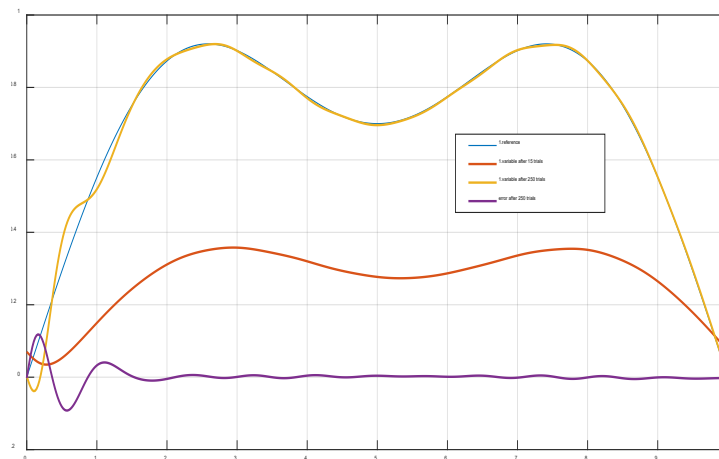


Figure 8. The output response of the first common variable after trying 15 and 250 times.

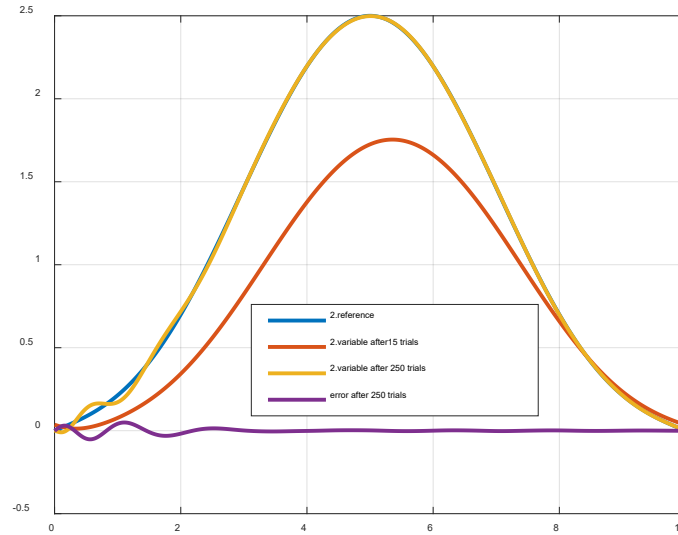


Figure 9. The output response of the second common variable after trying 15 and 250 times.

These visual tracking results demonstrate the convergence of both joint variables to the required references. Over the course of the working day, the highest tracking error value is around $\max |e_{250}(1)| \approx 0.03$ for the first joint variable and $\max |e_{250}(2)| \approx 0.02$ for the second joint variable, respectively. Furthermore, they confirmed that the tracking error will decrease with the number of trials conducted. Therefore, these simulation findings fully validate all of the above-mentioned theoretical claims.

6. The structure of the second iterative learning controller with model-free determination of online learning parameters for industrial robot

The structure of two control loop circuits is still used, as shown in Figure 3.

6.1. Control the inner loop

Next is the inner loop. This loop uses Taylor series analysis to estimate $\hat{\eta}$ at time $\tau = kT + iT_s$, the previously measured value of $q_k(j), \dot{q}_k(j)$, with $j = i; i-1$. This estimation formula is given in (35).

The remarkable thing is that while the iterative learning controller works in a cyclic cycle T , the inner loop controller is not cyclic. In other words, the internal loop controller also estimates the non-periodic functional uncertainty component. It can be seen that neither control loop uses the mathematical model (1) of the robot at all, and that is their intelligent feature.

6.2. Outer loop control is by iterative learning controller design.

This is a loop circuit that uses cybernetics **P-style** iteration with diagonal learning function parameter matrix

$$K_k = \text{diag} \left(K_k^{(j)} \right), j = 1, 2, \dots, n \quad (35)$$

The number of elements equals the number of joint variables $q \in R^n$ (output) and the input variable u because robot (1) is a sufficient actuator. Each $K_k^{(j)}$ element is determined according to:

$$K = \underline{a}_k \cdot \arg \min_{0 \leq z \leq l} \|z \underline{e}_k (1-z) \underline{e}_{k-l}\| \quad \text{or}$$

$$K = \underline{a}_k \cdot \arg \min_{0 \leq z \leq l} \|\underline{e}_k (1-z) + z \underline{e}_{k-l}\|.$$

This loop generates the control signal.

$$\underline{u} = -A_1 \underline{q} - A_2 \dot{\underline{q}} - \hat{\underline{\eta}} \quad (36)$$

where choose two matrices A_1, A_2 in (6), which become Hurwitz and a sufficiently small constant $0 < T_s \ll l$, $\underline{v} \in R^n$ is obtained from the iterative learning controller $\underline{v}(t) = \underline{v}_k(\tau)$ piecewise constant form, i.e.

$$\underline{v}_k(\tau) = \underline{v}_k(i) \quad \text{when} \quad iT < \tau \ll (i+1)T_s \quad \text{with} \quad \tau \ll iT_s, i = 0, 1, \dots, N-1$$

$$\text{and} \quad \underline{v}_{k+1}(\tau) = \underline{v}_k + K_k \underline{e}_k$$

$$\text{with} \quad \underline{v} = \underline{e} = \text{vec}(\underline{e}(0), \dots, \underline{e}(N-1)) \text{ and } \text{vec}(\underline{v}(0), \dots, \underline{v}(N-1))$$

The remaining component \underline{m} in (3.39) is the estimated value of \underline{m} given in (3.18), including input uncertainty \underline{d} and model bias, i.e

$$\underline{\eta} = \underline{d} + [I_n - M(\underline{q}, \underline{\theta})] \ddot{\underline{q}} - C(\underline{q}, \dot{\underline{q}}, \underline{\theta}) \dot{\underline{q}} - G(\underline{q}, \underline{\theta}) - F(\underline{\theta}) \dot{\underline{q}} \quad (37)$$

at time $\tau = kT + iT_s$. The inner loop controller determines it.

To facilitate installation, the following algorithm summarizes the calculations for implementing the controller (16) proposed above.

Algorithm 3: the structure of the second iterative learning controller with model-free determination of online learning parameters for industrial robot

Choose two matrices A_1, A_2 , in (6), which become Hurwitz and a sufficiently small constant $0 < T_s \ll l$.

Calculate $N = T / T_s$.

Determine $D_1^T = 1/T_s$; $D_0^T = -D_1^T$;

Choose learning $\hat{\underline{\eta}}$ and tracking error \underline{e}_0 .

Assign the robot's initial state and initial output to the outer loop controller (iterative learning controller)

$$\underline{v}(i) = \underline{r}(i), i = 0, 1, \dots, S-1 \text{ and } \underline{z} = \underline{0}$$

while keeping the controls in place

for $i = 0, 1, \dots, S-1$ do

send $\underline{u} = A_1 \underline{q} - A_2 \dot{\underline{q}} - \hat{\underline{\eta}}$ to robot for a while of T_s .

measure $\underline{y}(i) = \underline{q}$ and $\underline{x} = \text{vec}(\underline{e}, \dot{\underline{e}})$.

determine $\underline{e}_k(i) = \underline{r}(i) - \underline{y}(i)$.

Calculate $\hat{\underline{\eta}}_k \leftarrow (\Theta, I) \left[\underline{z} D_0^T + \underline{x} D_1^T - A \underline{x} \right] - (\underline{v}(i) - \hat{\underline{\eta}})$. Set $\underline{z} \leftarrow \underline{x}$.

end for

establish $\underline{v} = \text{vec}(\underline{v}(0), \dots, \underline{v}(S-1))$ and $\underline{e} = \text{vec}(\underline{e}(0), \dots, \underline{e}(S-1))$.

Calculate $K = \underline{a}_k \cdot \arg \min_{0 \leq z \leq l} \|z \underline{e}_k (1-z) \underline{e}_{k-l}\|$ or $K = \underline{a}_k \cdot \arg \min_{0 \leq z \leq l} \|\underline{e}_k (1-z) + z \underline{e}_{k-l}\|$

and $\underline{v} \leftarrow \underline{v} + K \underline{e}$; $\underline{e}_0 \leftarrow \underline{e}$;

end while

6.3. Applied to robot control

Evaluate and verify the quality of the control algorithm for a two-DOF planar robot with the following data:

Friction components:

$$F(\underline{q}, \underline{\dot{q}}, \underline{\theta})\underline{\dot{q}} = \begin{pmatrix} \theta_{11}f_1 \\ \theta_{21}f_2 \end{pmatrix}; f_1 = \theta_{11}\dot{q}_1 + \theta_{12}q_1 \tanh \dot{q}_1; f_2 = \theta_{21}\dot{q}_2 + \theta_{22}q_2 \tanh \dot{q}_2$$

With input noise assumed as follows:

$$\underline{d}(t) = \begin{pmatrix} \theta_{31} \sin(2.3\pi t) \\ \theta_{32} \sin(0.85\pi t) \end{pmatrix} \quad (38)$$

Instead of using the controller design, the parameters (20) and disturbances (21) will be employed to simulate the robot dynamics as in model (1). The parameter $\theta_i, i = 1 \div 14$ is chosen randomly.

A robot with 2 degrees of freedom has the following parameters:

$$l_1 = 0.4; l_2 = 0.6; g = 9.81$$

The robot is assumed to have a duty cycle $T = 10s$. The time to update the control signal (and also to measure the output) is chosen as $T = 0.02s$. The signal given to the two channels is two periodic functions of period T as follows:

$$r_1(t) = r_2(T-t) \text{ khi } \frac{T}{2}; r_2(t) = \sin(2\pi t/T);$$

$$A_1 = \begin{bmatrix} 31 & 0 \\ 0 & 12 \end{bmatrix}, A_2 = \begin{bmatrix} 12 & 0 \\ 0 & 8 \end{bmatrix}$$

We obtain the simulation results in Figures 10, 11, 12, 13, 14, 15, 16 and Fig 17.

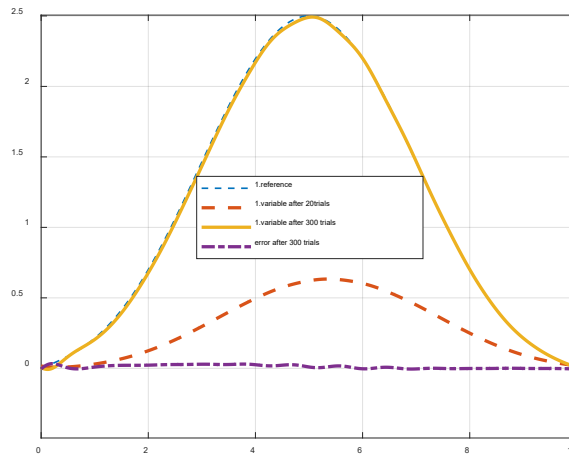


Figure 10. Meets the position of the first joint when used (39).

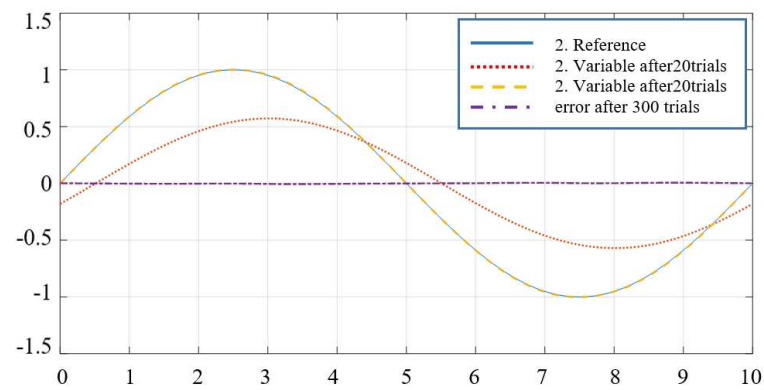


Figure 11. Meets the position of the second joint when used (39).

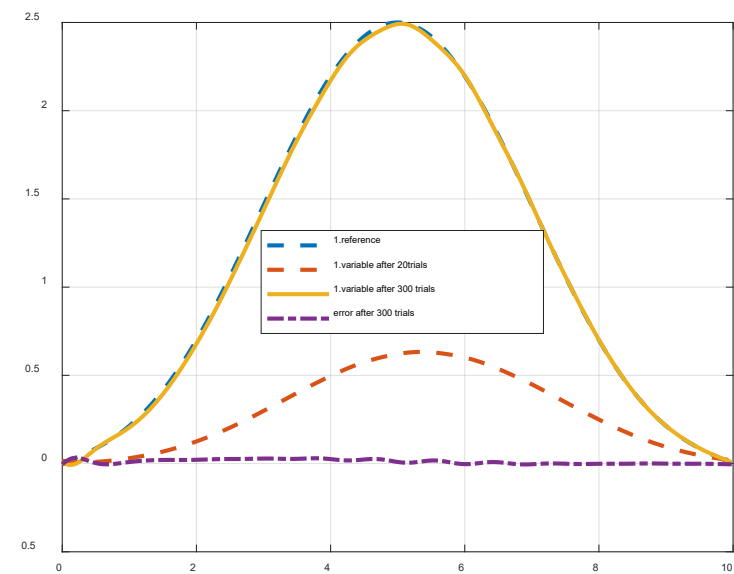


Figure 12. Meets the position of the first joint when used (40).

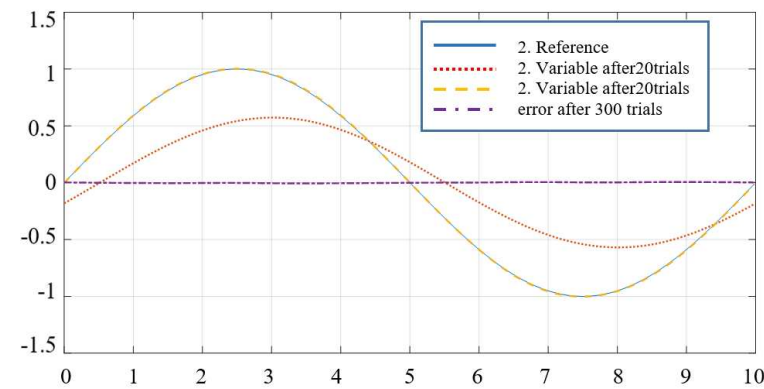


Figure 13. Meets the position of the second joint when used (40).

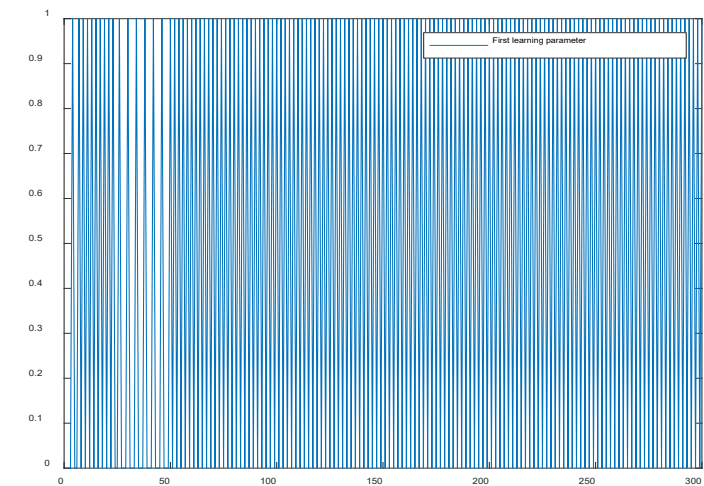


Figure 14. Change in the first learning function parameter when using (39).

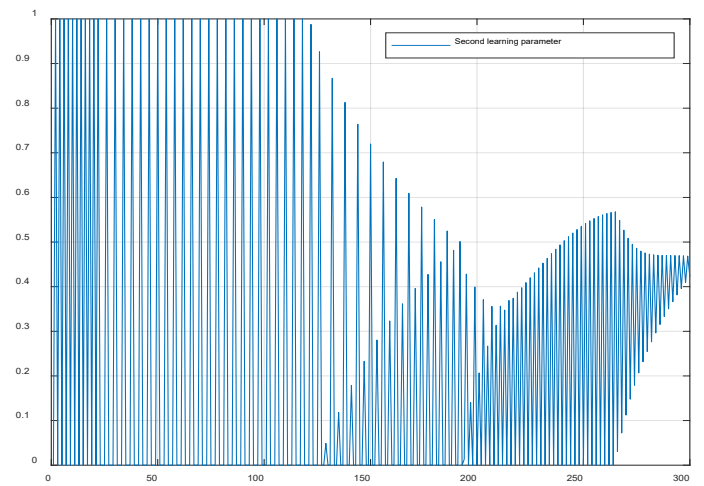


Figure 15. Change in the second learning function parameter when using (39).

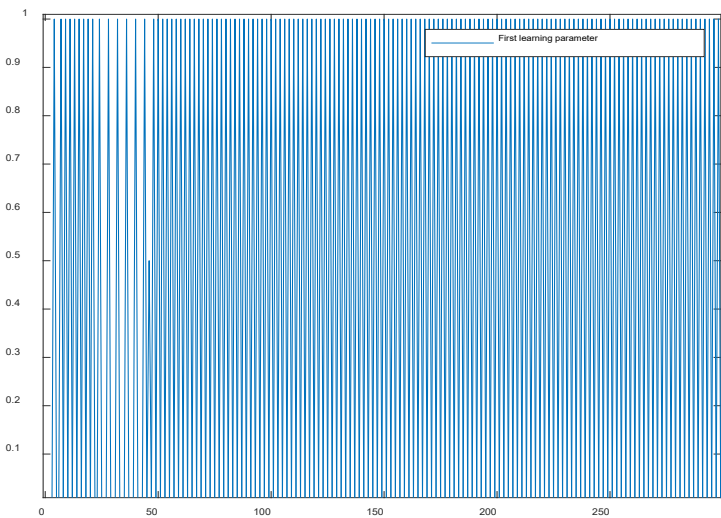


Figure 16. Change in the first learning function parameter when using (40).

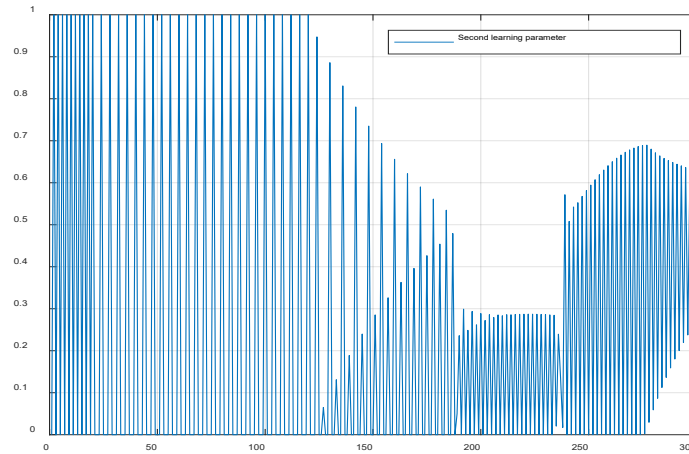


Figure 17. Change in the second learning function parameter when using (40).

Comment: Figure 10 and Figure 11 are the results of two output channels, respectively, after 20 and 300 trials when the controller learns iteratively using the formula.

$$K = \underline{a}_k \cdot \arg \min_{0 \leq z \leq 1} \|z \underline{e}_k (1-z) \underline{e}_{k-1}\| \quad (39)$$

$$K = \underline{a}_k \cdot \arg \min_{0 \leq z \leq 1} \|\underline{e}_k (1-z) + z \underline{e}_{k-1}\| \quad (40)$$

For "Smart" determination, the learning function parameter is $K_k = \text{diag} \left(K_k^{(j)} \right), j = 1, 2, \dots, n$ the input/output signal pair index $u_j \mapsto q_j$. This result shows that the controller has provided the required tracking quality, and after 300 trials, the output signal has tracked the password set in both channels.

When using formula (40) to determine the intelligent learning parameter in the iterative learning controller instead of (39), the quality of tracking the set signal is unchanged. The difference now is only the appropriate change. The results of the two learning parameters K_k^1, K_k^2 are shown in Figures 14, 15, 16, and 17.

7. Conclusions

Propose a two-loop control structure for the robot, in which the inner loop is an uncertain function component estimator to compensate for the robot's input, and the outer circle is an iterative learning controller (Figure 2). The Taylor series analysis method was used to create an inner loop controller. This method does not necessitate the creation of a mathematical model of the robot. For estimating, it simply takes measured signal values from previous systems. Develop two options for installing an iterative learning controller in the outer loop. The difference between these two options lies in the online determination of learning function parameters. The two proposed control circuits have been installed into the control algorithm and simulated with the planar robot to evaluate the quality. The correctness of the above-mentioned solutions has been demonstrated by theory and validated by simulation - Create a control algorithm based on the concepts above for controlling two items that work in batch processes in industry robots. These control algorithms are all distinguished because they do not (or only very infrequently) employ the mathematical model of the control object. The quality of these control algorithms has also been verified through a 2-DOF robot simulation. However, there are still disadvantages when building methods to determine convergence parameter sets for nonlinear learning functions with known structures. The problem of constrained control by iterative learning still needs to be solved. These will be the following research directions.

References

1. L.Lewis, D.M. Dawson and C.T.Abdallah, " *Robot manipulator control theory and practice*", Marcel Dekker, 2004.
2. W. Spong, S. Hutchinson, and M. Vidyasagar," *Robot modeling and control*", New York, Wiley, 2006.
3. Z. S. Jiang, F. Xei, X. Wang and Z. Li, " *Adaptive dynamic sliding mode control for space manipulator with external disturbance*", Journal of Control and, 2019.
4. Goel, A Swarup, " *MIMO uncertain nonlinear system control via adaptive high-order super twisting sliding mode and its application to robotic manipulator*", Journal Control Autom. Electr. Syst., 28, 36-49, 2017.
5. Wang, Y et.al.; " *Servey on iterative leaning control, repetitive control and run to run control. Journal of process control*", 19, (10), 589-1600, 2009.
6. R. Lee, L. Sun, Z. Wang, M.Tomizuka," *AdaptivebILCcontrol of robot manipulators for friction compensation*". IFAC PapersOnline 52 (15), 175-180, 2019.
7. F. Boiakrif, D.Boukhetala and F.Boudjema Velocity observer-based ILCcontrol for robot manipulators. F Bouakrif, D Boukhetala, F Boudjema. International Journal of Systems Science 44 (2), 214,222, 2013.
8. P.D. Nguyen, N.H Nguyen, " *Adaptive control for nonlinear non-autonomous systems with unknown input disturbance*. International Journal of Control, European Journal of Control, 61,91-100, 2021.
9. R. Jeyasenthil and S-B. Choi." *A robust controller for multivariable model matching system utilizing a quantitative feedback theory: Application to magnetic levitation*". Applied Sciences, 27 April, 2019.
10. P.D. Nguyen, N.H Nguyen, " *Adaptive control for nonlinear non-autonomous systems with unknown input disturbance*. International Journal of Control, 2021
11. K. G. Tran, N. H. Nguyen, P. D. Nguyen "Observer based controller two - wheeled inverted robots with unknown input disturbance and model uncertainty", Journal of Control Science and Engineering, 1-12, 2020.
12. Sabir Husnain, Rasheed Abdulkader, " *Fractional Order Modeling and Control of an Articulated Robotic Arm*", Engineering, Technology & Applied Science Research (ETASR), 12026-12032, 2023.
13. L. Zouari, S. Chtourou, M. Ben Ayed, S. A. Alshaya, "A Comparative Study of Computer-Aided Engineering Techniques for Robot Arm Applications", Engineering, Technology & Applied Science Research (ETASR), Pages: 6526-6532, 2020.
14. M. Ben Ayed, L. Zouari, M. Abid, "Software In the Loop Simulation for Robot Manipulators", Engineering, Technology & Applied Science Research (ETASR), Pages: 2017-2021, 2017.
15. D. M. Duc, T. X. Tuy, P. D. Phuoc, "A Study on the Response of the Rehabilitation Lower Device using Sliding Mode Controller", Engineering, Technology & Applied Science Research (ETASR), Pages: 7446-7451, 2021.

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.