CONTROL OF INDUSTRIAL ROBOTS USING PD ITERATIVE LEARNING TYPES USING SMART ONLINE LEARNING PARAMETERS

Vo Thu Ha
Department of Control and Automation, Faculty of Electrical Engineering University of Economics-Technology for Industries
Hanoi, Vietnam
vtha@uneti.edu.vn

Than Thi Thuong
Department of Control and Automation, Faculty of Electrical Engineering University of Economics-Technology for Industries
Hanoi, Vietnam
ttthuong.dien@uneti.edu.vn

Vo Quang Vinh
Faculty Control and Automation Electric Power University
Hanoi, Vietnam
vinhvq@epu.edu.vn

ABSTRACT

A robot motion system always depends on mathematical model parameters that are uncertain or not precisely known and are unavoidably affected by external disturbances. There are many traditional control methods, modern control methods, and intelligent control methods to handle this case, however, those control methods are all based on the uncertain robot mathematical model, which requires at least estimating those parameters or assuming the parameters are constant and uncertain. The article's content is to present a control method for moving robots to follow precise orbits without relying entirely on the model, which is an adaptive controller thanks to iterative learning types PD. Used smart online learning parameters. Simulation results for a two-degree-of-freedom manipulator have provided the required tracking quality, and after 50 trials, the output signal can track the signal set at both joints.

Keywords: *Robot with 2 degrees of freedom, iterative learning control; estimated according to the Taylor series*

1. INTRODUCTION

Robots have been widely used in many industrial fields, so there have been countless robot control methods researched and applied, helping robots achieve accurate trajectory tracking. Qualified. 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 type of mathematical model commonly used for designing and synthesizing controllers is the Euler - Lagrange model, which is affected by external input noise d(t) as follows:

$$u = M\left(q,\theta\right).\ddot{q} + C(q,\dot{q},\theta)\dot{q} + G(q,\theta) + F(\dot{q},\theta) + d(t) \quad (1)$$

In which q is the vector of joint variables; θ is the vector of uncertain parameters; $M\left(q,\theta\right)$ is the inertia matrix that is always positive definite symmetry; $C(q,\dot{q},\theta)\dot{q}$ is the vector of Coriolis and centrifugal components; $F(\dot{q},\theta)$ describes the effect of friction; $G(q,\theta)$ is the vector gravity; $d(t)$ 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 q that follow the desired trajectory r(t), and η the tracking quality does not depend on uncertain parameters and components and external noise d(t). owever, 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 d(t) 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 and 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, which as the control trend of not using the robot's dynamic model (1), so that the control quality is not affected. by components q(t) and d(t). The intelligent control method mainly applied to robots mentioned in the article is the iterative learning method [5]. This is an integrated control method with cyclic working systems and requires the same state. At the same time, it is also required that the parameters q(t) and d(t) are periodic and have the same change period as the working cycle of industrial robots. In reality, the iterative learning control method is only sometimes applied to meet the desired control quality. Currently, there have been many improvements to iterative learning to improve control quality and expand the scope of practical applications. The first is to improve the combination of iterative learning with traditional control methods, such as separating the friction component in [6], which is impossible in [7]. The disadvantage of stabilizing the control object with a traditional controller is that it loses the "intelligence" of the final controller. In other words, the preprocessing and feedforward iterative learning controller, as above, is no longer an intelligent controller. Next, stabilize through "smart" adaptive control according to the sample model with the structure shown in Figure 2 [8]. "Intelligence" is the block of estimating uncertain components without using the mathematical model of the system (control object). The insecure features of the system include input noise and model deviations (compared to the sample model), and this is a controlled trend that can be applied to the control of industrial robots. The main content of the article will present a control algorithm to stabilize the robot system without needing to use traditional control methods. This creates the possibility of applying iterative learning control to unstable systems without using the mathematical model of the robot system, applicable to the case where q(t) d(t) depends on time, with no need for circulation. This control method is built on the theoretical basis of optimal control of each segment on the time axis [9]. The content of the article includes four parts. Part 1 is the problem of researching iterative learning for robot systems. Part 2 will present an intelligent adaptive controller based on a sample model for a robot system. Part 3 will apply the proposed method to a two-degree-of-freedom robot to verify the control method. Part 4 is conclusions and future research directions.

2. INTELLIGENT ADAPTIVE CONTROL, ACCORDING TO THE MODEL FOR INDUSTRIAL ROBOT SYSTEMS

2.1. Proposed control structure diagram

There are many industrial robot actuators, but they all have the same mathematical model as the Euler – Lagrange equation (1). Build an intelligent controller so that the joint variable vector q can follow the sample trajectory r(t) and the quality of robot joint tracking is unaffected by d(t). Use the control structure diagram shown in Figure 1, which is based on a 2-circuit structure, to help apply iterative learning control better.

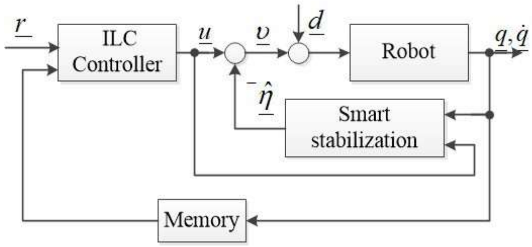

Fig. 1. *Control structure without using robots' mathematical model*

The proposed method of this inner loop controller uses an intelligent linearized controller with a state feedback unit in the inner loop, as depicted in Figure 1. The task of this smart linearized controller is the initial robot system transfer (1) rewritten in an equivalent form with total noise, including initial d and model error as follows:

$$\ddot{q} = u + \eta \qquad (2)$$

With:

$$\eta = d + \left[I_n - M(q,\theta)\right]\ddot{q} - C(q,\dot{q},\theta)\dot{q} - F(\theta)\dot{q} - g(q,\theta) \qquad (3)$$

And (2) becomes a linear system through estimation η and $\hat{\eta}$ then proceeds to compensation through the input signal:

$$\ddot{q} = u + \eta - \hat{\eta} = u + \delta \qquad (4)$$

With $\delta = \eta - \hat{\eta}$ is the remaining estimate error.

The task of the inner loop controller is to make the estimated results as accurate as possible. The more precise the calculated result, the smaller the remaining uncertainty component. In addition, the linearizer, thanks to estimation and uncertainty compensation, must be "smart," which means there is no need to use the robot dynamics equation (1).

The task of the outer loop controller is: The controller learns iteratively to determine the control signal u, making the trajectory of the joint variables q stick firmly to the given sample trajectory r. This iterative learning controller will use a PD-style learning function with learning parameters adjusted online after every kth trial according to minimizing the sum of squared tracking errors.

For convenience of presentation, we will use the state vector symbol and input signal for system (2) as follows:

$$x = \begin{pmatrix} q \\ \dot{q} \end{pmatrix} \text{và } u = \upsilon - A_1 q - A_2 \dot{q} \qquad (5)$$

Where A_1 and A_2 are two optional matrices. Then, the equivalent model (2) of the robot above becomes the canonical state equation form.

$$\dot{x} = Ax + B\left[\upsilon + \delta\right] \qquad (6)$$

$$A = \begin{bmatrix} 0_n & I_n \\ -A_2 & -A_2 \end{bmatrix} \text{và } B = \begin{pmatrix} 0_n \\ I_n \end{pmatrix}, \quad \delta = \eta - \hat{\eta} \qquad (7)$$

And are the zeros and the identity matrix of dimension n x n, respectivity.

It can be seen that except in (3), this equivalent model no longer contains any information about the Euler – Lagrange model of the original robot motion system. Therefore, if compensated according to the principle of Taylor series analysis by an inner loop controller, this controller does not need to use model (1), which is intelligent. That is the reason for the name smart internal loop controller or intelligent stabilization and linearization controller with state feedback.

2.2. Estimate the derivative of the function vector from measured data.

To estimate the derivative value of a function vector, we use the estimation method using Taylor analysis, specifically as follows:

$$\dot{x}_k(i) = B_1^T x_k(i) - B_0^T \dot{x}_k(i-1) + \frac{T_s^2}{2}\ddot{x}_k(\zeta) \quad (8)$$

$$B_1^T = -B_0^T = \frac{1}{T_s} \quad (9)$$

Are the corresponding Taylor series coefficients and $(i-1)T_s < \zeta < iT_s$. It can be seen that, in this case $\ddot{q}_k(\zeta) = 0, \forall\zeta$ the formula approximates the derivative:

$$\dot{x}_k(i) \approx \frac{x_k(i) - x_k(i-1)}{T_s} \quad (10)$$

It becomes exact, i.e., the Taylor series analysis is error-free.

2.3. Design an external circuit controller.

The iterative learning controller uses a PD-type learning function:

$$u_{k+1}(i) = u_k(i) + K_1 e_k(i) + K_2 e_k(i+1), i = 0,1,\ldots,N-1$$

(11)

Correspondingly, at the end of the working cycle at i = N-1 it becomes:

$$u_{k+1}(N-1) = u_k(N-1) + (K_1 + K_2) e_k(N-1)$$

(12)

Sufficient conditions to ensure convergence for the learning process when using PD-style tuning, the learning parameters need to be satisfied:

$$\left\| I_m - \hat{C}\hat{B}K_2 \right\| + \left\| \hat{C}\hat{B}K_2 \right\| < 1$$

(13)

2.4. Principles for online determination of learning function parameters

To separate the mathematical model of the control object from the task of determining the convergence parameter for the learning function, that convergence parameter can only be based on the information measured from the thing, which here is the clinging error e_k. So, the principle of determining the learning function parameter K_k for the k + 1 trial from the two tracking error vectors measured in the entire two immediately preceding tests ε_k and ε_{k-1} is such that with this result, we will get a vector of errors ε_{k+1}. The adhesion for the k + 1 trial is similar to the previous two.

2.5. Design an internal loop controller

This is a loop circuit that uses Taylor series analysis to estimate $\hat{\eta}$ at time $t = kT + iT_s$ from 2 previously measured values including with . Follow these 2 steps:

Step 1: Use intelligent adaptive compensation method $u = \upsilon - \hat{\eta}$ to stabilize the robot:

$$\ddot{q} = -A_1 q - A_2 \dot{q} + \upsilon - \hat{\eta} + \eta$$

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

Step 2: Use the iterative learning principle to design the controller

Perform step 1: Convert the system when compensating

$$\Leftrightarrow \frac{dx}{dt} = \begin{pmatrix} 0 & I \\ -A_1 & -A_2 \end{pmatrix} x + \begin{pmatrix} 0 \\ I \end{pmatrix}\left[\upsilon - \hat{\eta} + \eta\right] = = Ax + B\left[\upsilon - \hat{\eta} + \eta\right]$$

(14)

Then the estimation formula is determined as follows:

$$\hat{\eta}(i) = B^T\left[\frac{x_k(i) - x_k(i-1)}{T_s} - Ax_k(i)\right]\ddot{q} - \upsilon_k(i) + \hat{\eta}(i-1)$$

(15)

2.6 Control algorithm

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

Choose two matrices A_1 , A_2 and choose learning parameter K . Choose T_s and calculate $N = T / T_s$. Assign $\upsilon(i) = r(i), \mathrm{i} = 0,1,...,N-1; z = 0$. Choose arbitrarity $\hat{\eta}$

while continue the control **do**

for $i = 0,1,...,\mathrm{N}-1$ **do**

Send $u(\mathrm{i}) = \upsilon(i) - A_1 q - A_2 \dot{q} - \hat{\eta}$ to robot for a white of T_s

Measure x . Detemine $e_k(i) = r(i) - q(i)$.

Calculate

$\hat{\eta} \leftarrow B^T\left[(x-z)/T_s - Ax\right] - \upsilon(i) + \hat{\eta}; \hat{z} \leftarrow x$

end for

Establish

$\mathbf{v} = vec(\upsilon(0),....,\upsilon(N-1)); e = vec(e(0),....,$

Set $v \leftarrow v + K\dot{e}$

end while

In this algorithm, each while - do loop represents one working cycle (one attempt). In addition, it can be seen that the above control algorithm does not use the mathematical model of the robot at all.

3. Mathematical mode of 2 – dof planar robot

To test the control quality of the above control algorithm, it has been installed for a 2-degree-of-freedom robot, figure 2.

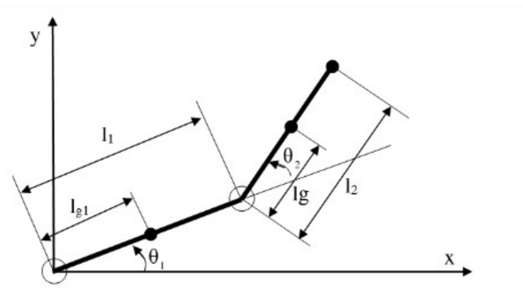

Fig. 2. *Model of 2 - DOF*

The 2-degree-of-freedom robot system (Figure 4) has the Euler - Lagrange model (1) with the following specific results:

$$\begin{cases} M_{11} = & m_1 l_{C1}^2 + I_1 + m_2\left(l_1^2 + l_{C2}^2 + 2 l_1 l_{C2} \cos\theta_2\right) + I_2 \\ M_{12} = & m_2\,(l_{C2}^2 + l_1 l_{C2} \cos\theta_2) + I_2 \\ M_{21} = & m_2\,(l_{C2}^2 + l_1 l_{C2} \cos\theta_2) + I_2 \\ M_{22} = & m_2\, l_{C2}^2 + I_2 \end{cases}$$

$$\begin{cases} H_{11} = & -m_2 l_1 l_{C2} \sin\theta_2 (2\dot\theta_1 \dot\theta_2 + \dot\theta_2^2) \\ H_{12} = & m_2 l_1 l_{C2}\left(\dot\theta_1^2 + \dot\theta_1\dot\theta_2\right)\sin\dot\theta_2 - m_2 l_1 l_{C2} \sin\theta_2 \dot\theta_1 \dot\theta_2 \end{cases}$$

$$\begin{cases} G_{11} = & m_1 l_{C1} \cos\theta_1 + m_2 g\left(l_1\cos\theta_1 + l_{C2}\cos(\theta_1+\theta_2)\right) \\ G_{12} = & m_2 g l_{C2} \cos(\theta_1+\theta_2) \end{cases}$$

Friction components::

$$F(q,\dot q,\theta)\,\dot q = \begin{pmatrix} \theta_{11} f_1 \\ \theta_{12} f_2 \end{pmatrix}; f_1 = \dot q_1 + q_1 \tanh \dot q_1; f_2 = \dot q_2 + q_2\,\text{ta}$$

With input noise assumed as follows:

$$d_1 = q_1 \dot q_2 \sin(2\pi t),\; d_2 = q_2 \dot q_1 \sin(0.6\pi t)$$

The parameters $\theta_i, i = 1 \div 20$ are random. The robot is assumed to have a duty cycle $T_s = 10s$. Control signal update time is $T_s = 0.025s$ selected. The orbits are set as two periodic functions of period T as follows:

$$r_1 = 2\sin(\pi t/T) - \sin(5\pi t/T);$$
$$r_2 = \sin(\pi t/T) - 0.5\sin(5\pi t/T);$$

And controller parameters:

$$A_1 = \begin{bmatrix} 31 & 0 \\ 0 & 12 \end{bmatrix}, A_2 = \begin{bmatrix} 12 & 0 \\ 0 & 8 \end{bmatrix}, K_1 = 12, K_2 = 7$$

Comment: The matching variables q_2, q_2 and desired values, as shown in Figure 4 and Figure 5, show that the matching variables all approach the preset value with the tracking error after 50 trials being small enough and acceptable specifically for the joint. The first is:

$$\max|e_1| \approx 0.04; \max|e_2| \approx 0.02;$$

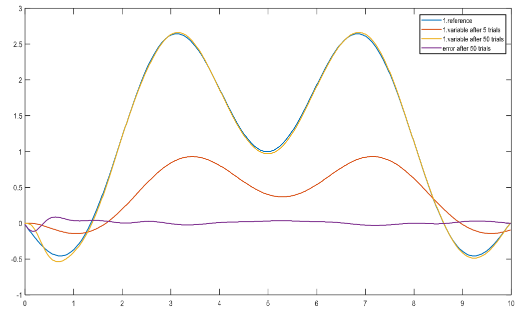

Fig. 3. *Results of adhesion at the first joint*

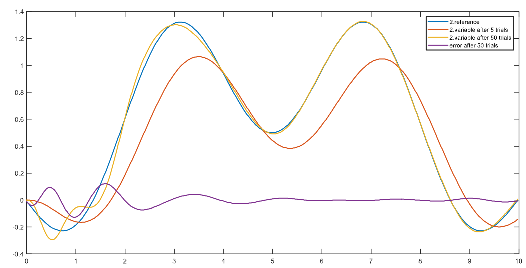

Fig. 4. *Adhesion results at the second joint*

CONCLUSION

In the article, an algorithm has been built to install an iterative learning controller in the outer loop of type D learning and online determination of the proposed learning function parameters. Simulation results for a 2-degree-of-freedom robot have demonstrated the effectiveness of the model-based control method. The remaining problems in the following research results are to develop strategies to determine the set of convergence parameters K for a nonlinear function with a known structure. In addition, the problem of constrained control has yet to be considered by iterative learning. The following research direction will be to solve the problem of constrained iterative learning conditions shortly.

REFERENCES

[1] F.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. Rovot modeling and control. New York, Wiley, 2006.

[3]]. Z. S. Jiang, F. Xei, X. Wang anf Z. Li. Adaptive dynamic sliding mode control for space manipulator with external disturbance. Journal of Control and, 2019.

[4] A. Goel, A Swarup. MIMO uncertain nonlinear system control via adaptive hight-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.; Servy 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. Adaptivebiterative learning control of robot manipulators for friction compensation. IFAC PapersOnline 52 (15), 175-180, 2019Writer's Handbook. Mill Valley, CA: University Science, 1989.

[7] R. Lee, L. Sun, Z. Wang, M.Tomizuka. Adaptivebiterative learning control of robot manipulators for friction compensation. IFAC PapersOnline 52 (15), 175-180, 2019.

[8] 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

[9] P.D. Nguyen, N.H Nguyen. Adaptive control for nonlinear non-autonomous systems with unknown input disturbance. International Journal of Control, 2021.