Learning and Uncertainty Compensation in Robotic Motion Systems using Li-Slotine Adaptive Tracking and Intelligent Adaptive Control

Vo Thu Ha
Faculty of Electrical-Automation Engineering, University of Economics - Technology for Industries (UNETI), Hanoi, Vietnam
vtha@uneti.edu.vn (corresponding author)

Thanh Trung Cao
School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam
trung.caothanh@hust.edu.vn

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

Thi Thanh Nguyen
Faculty of Electrical-Automation Engineering, University of Economics - Technology for Industries (UNETI), Hanoi, Vietnam
Nnthanh.đt@uneti.edu.vn

Hong Diem Bui Thi
Faculty of Electrical-Automation Engineering, University of Economics - Technology for Industries (UNETI), Hanoi, Vietnam
bthdiem@uneti.edu.vn

Received: 5 December 2024 | Revised: 25 December 2024 | Accepted: 1 January 2025

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | DOI: https://doi.org/10.48084/etasr.9843

ABSTRACT

This paper examines adaptive control strategies for stabilizing robots, focusing on Li-Slotine adaptive control and Iterative Learning Control (ILC). Both methods handle uncertainties through learning and compensation, ensuring stability and precision. Li-Slotine control, based on Lyapunov theory, dynamically adjusts parameters for asymptotic stability in uncertain systems. ILC improves performance in repetitive tasks by refining control inputs using tracking errors, making it suitable for robotics and manufacturing. While Li-Slotine excels in real-time adaptation and robustness to disturbances, its computational demands challenge high-degree-of-freedom systems. ILC enhances accuracy through iterative learning but is sensitive to noise and requires careful tuning. MATLAB simulations and experimental results demonstrate the effectiveness of both approaches. Future work will explore hybrid frameworks that combine the adaptability of Li-Slotine with the data-driven refinement of ILC to provide robust solutions for complex, dynamic robotic systems.

Keywords-iterative learning control; industrial robots; Taylor series analysis

I. INTRODUCTION

The robot's dynamic equation in its general form [1] defines the control quantity as torque, with the joint variable and moving trajectory as the control values. Unknown parameters and joint interconnections influence trajectory tracking accuracy. Thus, a controller must be designed to ensure precise trajectory tracking without relying on uncertain model parameters or axis interdependence. The error between desired and actual joint angles must remain minimal (<0.1%). Traditional regulators with fixed parameters, such as gravity-compensated PD controllers [2], PID controllers [1], and model-based nonlinear control [1], cannot meet these requirements as they are ineffective for systems with channel interleaving. Therefore, an adaptive control system for multi-degree-of-freedom robot motion control is required to address these challenges.

The adaptive control method estimates unknown parameters during operation using the system's input and output signals and adapts them to calculate control signals [3-6]. These controllers typically follow a fixed structure, often based on classical designs, while their parameters are updated after each sampling cycle according to the adaptive update law. The resulting signal is then sent to the controller. Adaptive control methods include standard model-based approaches, explicit assumptions, and integrated adaptive laws. Other techniques, such as sustainable adaptive control and adaptive control with state observers, improve control performance by maximizing advantages and minimizing disadvantages.

Manipulator models often have uncertain parameters such as load, mass, inertia, and joint friction. The Li-Slotine adaptive control law addresses this by continuously updating dynamic parameters to ensure stability and minimize joint position errors based on Lyapunov stability criteria. While effective, it suffers from chattering and incomplete error elimination, which may harm actuators. Like neural networks for error correction, this approach requires high computational effort to solve constrained nonlinear optimization problems in real time.

Most industrial control methods rely heavily on mathematical models, yet such models lose accuracy over time as device properties change, affecting control performance. Although initially accurate, traditional controllers degrade over time. Iterative Learning Control (ILC) offers a practical solution by iteratively adjusting the control signal. While sometimes classified as adaptive, ILC differs from adaptive controllers in that it only updates the control signal, not the dynamic controller itself. Similarly, ILC differs from intelligent control systems using neural networks in that it does not re-adjust controller parameters, but only re-edits control signals, according to [7]. Such a control principle is called learning based on experience to improve the quality of current and future control, also known as control through the automatic learning process. As a result, this technology can be added to the traditional control system.

The conventional procedure is to develop a new mathematical model for the control object, including the actuator, and then calibrate it with this model to restore control quality. Over time, the traditional system struggles to maintain its initial quality due to equipment wear and model inaccuracies. Recalibration of such controllers typically requires rebuilding the system's mathematical model, redesigning, and reinstalling the controller. Studies [8-10] suggest combining iterative learning with the traditional controller to simplify this process, forming a feedforward iterative learning controller. However, iterative learning is only suitable for volatile processes [11], where the system is first stabilized using a traditional controller [12-19], a method known as indirect iterative learning control. This combined controller, while effective, still relies on the system's mathematical model, limiting its intelligent application. Thus, the focus shifts to intelligently stabilising the system (e.g., BIBO, ISS, or uniformly ultimately bounded) without heavy dependence on the model, allowing iterative learning to handle output tracking. This paper focuses on extending ILC to unstable systems without relying on traditional stabilization methods. Solving this issue would enable the direct application of ILC to unstable and nonlinear systems, eliminating the need for transmission iterative control structures. While linear or pre-processed ILC has been widely used for various unstable nonlinear systems, this study aims to broaden its scope.

II. ROBOT CONTROL USING LI-SLOTINE METHOD

A. *Algorithmic Content*

Consider the dynamic equation of a manipulator without indeterminate constant parameters, written as:

$$\tau = M.\ddot{q} + C\dot{q} + G \tag{1}$$

Using the pseudo-control definer, defined as:

$$\tau = M.\dot{v} + Cv + G + K_d(v - \dot{q}) \tag{2}$$

where:

$$\begin{aligned} v &= \dot{q}_d + \Lambda(q_d - q) = \dot{q}_d + \Lambda e \\ r &= \dot{e} + \Lambda e = \dot{q}_d - \dot{q} + \Lambda(q_d - q) = v - \dot{q} \end{aligned} \tag{3}$$

K_d is the optional positive-definite symmetric matrix, and $\Lambda = diag(\lambda_i)$ is the positive definite diagonal matrix $(\lambda_i > 0)$. We must prove that $e \to 0$. Because Λ is positive definite, that is, $\dot{e} = -\Lambda e$ when $r = \dot{e} + \Lambda e = 0$ it always has $e \to 0$.

Using positive definite function:

$$V_1(r) = r^T M r \tag{4}$$

$$\dot{V}_1(r) = r^T \dot{M} r + r^T \dot{M} r + r^T M \dot{r} = 2r^T M \dot{r} + r^T \dot{M} r \tag{5}$$

since $M = M^T$, from (1) and (2) there is also:

$$\begin{aligned} &M\ddot{q} + C\dot{q} + G = M\dot{v} + Cv + G + K_d(v - \dot{q}) \\ \Leftrightarrow\ &0 = M(\dot{v} - \ddot{q}) + C(v - \dot{q}) + K_d(v - \dot{q}) \\ \Leftrightarrow\ &0 = M\dot{r} + Cr + K_d r \Rightarrow M\dot{r} = -(Cr + K_d)r \end{aligned}$$

$$\Rightarrow \dot{V}_1(r) = -2r^T\left(C+K_d\right)r + r^T\dot{M}r = -r^T\left[2C+K_d-\dot{M}\right]r \qquad (6)$$

because:

$$\dot{M} = C + C^T \Leftrightarrow r^T\dot{M}r = r^T\left(C+C^T\right)r = 2r^TCr$$

$$\Rightarrow \dot{V}_1(r) = -r^TK_dr \le 0$$

If $r \to 0$ the error has not yet reached zero ($e \to 0$), the system will appear to vibrate. Since the manipulator's motion equation contains uncertain constant parameters, the assumed controller (2) also includes these uncertain components, so the robot's motion system will no longer be as accurate as in the explicitly assumed controller (2). In this case, some indirect adaptive control methods should be considered. We will change the uncertain parameter with the adjustment mechanism to finally achieve $r \to 0$, which means we still have a stable tracking condition $e \to 0$. The following assumptions are needed to apply the Li-Slotine adaptive control law to the robot motion system with precise trajectory tracking: the dynamic model has enough actuators; the uncertainty in the dynamic model is that the constant parameters are not known precisely or not known. From (1), the Li-Slotine adaptive control law is given as follows:

$$\tau = \hat{M}\dot{v} + \hat{C}v + \hat{G} - K_dr = W(q,\dot{q},v,\dot{v})\hat{p} - K_dr \qquad (7)$$

where $\hat{C},\hat{G},\hat{M}$ are the estimated components of C,G,M, $\hat{p}$ is the adjustment component of the manipulator model, and $W(q,\dot{q},v,\dot{v})$ is the following determination matrix, abbreviated $W(.)$.

Combining (1) and (7) gives the closed dynamic equation:

$$M\ddot{q} + C\dot{q} + G = \hat{M}\dot{v} + \hat{C}v + \hat{G} - K_dr \qquad (8)$$

with:

$$\begin{cases}\ddot{q} = \dot{v} - \dot{r}\\ \dot{q} = v - r\end{cases} \Rightarrow \begin{cases}\dot{r} = \dot{v} - \ddot{q}\\ r = v - \dot{q}\end{cases} \qquad (9)$$

The deviations between the actual value and the estimated value are as follows:

$$\tilde{E}_M = M - \hat{M};\tilde{E}_C = C - \hat{C};\tilde{E}_G = G - \hat{G};\tilde{e}_p = p - \hat{p} \qquad (10)$$

With this setting, the closed dynamic equation of the robot becomes:

$$M\dot{r} + Cr + K_dr = \tilde{E}_M\dot{v} + \tilde{E}_Cv + \tilde{E}_G = W(q,\dot{q},v,\dot{v})\tilde{e}_p \qquad (11)$$

To build a structure for adjusting the parameter vector *p(t)*, we use a positive definite function:

$$V_2(r) = \frac{1}{2}r^TMr + \frac{1}{2}\tilde{e}_p{}^T\Gamma^{-1}\tilde{e}_p \qquad (12)$$

where Γ is the diagonal matrix, favourable definite, $\Gamma = diag(\lambda_1,\lambda_2,...,\lambda_n)$ option.

$$\begin{aligned}\Rightarrow \dot{V}_2(r) &= \frac{1}{2}\dot{r}^TMr + \frac{1}{2}r^T\dot{M}r + \frac{1}{2}r^TM\dot{r} + \tilde{e}_p{}^T\Gamma^{-1}\dot{\tilde{e}}_p\\ &= r^TM\dot{r} + \frac{1}{2}r^T\dot{M}r + \tilde{e}_r{}^T\Gamma^{-1}\dot{\tilde{e}}_p\end{aligned} \qquad (13)$$

so: $M(q) = M^T(q)$.

Combined with dynamic equations (1) and (7), we have:

$$M\dot{r} + C(q,\dot{q})r = [M\dot{v} + CV + G] - [M\ddot{q} + C\dot{q} + G]$$

$$\Rightarrow \quad M\dot{r} = \mathrm{W}(q,\dot{q},v,\dot{v})p - \tau - Cr \qquad (14)$$

Substituting (13) we can determine:

$$\dot{V}_2(r) = r^T(\mathrm{W}(q,\dot{q},v,\dot{v})p - \tau) + r^T\left(\frac{1}{2}\dot{M} - Cr + \tilde{e}_p{}^T\Gamma^{-1}\dot{\tilde{e}}_p\right) \qquad (15)$$

So, $S = C - \frac{1}{2}\dot{M}$ is a skewed symmetric matrix, i.e.:

$$r^T(\frac{1}{2}\dot{M} - C)r = 0 \qquad (16)$$

$$\Rightarrow \dot{V}_2(r) = r^T(\mathrm{W}(q,\dot{q},v,\dot{v})p - \tau) + \tilde{e}_p{}^T\Gamma^{-1}\dot{\tilde{e}}_p \qquad (17)$$

$$\begin{aligned}\Leftrightarrow \dot{V}_2(r) &=\\ &r^T(\mathrm{W}(q,\dot{q},v,\dot{v})p - \mathrm{W}(q,\dot{q},v,\dot{v})\hat{p} - K_dr) + \tilde{e}_p{}^T\Gamma^{-1}\dot{\tilde{e}}_p\\ &= -r^TK_dr + \tilde{e}_p{}^T\mathrm{W}^Tr + \tilde{e}_p{}^T\Gamma^{-1}\dot{\tilde{e}}_p\\ &= -r^TK_dr + \tilde{e}_p{}^T(Y^Tr + \Gamma^{-1}\dot{\tilde{e}}_p)\end{aligned} \qquad (18)$$

with adaptive update rules:

$$\dot{\hat{p}} = -\dot{\tilde{e}}_p = \Gamma Y^Tr \qquad (19)$$

$$\Rightarrow \dot{V}_2(r) = -r^TK_dr \le 0 \qquad (20)$$

means guaranteed system: $r = \dot{e} + \Lambda e \to 0$ when $t \to \infty$ $\Rightarrow \lim_{t\to\infty} r = 0 \Rightarrow \lim_{t\to\infty} e = 0$.

This ensures that the manipulator's trajectory approaches the set trajectory as time approaches infinity. The block diagram of the Li-Slotine adaptive control system is shown in Figure 1.

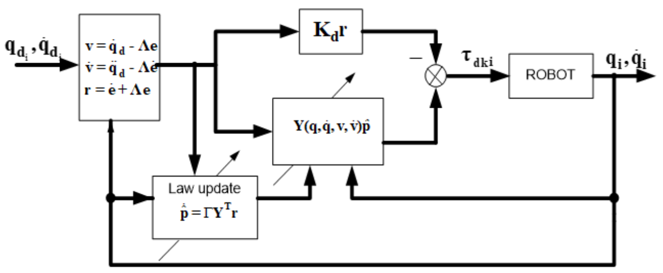
$q_{d_i}, \dot{q}_{d_i}$
$v = \dot{q}_d - \Lambda e$
$\dot{v} = \ddot{q}_d - \Lambda\dot{e}$
$r = \dot{e} + \Lambda e$
K_dr
−
τ_{dki}
ROBOT
$q_i, \dot{q}_i$
$Y(q,\dot{q},v,\dot{v})\hat{p}$
Law update
$\dot{\hat{p}} = \Gamma Y^Tr$

Fig. 1. Li-Slotine adaptive control system diagram.

B. Simulation and Verification via Robot with 3 Degrees of Freedom using Matlab/Simechanic Software

The control object estimation parameters are assumed as follows: $\hat{m}_1 = 67$ kg , $\hat{m}_2 = 52$ kg , $\hat{m}_3 = 16$ kg , $\hat{l}_1 = 0{,}11$ m , $\hat{a}_2 = 0.055$, and $\hat{a}_3 = 0.045$. The parameters of the Li-Slotine adaptive controller are shown in Table I, and the results of the 3-degree-of-freedom robot dynamic equation, determined in [20], are shown in Figures 2 and 3.

TABLE I. PARAMETERS OF THE LI-SLOTINE ADAPTIVE CONTROLLER

Symbol	Parameter name	Parameter values of joint axes
K_d	Adjustment factor	$K_{d_1} = 1000$, $K_{d_2} = 1500$, $K_{d_3} = 2000$
Γ	Positive diagonal matrix	$\Gamma_1 = 282$, $\Gamma_2 = 285$, $\Gamma_3 = 282$

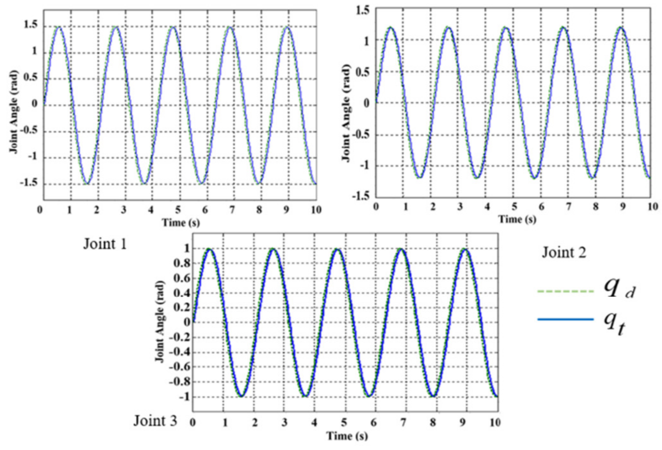

Fig. 2. Representation of the response between set joint angles (q_d) and actual joint angles (q_t).

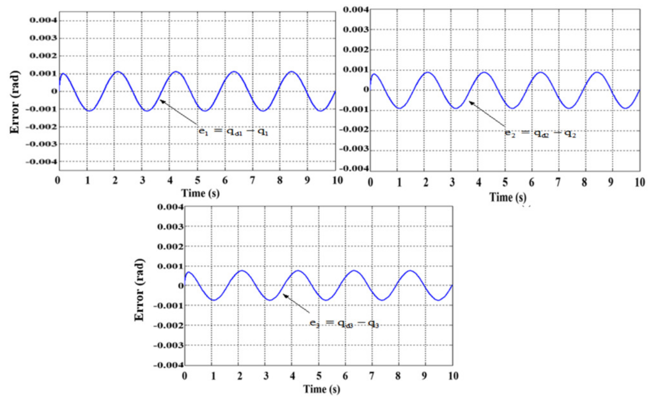

Fig. 3. Representation of the deviation between set joint angles (q_d) and actual joint angles (q_t)

The actual joint angles of the three joints all adhere to the set joint angles, and since the difference between the set joint angles and the actual joint angles is consistently small (<0.002), the system is stable.

C. Experimental Model for Applying the Li-Slotine Adaptive Controller for the Robot Motion System in the Real-Time Domain, Controlling the Motion Trajectory in Three-Dimensional Space with Three Robot Joints

1) Control Object

The detailed installation drawings of the control object are shown in Figure 4 and the specific parameters are shown in Table II.

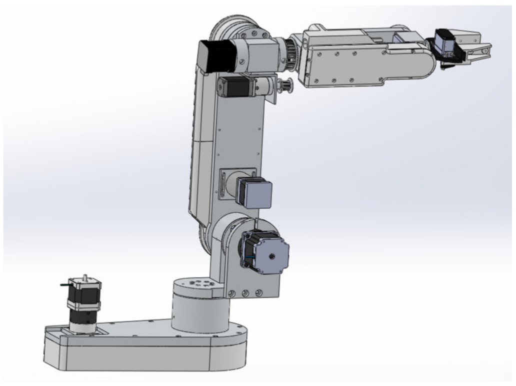

Fig. 4. Detailed assembly drawing of the control object.

TABLE II. SPECIFIC PARAMETERS OF THE CONTROL OBJECT

Robot	Swivel angle	Size (mm)
Shoulder	Fold: 180° External rotation: 90° Internal rotation: 70° Format: 180° Close: 45°; Stretch: 45°	220
Elbow joint	Stretching: 0°; Fold: 140°	250
Wrist joint	Stretch: 70°; Fold: 80°-90° Tilt and rotate: 20°; Pillar tilt: 35°	

2) Experimental Configuration

The main components in the system's experimental configuration include the FlexMotion-6C digital signal processing card, inverter system-servo motor, computer, and robot. The sequence of experimental steps is as follows:

1. Connect the servo motors of the Almega 16 robot to the corresponding inverters in the inverter cabinet and control them according to the diagram shown in Figure 5.
2. Connect the encoders attached to the servo motor to the inverter and from the inverter to the FlexMotion-6C control card via the UMI intermediate block.
3. Start the computer and the measurement & automation software to enter the control program and set the spatial trajectory of the robot's clamp movement block.
4. Adjust the forward, reverse (limit) and zero-point (home) switches according to the spatial trajectory.
5. Call the control interface to select the mode and parameters of rotation angle, velocity, acceleration, and deceleration

for the robot component movements according to the data contained in the measurement & automation software and FlexMotion-6C card, extrapolated from the above space orbit.

6. Create a measurement interface to store images and data on space orbits, orbital errors, and transit times.

The motion control system with an adaptive controller (Figure 5) compensates for factors such as gravity, friction, and external noise to ensure accurate torque control for each robot joint. This precise control aligns the actual joint angle *q(t)* with the desired angle, eliminating discrepancies and enabling the robot to follow a defined trajectory. The system's structure involves control software in the computer to set motion parameters, process data, and manage orbits. At the same time, the FlexMotion-6C card performs orbit interpolation and sets parameters for the three robot axes. Commands from the computer are executed concurrently. The adaptive controller is configured for the three axes using data files with a *.dll extension written in C. Sustainable adaptive controllers are used, with torque feedback provided by encoders and feedback loops managed by OMNUC inverters. These inverters handle the torque control loops corresponding to the axes, ensuring precise spatial trajectories. Key benefits of this system include synchronized axis operation, simultaneous start/stop for coordinated motion, and automatic parameter scaling for accurate movement within vector space. In the event of a problem such as overload, signal loss, or travel limit detection, all axes halt simultaneously, and the event is reported to the computer. The documentation and instructions provided allow configuration of parameter settings for both the adaptive controller and the inverters.

3) Actual Results

With the parameter set: $K_{d_1}=1100$, $K_{d_2}=2100$, $K_{d_3}=2100$, $\Gamma_1=285$, $\Gamma_2=282$, $\Gamma_3=285$ the results depicted in Figure 6 are obtained. The placement trajectory and the response trajectory of the robot clamp have minimal deviations (0.2.10 $^{-3}$) and the system transient time is small (t_{qd} = 452 ms). The average position error value of all three joints when using the Li-Slotine adaptive controller for three joints is minimal (0.1 %). The position error graph of each joint (joint 1, joint 2, joint 3) shows that the system fluctuates only slightly when it is stable. The application of the Li-Slotine adaptive control algorithm ensures accurate trajectory tracking quality without depending on the uncertain constant parameters of the dynamic model and the impact of interchanged components between joint axes. The Li-Slotine adaptive control law solves this problem by adjusting the uncertain parameters using the update rule (19) to compare the adjusted value with the actual value and input it to the controller. The control law is constructed based on Lyapunov stability criteria to ensure that the system is stable and the position errors of the rotating joints converge to zero when calculating the control torque τ_{dk} for each joint. The Li-Slotine adaptive control algorithm, which is suitable for controlling multi-degree-of-freedom robot motion systems, suffers from the disadvantage of extensive computation and the requirement of specific fundamental knowledge. However, modern high-speed microprocessors can meet the specified criteria. Even if the mathematical model is precise enough to assist us develop a conventional controller that achieves an acceptable level of quality, changes in the structure and architecture of the materials used in the manufacture of the control devices and the actuator will inevitably occur over a long period of work. As a result, the inadequacy of the initial mathematical model used to represent the object leads to a loss of accuracy, thereby reducing the control effectiveness of the conventional controller.

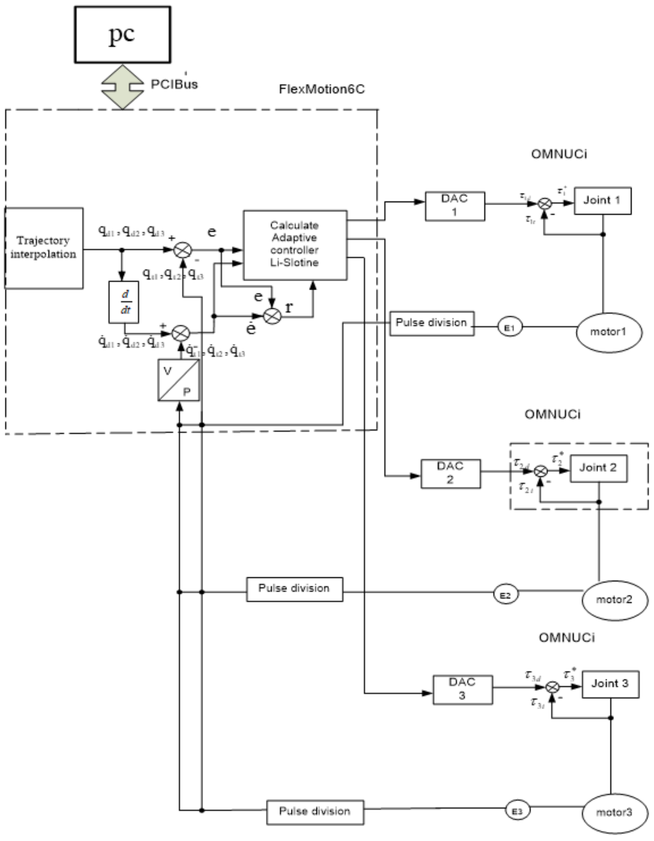

Fig. 5. Li-Slotine adaptive controller structure.

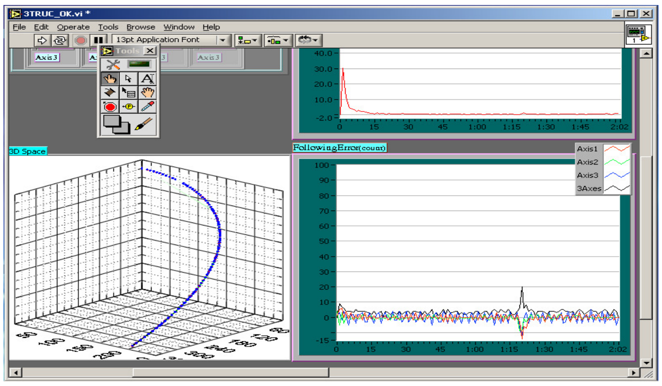

Fig. 6. Control response when using Li-Slotine adaptive controller for 3 robot joints.

III. QUALITY OF THE ROBOT TRANSFER CONTROL SYSTEM MEASURED BY AN ITERATIVE LEARNING METHOD

Without using a mathematical model of the robot, the output structure control consists of two loop controls. The first control loop (inner loop) is an intelligent component recognition unit independent of the function μ by $\hat{\mu}$ using the Taylor series analysis method to transform (1) into a linear system at the input (22) using the compensation control method, as shown in Figure 7. The second control loop (outer loop) is a learning controller that determines the signal U that makes the appropriate variable path q firmly follow the given sample path R . This learning controller uses the P-type function with the learning function parameter K defined online after each K-th trial according to the principle of minimizing the sum of squared flooding errors [8].

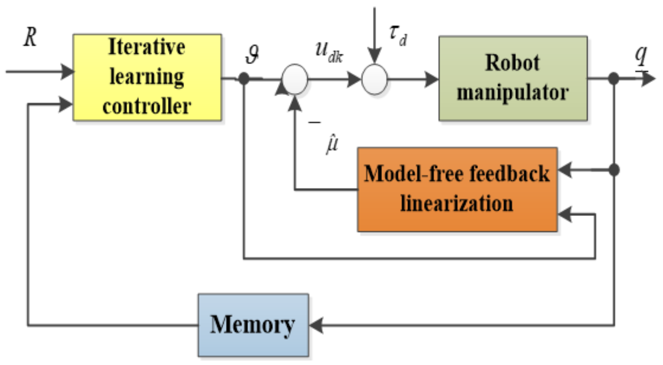

Fig. 7. Structure diagram for robot control.

The task of this intelligent linearization controller is to modify the original robot system (21) in the following ways:

$$u_{dk}+f_d = M(q,\theta)\ddot{q}+H(q,\dot{q},\theta)\dot{q}+G(q,\theta)+f(\dot{q},\theta) \quad (21)$$

Thus, (21) can be represented as (22) in a way that corresponds to the total noise, including the initial noise and the model line error:

$$\ddot{q}=-A_1q-A_2\dot{q}+U_{dk}+\mu \quad (22)$$

where A_1, A_2 are two randomly generated matrices, and:

$$\mu=f_d+\left[I_n-M(q,\theta)\right]\ddot{q}-\left[H(q,\dot{q},\theta)-A_2\right]\dot{q} -\left[G(q,\theta)-A_1q\right] \quad (23)$$

The estimation errors for the new unknown function vector are updated, as shown below:

$$U_{dk}=\vartheta-\hat{\mu} \quad (24)$$

with the estimation of bias λ as (25), system (22) will become linear:

$$\dot{X}=A*X+B\left[\vartheta+\lambda\right],Y=q\left(I_n,O_n\right)*X \quad (25)$$

with:

$$\dot{X}=\begin{pmatrix}q\\ \dot{q}\end{pmatrix},A=\begin{pmatrix}O_n & I_n\\ -A_1 & -A_2\end{pmatrix},B=\begin{pmatrix}O_n\\ I_n\end{pmatrix},\lambda=\mu-\hat{\mu} \quad (26)$$

The compensation system in (25) should be regulated so that its result $Y(t)$ is close to the desired reference value $R(t)$. Linearized feedback allows for model-free noise correction in the inner control loop. Equation (24) and design as in the present case with $t=k*T+\tau;\,0\le\tau<T$ and $\vartheta_k=\vartheta_k(\tau)$, $X_k=X_k(\tau)$, (25) is rewritten as:

$$\dot{X}_k(\tau)=AX_k(\tau)+B\left[\vartheta_k(\tau)-\hat{\mu}_k(\tau)+\mu_k(\tau)\right] \quad (27)$$

The automatic model-free feedback linearization block (Figure 7) aims to estimate $\hat{\mu}$ based on the measured values two times $\tau=nT_s$ and $t\text{-}T_s=T_s(n\text{-}1)$ for each corresponding $x_k(nT_s)$; $X_k(n-1)T_s$ value. With $0<T_s<<1$ being an arbitrarily chosen constant small value, we expand the Taylor function $x_k(\tau)$ of the function around $(n-1)T_s$ as follows:

$$X_k\left((n-1)T_s\right)= \frac{T_s^2}{2}\ddot{X}_k(\varsigma)-T_s\dot{X}_k(nT_s)+X_k(nT_s) \quad (28)$$

where:

$$(n\text{-}1)t_s\le\varsigma\le nt_s$$

or:

$$\dot{X}_k(n*T_s)\approx\frac{X_k*(n*T_s)-X_k*\left((n-1)*T_s\right)}{T_s} \quad (29)$$

Assuming that the last term of (28) is very small and can be ignored, the expression $\hat{\mu}_k*(nT_s)$ will be used to approximate equation (29) by replacing the signs "$\approx$" and $\mu_k*(nT_s)$ in (29) with "$=$" and $\hat{\mu}_k*(nT_s)$:

$$\frac{X_k(n*T_s)-X_k*\left((n-1)*T_s\right)}{T_s}= AX_k(n*T_s)+ B\left[\vartheta_k(n*T_s)-\hat{\mu}_k*\left((n-1)*T_s\right)+\mu_k*(n*T_s)\right] \quad (30)$$

From there, calculate:

$$B*\hat{\mu}_k*(n*T_s)=\frac{X_k*(n*T_s)-X_k*\left((n-1)*T_s\right)}{T_s} -A*X_k*(n*T_s)-B*\left[\vartheta_k*(n*T_s)-\hat{\mu}_{k*}\left((n-1)*T_s\right)\right]$$

$$\Rightarrow\hat{\mu}_k*(n*T_s)= B^T*\left[\frac{X_k*(n*T_s)-X_k*\left((n-1)*T_s\right)}{T_s}-A*X_k(n*T_s)\right] -\left[\vartheta_k*(n*T_s)-\hat{\mu}_k*\left((n-1)*T_s\right)\right] \quad (31)$$

Theorem 1: The computed value $\hat{\mu}_k*(n*t_s)$ in (31) reduces the approximation error in (30).

Proof: Denote the error of both sides of (30) with:

$$\sigma = A*X_k(n*T_s) + B*\left[\vartheta_k*(n*T_s)-\hat{\mu}_k\left((n-1)*T_s\right)+\hat{\mu}_k*(n*T_s)\right] - \frac{X_k*(n*T_s)-X_k*\left((n-1)*T_s\right)}{T_s} = B*\mu_k*(n*T_s)+\eta$$

where:

$$\eta = A*X_k(nT_s)+B*\left[\vartheta_{k*}(n*T_s)-\hat{\mu}_k*\left((n-1)*T_s\right)\right] - \frac{X_k*(n*T_s)-X_k*\left((n-1)*T_s\right)}{T_s}$$

The optimization problem is therefore formulated as:

$$\mu^* = \arg\min_{\mu_k}\|\varepsilon_k\|^2 = \arg\min_{\mu_k}\|B*\mu_k+\gamma\|^2 = \arg\min_{\mu_k}\|B*\mu_k+\gamma\|^T\|B*\mu_k+\gamma\| = \arg\min_{\mu_k}\left\|\mu_k^T*\mu_k+\underline{2*\gamma^T*B*\mu_k+\gamma^T*\gamma}\right\|^T$$

and has a unique solution $\mu^* = -B^T*\underline{\gamma}$ which coincides with $\hat{\mu}_k*(n*T_s)$ given in (31).

According to (31), this estimation method, which is designed to account for disturbance and uncertainty in a vector model, doesn't employ the computational framework (21). As a result, the noise compensation (24) with $\hat{\mu}$ calculated from (31) will be model-free. For iterative learning systems, it is rewritten in ILC language as follows:

$$\begin{cases} X_k(n+1)=\hat{A}*X_k(n)+\hat{B}\left[\vartheta_k(i)+\delta_k(n)\right] \\ Y_k(n)=\hat{C}*X_k(n) \end{cases} \quad (32)$$

where:

$$n=0,1,.....,N=T/t_s, X_k(N)=X_{k+1}(0)$$

and:

$$\hat{A}=\exp(At_s); \hat{B}=\int_0^{T_s}e^{At}Bdt\ ; \hat{C}=(I_n,O_n) \quad (33)$$

The control objective at this point is to choose the proper learning parameter *K* for the PD-Type update rule:

$$\vartheta_{k+1}(n)=\vartheta_k(n)+K*E_k \quad (34)$$

with $E_k(n)=R_k(n)+Y_k(n)$, to achieve the requisite agreement $\|E_k(n)\|\to 0$ for all *n*, or at least as close to the origin as possible.

Given that ILC (34) is partially constant, the obtained discrete-time model in (36) is equivalent to the continuous-time model in (25). Therefore, (32) can be applied to all robot controllers. From (32), we obtain with the assumption $\delta_k(n)=0$:

$$Y_{k+1}(n)=C*\hat{A}^n*X_{k+1}(0)+\sum_{j=0}^{n-1}C*\hat{A}^{n-n-1}\hat{B}*\vartheta_{k+1}(j)$$

Form (32) and the apparent display of repeatable competence $X_k(0)=X_{k+1}(0),\forall \text{k}$:

$$\begin{aligned} E_{k+1}(n) &= R*(n)-Y_{k+1}(n)=R(n)- \\ &\left[C*\hat{A}^nX_k(0)+\sum_{j=0}^{n-1}C*\hat{A}^{n-n-1}\hat{B}*(\vartheta(j)+K*E_k(j))\right] \\ &= R(n)-Y_k(n)-\sum_{j=0}^{n-1}C*\hat{A}^{n-j-1}\hat{B}*K*E_k(j) \\ &= (I-C*\hat{B}*K)*E_k(n)-\sum_{j=0}^{n-2}C*\hat{A}^{n-j-1}\hat{B}*K*E_k(j) \end{aligned}$$

Hence:

$$E_{k+1}=\Phi*E_k \quad (35)$$

where:

$$E_k=\begin{pmatrix} E_k(0) \\ E_k(1) \\ \vdots \\ E_k(N-1) \end{pmatrix}$$
$$\Phi=\begin{pmatrix} I-C\hat{B}K & 0 & \cdots & 0 \\ -C\hat{A}\hat{B}K & I-C\hat{B}K & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ C\hat{A}^{N-2}\hat{B}K & -C\hat{A}^{N-2}\hat{B}K & \cdots & I-C\hat{B}K \end{pmatrix} \quad (36)$$

Theorem 2: For circumstance $\delta_k(n)=0$, the needs $\|E_k(j)\|\to 0$ for all $n=0,1,.....,N-1$ will be fulfilled if and only if the P-Type training variable *K* is modified to produce the value provided in (32) Schur.

A. Performance and Control Algorithms for a Closed-Loop System.

The following method implements the suggested model-free controller (24) by obtaining the ϑ, μ values from (35). In this type of control technique, each cycle tests the robot work time *T* many times

Theorem 3: The proposed model-free control architecture in Figure 1 includes feedback to the linearity block via the noise compensator (24), (31), and the ILC block (34). If *d* is restricted and continuous, this pushes the robot manipulators' (21) output error in tracking $|E_k(\tau)|$ to a dependent neighbourhood *O* of origin. If a less significant option is picked, the *O* outcome will be small.

Proof: f_d is continuous and bounded, and the total noise μ and components δ are also constant and bounded. The upper limit of O is Δ, and this limit can be reduced arbitrarily by reducing t_s according to Theorem 1.

Therefore, the correctness of Theorem 2 is determined:

$$\dot{\omega} = A\omega + B\vartheta \text{ with } \omega = vec(R, \dot{R}) \tag{37}$$

Subtracting (37) from (25) gets:

$$\dot{\sigma} = A\sigma - B\delta \text{ where } \sigma = vec(E, \dot{E}) \tag{38}$$

Select a function using Lyapunov's theory:

$$V(\sigma) = \sigma^T P \sigma$$

$$\begin{aligned} \Rightarrow \dot{V} &= (A*\sigma - B*\delta)^T + \sigma^T * P(A*\sigma - B*\delta) \\ &= \sigma^T * (A^T * P + P*A)\sigma - 2*\sigma^T P*B*\delta \\ &= -\sigma^T * (-A^T * P - P*A)*\sigma - 2*\sigma^T * P*B*\delta \\ &=\leq -\gamma_{\max} * (-A^T * P - P*A)*\|\sigma\|^2 + 2*\|P*B\|\Delta*\|\varepsilon\| \\ &= \|\varepsilon\| * \left[-\gamma_{\max}(-A^T * P - P*A)*\|\sigma\| + 2*\|P*B\|\Delta\right] \end{aligned} \tag{39}$$

With $\gamma_{\max}(-A^T * P - P*A) \Rightarrow \dot{V}(\varepsilon) < 0$ as long as:

$$2*\|P*B\|*\Delta < \gamma_{\max}(-A^T * P - P*A)\|\varepsilon\|$$

$$\text{or } \frac{2\|P*B\|*\Delta}{\gamma_{\max *}(-A^T * P - P*A)} < \|\varepsilon\| \tag{40}$$

$$\Rightarrow O = \left\{ \sigma \in R^{2n} * \middle| \|\sigma\| \leq \frac{2*\|P*B\|\Delta}{\lambda_{\max}(-A^T * P - P*A)} \right\} \tag{41}$$

The control algorithm is as follows:

```
Choose 2 matrices $A_1$, $A_2$ in (33) to become Hurwitz
Calculate $\widehat{A}$, $\widehat{B}$, $\widehat{C}$ in (33) and $\Phi$ in (36.
Find the estimated matrices $\widehat{A}$, $\widehat{B}$, $\widehat{C}$ in (33) and $\Phi$ in (36)
Select $0 < T_s << 1$. Calculate $S = T/T_s$
Select $D_1^T = 1/T_s$; $D_0^T = -D_1^T$ in (31)
Select learning $\hat{\mu}$ and $E_0$
Allocate $J(n) = R(n)$ with $n = 0,1,...,S-1$ and $Z = 0$.
Identify learning parameter $K$ so that $\Phi$ of (33) becomes Schur.
While continue the control do
   For $n = 0,1,...,S-1$ do
      $u_{dk} = A_1 q - A_2 \dot{q} - \hat{\mu}$
      $X = vec(E, \dot{E})$, $Y(n) = q$
      $\hat{\mu}_k \leftarrow B^T \left[\frac{X - Z}{t_s} - A * X\right] - (\vartheta(n) - \hat{\mu})$
      Set $Z \leftarrow X$
   End for
   $\vartheta = vec(\vartheta(0), ....., \vartheta(N-1))$,
   $E = vec(E(0), ....., E(N-1))$
   $K = \arg\min_{a \le K \le b} \|(I - \Phi * K) * E\|$
   $\vartheta \leftarrow \vartheta + K * E$;   $E_0 \leftarrow E$;
End while
```

To demonstrate the computation of the algorithm, it is properly applied to robot control as follows:

With T_s=0.02 s , $n = 1 \div 14$ are random, $T = 10$ s ; $l_1 = 0.35$, $l_2 = 0.55$, $l_3 = 0.6$, $g = 9.81$

$$R_1(t) = \sin(\pi t/T) + 0.3\sin(3.2\pi t/T); R_2(t) = \sin(3\pi t/T)$$

$$R_3(t) = 2.5\sin(2\pi t/T) - 0.2\sin(3\pi t/T)$$

$$A_1 = \begin{bmatrix} 31 & 0 & 0 \\ 0 & 12 & 0 \\ 0 & 0 & 12 \end{bmatrix}; A_2 = \begin{bmatrix} 12 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 10 \end{bmatrix}; K = \begin{bmatrix} 0.8 & 0 & 0 \\ 0 & 0.3 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$$

and the results are shown in Figures 8 through 10.

Figures 8 to 10 show the positions of the three joints q_1, q_2 and q_3 and after 20 and 350 trials, respectively, when the iterative learning controller uses the $K = \arg\min_{a \le K \le b} \|(I - \Phi * K) * E\|$ formula for the "intelligent" determination of the learning function $K = diag(K_k^j)$. The highest value of the tracking error over the whole working duration in about 350 trials is $\max|E_{350}(1)| \approx 0.025$ for the first joint variable, $\max|E_{350}(2)| \approx 0.001$ and $\max|E_{350}(3)| \approx 0.022$ for the second and third joint variables, respectively.

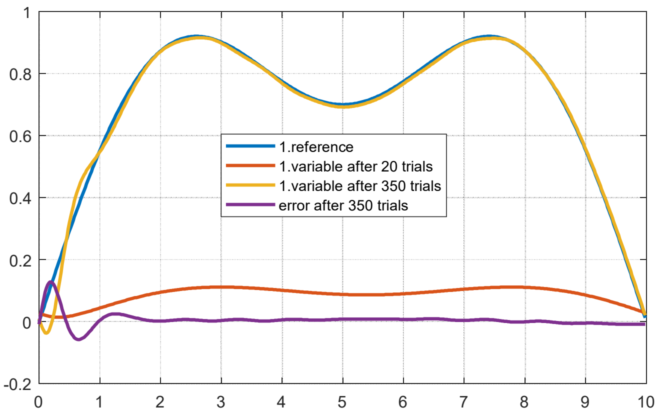

Fig. 8. Representation of the position of the first variable after 20 and 350 trials.

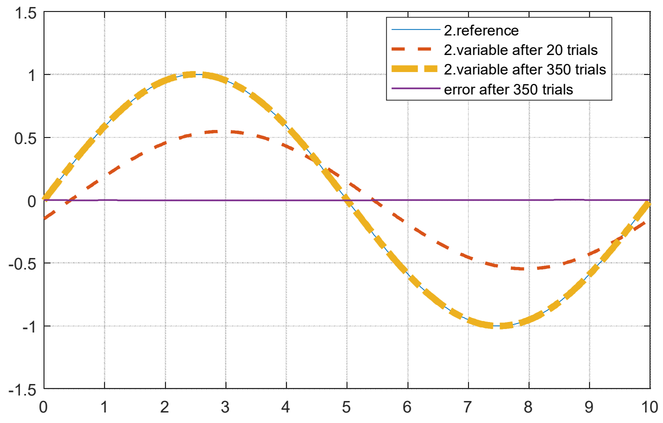

Fig. 9. Representation of the position of the second variable after 20 and 350 trials.

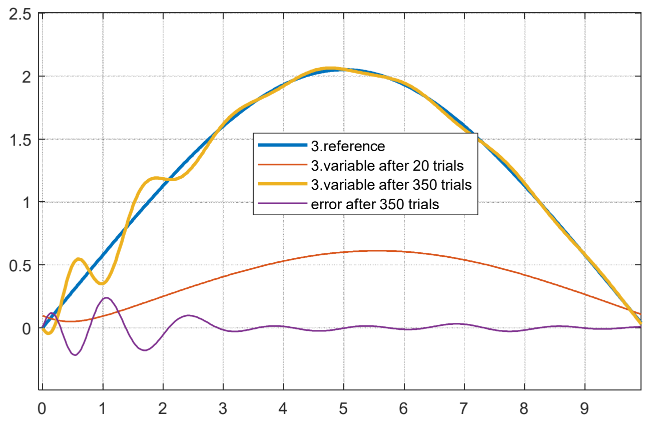

Fig. 10. Representation of the position of the third variable after 20 and 350 trials.

IV. CONCLUSION

The article discusses the implementation of the Li-Slotine adaptive control and the Iterative Learning Control (ILC) as an appropriate approach for the robot motion system, which involves uncertain model parameters and experiences undesirable interactions between joint axes. The primary findings are presented based on simulation and experimental outcomes. The recent theoretical study and experimental results indicate that the effectiveness of the Li-Slotine automated control method relies on the mathematical model's accuracy level. This enables us to develop a conventional controller to successfully achieve the desired results. Following an extended work period, the desired quality will alter the physical composition of the materials used to manufacture the controller and the actuator. As a result, the inadequacy of the initial mathematical model used to represent the object leads to a loss of accuracy, which degrades the control quality of the traditional controller. An intelligent control approach for 3-degree-of-freedom robot motion systems that utilizes adaptive iterative learning and can operate with partial or no mathematical models, requires further refinement before it can be effectively implemented in practical applications. This outcome demonstrates that the ILC has provided the necessary tracking quality, and after 350 trials, the output signal has accurately tracked the reference signal across all three joint axes through the adaptive adjustment of two learning parameters, termed "smart". The following research direction will focus on improving the self-learning capabilities by combining ILC with deep learning or reinforcement learning algorithms so that robots can learn faster and more effectively from repeated data.

ACKNOWLEDGMENT

This research is funded by Hanoi University of Science and Technology under project number T2023-TĐ-004.

REFERENCES

[1] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot Manipulator Control: Theory and Practice*, 2nd ed. New York, NY, USA: Marcel Dekker, Inc., 2004.

[2] A. Abdessameud and M. Khelfi, "A variable structure observer for the control of robot manipulators," *International Journal of Applied Mathematics and Computer Science*, vol. 16, no. 2, pp. 189–196, 2006.

[3] W. Yu and J. A. Heredia, "PD control of robot with RBF networks compensation," in *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, 2000, vol. 5, pp. 329–334, https://doi.org/10.1109/IJCNN.2000.861488.

[4] B. Yao, "Adaptive Robust Control of Robot Manipulators: Theory and Comparative Experiments," in *Proceedings of the 2nd Chinese World Congress on Intelligent Control and Intelligent Automation*, Xian, China, 1997, pp. 442–447.

[5] J.-J. E. Slotine and J. A. Coetsee, "Adaptive sliding controller synthesis for non-linear systems," *International Journal of Control*, vol. 43, no. 6, pp. 1631–1651, Jun. 1986, https://doi.org/10.1080/00207178608933564.

[6] K. Narendra and A. Annaswamy, "Robust adaptive control in the presence of bounded disturbances," *IEEE Transactions on Automatic Control*, vol. 31, no. 4, pp. 306–315, Apr. 1986, https://doi.org/10.1109/TAC.1986.1104259.

[7] H. Medjoubi, A. Yassine, and H. 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, Apr. 2021, https://doi.org/10.48084/etasr.3950.

[8] C. T. Trung, "Improving the quality of traditional control systems using iterative learning method," Ph.D. dissertation, Hanoi University of Science and Technology, Hanoi, Vietnam.

[9] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, Jun. 2006, https://doi.org/10.1109/MCS.2006.1636313.

[10] M. Norrlöf, "Iterative Learning Control: Analysis, Design, and Experiments," Ph.D. dissertation, Department of Electrical Engineering, Automatic Control, Linköping University, Linköping, Sweden, 2000.

[11] B. Kasmi and A. 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, Apr. 2021, https://doi.org/10.48084/etasr.4031.

[12] F. Bouakrif, D. Boukhetala, and F. Boudjema, "Velocity observer-based iterative learning control for robot manipulators," *International Journal of Systems Science*, vol. 44, no. 2, pp. 214–222, Feb. 2013, https://doi.org/10.1080/00207721.2011.600467.

[13] K. G. Tran, N. H. Nguyen, and P. D. Nguyen, "Observer-Based Controllers for Two-Wheeled Inverted Robots with Unknown Input Disturbance and Model Uncertainty," *Journal of Control Science and Engineering*, vol. 2020, no. 1, Jun. 2020, Art. no. 7205737, https://doi.org/10.1155/2020/7205737.

[14] T. T. Cao, P. D. Nguyen, N. H. Nguyen, and H. T. Nguyen, "An indirect iterative learning controller for nonlinear systems with mismatched uncertainties and matched disturbances," *International Journal of*

Systems Science, vol. 53, no. 16, pp. 3375–3389, Dec. 2022, https://doi.org/10.1080/00207721.2022.2083259.

[15] 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*, vol. 9, no. 9, May 2019, Art. no. 1753, https://doi.org/10.3390/app9091753.

[16] P. D. Nguyen and N. H. Nguyen, "Adaptive control for nonlinear non-autonomous systems with unknown input disturbance," *International Journal of Control*, vol. 95, no. 12, pp. 3416–3426, Dec. 2022, https://doi.org/10.1080/00207179.2021.1974571.

[17] L. Fortuna and A. Buscarino, "Microrobots in Micromachines," *Micromachines*, vol. 13, no. 8, Aug. 2022, Art. no. 1207, https://doi.org/10.3390/mi13081207.

[18] M. Bucolo, A. Buscarino, L. Fortuna, and M. Frasca, "Forward Action to Stabilize multiple Time-Delays MIMO Systems," *International Journal of Dynamics and Control*, vol. 12, no. 4, pp. 1157–1165, Apr. 2024, https://doi.org/10.1007/s40435-023-01221-6.

[19] D. M. Do, D. Hoang, N. H. Nguyen, and P. D. Nguyen, "Data-Driven Output Regulation of Uncertain 6 DOF AUV via Lagrange Interpolation," in *2022 11th International Conference on Control, Automation and Information Sciences*, Hanoi, Vietnam, 2022, pp. 73–78, https://doi.org/10.1109/ICCAIS56082.2022.9990560.

[20] V. T. Ha, "Some control solutions to improve the quality of motion of industrial manipulators," Ph.D. dissertation, Hanoi University of Science and Technology, Hanoi, Vietnam, 2012.

AUTHORS PROFILE

Vo Thu Ha received a B.S degree in Control and Automation Engineering from Thai Nguyen University of Technology, Vietnam in 2002, the Master's degree from Hanoi University of Science and Technology, Vietnam in 2004, and the Ph. D from Hanoi University of Science and Technology, Vietnam in 2012. She received the Assoc. Prof. degree in automation engineering from the University of Economics - Technology for Industries in 11/2017. She has worked in the Faculty of Electrical Engineering, University of Economics - Technology for Industries since 2003, Vietnam. Assoc. Prof Vo Thu Ha's research includes robot control, electrical drive, power electronics, modeling and simulation. She can be contacted at vtha@uneti.edu.vn.

Trung T. Cao was born in Vietnam in 1978. He received the B.Eng., the M.S degrees, and PhD. degree in automatic control from Hanoi University of Science and Technology, Hanoi, in 2001, 2010 and 2024, respectively. Since 2002, he has been a lecturer at School of Electrical and Electronic Engineering, Hanoi University of Science and Technology. His research interests include fault detection and isolation, optimization, and optimal control. He can be contacted at trung.caothanh@hust.edu.vn.

Than Thi Thuong received a Control and Automation Engineering degree from the University of Economics - Technology for Industries, Vietnam in 2011, and a Master's degree from the University of Transport and Communications, Vietnam in 2016. She has worked in the Faculty of Electrical Engineering, University of Economics - Technology for Industries since 2018, Vietnam. Than Thi Thuong's research includes robot control, process control, modeling, and simulation. She can be contacted at ttthuong.dien@uneti.edu.vn.

Nguyen Thi Thanh received a B.S. degree in Control and Automation Engineering from the University of Economics - Technology for Industries, Vietnam, in 2012 and a Master's degree from Hanoi University of Science and Technology, Vietnam, in 2016. She has worked in the Faculty of Electrical Engineering, University of Economics - Technology for Industries, Vietnam since 2012. Her research areas are electrical drive, robot control, power electronics, modeling, and simulation. She can be contacted at ntthanh.ddt@uneti.edu.vn.

Bui Thi Hong Diem received a Control and Automation Engineering degree from the University of Economics - Technology for Industries, Vietnam in 2019. She has worked in the Faculty of Electrical Engineering, University of Economics - Technology for Industries, Vietnam since 2023. Bui Thi Hong Diem's research includes robot control. She can be contacted at bthdiem@uneti.edu.vn.