Vietnam Journal of Science and Technology ***61*** *(1) (2023) 122-132*

doi:10.15625/2525-2518/16840

A model-free controller for uncertain robot manipulators with matched disturbances

Cao Thanh Trung*, Nguyen Hoai Nam, Nguyen Doan Phuoc

Hanoi University of Science and Technology, No.1 Dai Co Viet, Hai Ba Trung, Ha Noi, Viet Nam

*Email: *trung.caothanh@hust.edu.vn*

Received: 23 December 2021; Accepted for publication: 14 January 2023

Abstract. Precise control is critical for robots, but it is difficult to obtain an accurate dynamic model of the robot due to the presence of modeling errors and uncertainties in the complex working environment, resulting in decreased control performances. The article proposes a method for designing a digital controller for output tracking of disturbed repetitive robot manipulators that does not require a mathematical model of the controlled robots, with the goal of improving tracking accuracy. This controller consists of two separate intelligent parts. The first part aims to stabilize the original robot manipulators via a model-free state feedback linearization technique. The suggested model-free state feedback linearization technique does not make use of the original Euler-Lagrange model of the robot. The second part will then employ the concept of iterative learning control to asymptotically drive the obtained stable linear system to desired references. Both of these parts use only the robot's measured data from the past for carrying out their tasks instead of robot models. Moreover, the proposed controller is structurally simple and computationally efficient. Finally, to validate the theoretical results, a simulation verification on a 2-degree-of-freedom (DOF) uncertain planar robot is performed, and the results show that excellent tracking performance is feasible.

Keywords: intelligent feedback linearization, iterative learning control, disturbance compensation.

Classification numbers: 4.10, 5.3, 5.8

1. INTRODUCTION

Robotic manipulators are widely used in industries to perform repetitive tasks. They can carry heavy payloads, work faster, more accurately and smarter than humans. In addition, the application of robotic manipulators increases the productivity and quality of products to a greater extent. Therefore, many control methods have been established and applied to them to ensure that they will achieve the required accuracy.

At the beginning, the research mainly focused on model-based control methods [1 - 7]. It means that the design of controller was established mainly based on the following system of mathematical equations of robotics (also called the Euler-Lagrange model):

$$\underline{u}+\underline{d}=M(\underline{q},\underline{\theta})\underline{\ddot{q}}+C(\underline{q},\underline{\dot{q}},\underline{\theta})\underline{\dot{q}}+\underline{g}(\underline{q},\underline{\theta}) \tag{1}$$

where $\underline{u}=(u_1,\ \dots\ ,u_n)^T$ is an n dimensional vector of control inputs (torques), $\underline{d}$ is a vector of the same dimension which includes all unknown viscous friction torques and exogenous disturbances affecting the robot manipulator's performance, $M(\underline{q},\underline{\theta})$ is an $n\times n$ symmetric positive definite inertia matrix, $\underline{q}=(q_1,\ \dots\ ,q_n)^T$ is a vector of n joint variables, the vector $\underline{\theta}$ contains all uncertain parameters of robot models, $C(\underline{q},\underline{\dot{q}},\underline{\theta})\underline{\dot{q}}$ is a vector of Coriolis and centrifugal terms, $\underline{g}(\underline{q},\underline{\theta})$ is an n dimensional vector of gravitational torques. All the above existing model-based control methods are classified into different types depending on whether $\underline{\theta}$ and $\underline{d}$ can be determined.

If both $\underline{\theta}$ and $\underline{d}$ can be determined exactly, the feedback linearization method is applicable [1 - 3]. However, the assumption that $\underline{\theta}$ and $\underline{d}$ are known is difficult to satisfy in the real world, so adaptive methods are preferred. For the case where $\underline{\theta}$ is unknown, but $\underline{d}$ is negligible, certainty-equivalent and Li-Slotine methods are suitable [1, 4, 5]. In the most practical scenario that both $\underline{\theta}$ and $\underline{d}$ are unknown, the sliding mode control (SMC) technique [6, 7] is an appropriate one. However, due to the inherent chattering characteristic of SMC, the main obstacle to implementing this technique in practice is that the actuators have to switch their value signs with a very high frequency on the sliding surface, which causes premature failures in the whole system.

This analysis of the disadvantages of the model based control methods has pointed out that all difficulties of conventional control methodologies can be overcome by using supplementary intelligent control techniques as a separate part of the control system, such as fuzzy control [8], neural network-based methods [9, 10], and iterative learning control (ILC) [11, 12].

ILC is known as an effective output tracking control concept for repetitive controlled systems with a fixed working period T [13]. It utilizes system tracking errors recorded along the past working period $(k-1)T\le t<kT$ to refine the system inputs $\underline{u}(t)$ in order to reduce its output tracking errors during the next working period $kT\le t<(k+1)T$ based on an update law. According to the ILC concept, the input and output $\underline{u}(t),\ \underline{y}(t)$ are often symbolized with $\underline{u}_k(\tau),\ \underline{y}_k(\tau)$, respectively, where k indicates the current working period, also known as trial, and $0\le\tau<T$ is a time instant among this trial. Essential researches on the ILC concept have focused mainly on how to determine an appropriate update law and its parameters for a particular controlled system, which guarantees the convergence of the system tracking error $\underline{e}_k(\tau)=r(\tau)-\underline{y}_k(\tau)$ in the sense of $\lim_{k\to\infty}\|\underline{e}_k(\tau)\|=0$.

Different from the works [11, 12], where nonlinear update laws are employed and the convergence is not analyzed in detail, this article will show that the application of a supplemental model-free feedback linearization controller using the ILC concept with the basic linear P-Type update law for tracking control of robot manipulators is possible. Furthermore, a sufficient condition for required convergence $\|\underline{e}_k(\tau)\|\to 0$ is also given.

The article is organized as follows. First, Section 2 provides the main theoretical results with a complete control algorithm in it. Then, a numerical example with an uncertain 3 DOF robot is illustrated in Section 3. Finally, conclusions and future works are presented at Section 4.

2. MAIN RESULTS

2.1. Control framework

It is obvious that the Euler-Lagrange model (1) of robot manipulators is equivalent to

$$\underline{\ddot{q}} = -A_1\underline{q} - A_2\underline{\dot{q}} + \underline{u} + \underline{\eta} \tag{2}$$

where A_1, A_2 are two arbitrarily chosen matrices and

$$\underline{\eta} = \underline{d} + \left[I_n - M(\underline{q},\underline{\theta})\right]\underline{\ddot{q}} - \left[C(\underline{q},\underline{\dot{q}},\underline{\theta}) - A_2\right]\underline{\dot{q}} - \left[\underline{g}(\underline{q},\underline{\theta}) - A_1\underline{q}\right] \tag{3}$$

is an unknown function vector, which consists of matched disturbances and model uncertainties. Moreover, this new unknown function vector $\underline{\eta}$ also contains the nonlinearity of the original model (1). Based on the obtained equivalent model (2), an intelligent control framework for output tracking controlled robot manipulators (1) can be suggested correspondingly as follows:

– First, the new unknown function vector $\underline{\eta}$ will be compensated by its estimated value $\underline{\hat{\eta}}$. If this is already done, then by using the compensator

$$\underline{u} = \underline{v} - \underline{\hat{\eta}} \tag{4}$$

the system (2) becomes linear with a small remaining estimation error $\underline{\delta}$ as below

$$\underline{\dot{x}} = A\underline{x} + B\left[\underline{v} + \underline{\delta}\right],\ \underline{y} = \underline{q} = \left(I_n\ ,\ \mathbf{0}_n\right)\underline{x} \tag{5}$$

where

$$\underline{x} = \begin{pmatrix} \underline{q} \\ \underline{\dot{q}} \end{pmatrix},\ A = \begin{pmatrix} \mathbf{0}_n & I_n \\ -A_1 & -A_2 \end{pmatrix},\ B = \begin{pmatrix} \mathbf{0}_n \\ I_n \end{pmatrix},\ \underline{\delta} = \underline{\eta} - \underline{\hat{\eta}} \tag{6}$$

and $\mathbf{0}_n$, I_n are the zeros and the identity matrix of dimension $n \times n$, respectively.

Note that for holding the intelligent ability of the obtained output tracking controller later, the created estimator $\underline{\hat{\eta}} \approx \underline{\eta}$ must be model-free. It means that this estimator could only use the measured data $\underline{x}$ of robots to calculate $\underline{\hat{\eta}}$ for its action, not the model (1).

– Secondly, the compensated system (5) will be controlled so that its output $\underline{y}(t)$ converges to a desired reference $\underline{r}(t)$. For this purpose, the ILC concept is utilized.

Figure 1 below illustrates the aforementioned control framework.

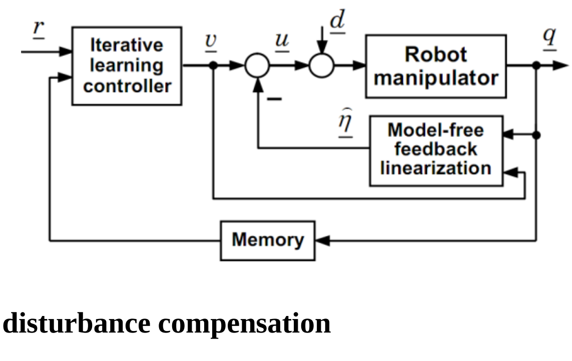

Figure 1. Suggested control scheme.

2.2. Feedback linearization via model-free disturbance compensation

Back to the compensator $\underline{v}(t)$ in (4) and with the notation of k as the current working period, i.e. with $t = kT + \tau,\ 0 \le \tau < T$ and $\underline{v}(t) = \underline{v}_k(\tau),\ \underline{x}(t) = \underline{x}_k(\tau)$, then (5) is rewritten as:

$$\dot{\underline{x}}_k(\tau) = A\underline{x}_k(\tau) + B\left[\underline{v}_k(\tau) - \hat{\underline{\eta}}_k(\tau) + \underline{\eta}_k(\tau)\right]. \tag{7}$$

The purpose of the design of model-free feedback linearization block (see Fig. 1) is now to estimate $\hat{\underline{\eta}}$ from measured values of $\underline{x}_k(\tau)$ for compensating $\underline{\eta}$.

Denote the last two measured values of $\underline{x}_k(\tau)$, one at current time instant $\tau = iT_s$ and the other at previous time instant $\tau - T_s = (i-1)T_s$, with $\underline{x}_k(iT_s),\ \underline{x}_k\left((i-1)T_s\right)$, respectively, where $0 < T_s \ll 1$ is an arbitrarily small chosen constant, then based on Taylor expansion of $\underline{x}_k(\tau)$ around $(i-1)T_s$ as below

$$\underline{x}_k\left((i-1)T_s\right) = \underline{x}_k(iT_s) - T_s\dot{\underline{x}}_k(iT_s) + \frac{T_s^2}{2}\ddot{\underline{x}}_k(\varsigma) \tag{8}$$

where $(i-1)T_s \le \varsigma \le iT_s$, or

$$\dot{\underline{x}}_k(iT_s) \approx \frac{\underline{x}_k(iT_s) - \underline{x}_k\left((i-1)T_s\right)}{T_s}. \tag{9}$$

If the last term of (8) is negligible, the equation (7) will be approximated by

$$\frac{\underline{x}_k(iT_s) - \underline{x}_k\left((i-1)T_s\right)}{T_s} \approx A\underline{x}_k(iT_s) + B\left[\underline{v}_k(iT_s) - \hat{\underline{\eta}}_k\left((i-1)T_s\right) + \underline{\eta}_k(iT_s)\right]. \tag{10}$$

The obtained equation (10) will be used for calculating the estimated value $\hat{\underline{\eta}}_k(iT_s)$ at the current time instant in a straightforward manner as follows. First, both the sign '$\approx$' and $\underline{\eta}_k(iT_s)$ in (10) are replaced with = and $\hat{\underline{\eta}}_k(iT_s)$, respectively.

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

Then, calculate

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

which yields

$$\hat{\underline{\eta}}_k(iT_s) = B^T\left[\frac{\underline{x}_k(iT_s) - \underline{x}_k\left((i-1)T_s\right)}{T_s} - A\underline{x}_k(iT_s)\right] - \left[\underline{v}_k(iT_s) - \hat{\underline{\eta}}_k\left((i-1)T_s\right)\right]. \tag{11}$$

Theorem 1: *The value* $\hat{\underline{\eta}}_k(iT_s)$ *obtained from* (11) *minimizes the* approximation *error of* (10).

Proof: Denote the error of both sides of (10) with

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

where

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

then the following optimization problem

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

has a unique solution

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

which coincides with $\underline{\hat{\eta}}_k(iT_s)$ given in (11). ■

It is worth noting here that the created estimator (11) for compensating the summarized vector $\underline{\eta}$ of disturbances and model error does not use the original mathematical model (1) of robot manipulators. Hence, the compensator (4) with $\underline{\hat{\eta}}$ obtained from (11) is model-free.

2.3. Iterative learning controller design

After the summarized disturbances $\underline{\eta}$ defined in (3) are compensated with the compensator (4), the system (2) becomes LTI as described in (5), which is now rewritten in the ILC language for repetitive systems as follows

$$\begin{cases}\underline{x}_k(i+1) = \hat{A}\underline{x}_k(i) + \hat{B}\left[\underline{v}_k(i) + \underline{\delta}_k(i)\right] \\ \underline{y}_k(i) = C\underline{x}_k(i)\end{cases} \tag{12}$$

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

$$\hat{A} = \exp\left(AT_s\right),\ \hat{B} = \int_0^{T_s} \exp\left(At\right)B dt \ \text{ and } \ C = \left(I_n\ ,\ \mathbf{0}_n\right). \tag{13}$$

Under the assumption that both matrices A_1, A_2 are suitably chosen such that the matrix A given in (6) becomes Hurwitz, the next control task is now to determine an appropriate learning parameter K for a P-Type update law

$$\underline{v}_{k+1}(i) = \underline{v}_k(i) + K\underline{e}_k(i) \ \text{ with } \ \underline{e}_k(i) = \underline{r}(i) - \underline{y}_k(i) \tag{14}$$

in order to satisfy the required convergence $\|\underline{e}_k(i)\| \to 0$ for all i, or at least as close as possible to the origin.

Since $\underline{v}_k(i)$ provided by the ILC (14) is a piecewise constant, the obtained discrete-time model (12) is absolutely equivalent to the continuous-time one (5). Moreover, the model (12) does not use any information of the system (1). Hence, it can be applied to all robot manipulators.

From (12), it is obtained with the assumption $\underline{\delta}_k(i) = \underline{0}$

$$\underline{y}_{k+1}(i)=C\widehat{A}^{i}\underline{x}_{k+1}(0)+\sum_{j=0}^{i-1}C\widehat{A}^{i-i-1}\widehat{B}\underline{v}_{k+1}(j),$$

which yields, based on (14) and the obviousness of repetitive ability $\underline{x}_k(0)=\underline{x}_{k+1}(0),\ \forall k$,

$$\begin{aligned}\underline{e}_{k+1}(i)&=\underline{r}(i)-\underline{y}_{k+1}(i)=\underline{r}(i)-\left[C\widehat{A}^{i}\underline{x}_k(0)+\sum_{j=0}^{i-1}C\widehat{A}^{i-i-1}\widehat{B}\left(\underline{v}_k(j)+K\underline{e}_k(j)\right)\right]\\&=\underline{r}(i)-\underline{y}_k(i)-\sum_{j=0}^{i-1}C\widehat{A}^{i-j-1}\widehat{B}K\underline{e}_k(j)=\left(I-C\widehat{B}K\right)\underline{e}_k(i)-\sum_{j=0}^{i-2}C\widehat{A}^{i-j-1}\widehat{B}K\underline{e}_k(j),\end{aligned}$$

or

$$\begin{pmatrix}\underline{e}_{k+1}(0)\\ \underline{e}_{k+1}(1)\\ \vdots\\ \underline{e}_{k+1}(N-1)\end{pmatrix}=\begin{pmatrix}I-C\widehat{B}K & \mathbf{0} & \cdots & \mathbf{0}\\ -C\widehat{A}\widehat{B}K & I-C\widehat{B}K & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ -C\widehat{A}^{N-1}\widehat{B}K & -C\widehat{A}^{N-2}BK & \cdots & I-C\widehat{B}K\end{pmatrix}\begin{pmatrix}\underline{e}_k(0)\\ \underline{e}_k(1)\\ \vdots\\ \underline{e}_k(N-1)\end{pmatrix}.$$

Hence,

$$\underline{\mathbf{e}}_{k+1}=\Phi\underline{\mathbf{e}}_k \tag{15}$$

where

$$\underline{\mathbf{e}}_k=\begin{pmatrix}\underline{e}_k(0)\\ \underline{e}_k(1)\\ \vdots\\ \underline{e}_k(N-1)\end{pmatrix}\ \text{and}\ \Phi=\begin{pmatrix}I-C\widehat{B}K & \mathbf{0} & \cdots & \mathbf{0}\\ -C\widehat{A}\widehat{B}K & I-C\widehat{B}K & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ -C\widehat{A}^{N-1}\widehat{B}K & -C\widehat{A}^{N-2}\widehat{B}K & \cdots & I-C\widehat{B}K\end{pmatrix}. \tag{16}$$

Theorem 2: *For the scenario of* $\underline{\delta}_k(i)=\underline{0}$ *the requirement* $\left\|\underline{e}_k(i)\right\|\to 0$ *for all* $i=0,1,\ \dots\ ,N-1$ *will be satisfied if and only if the P-Type learning parameter* K *is chosen so that* Φ *given in* (16) *becomes Schur*.

Proof: Since it is obviousness that the autonomous system (15) is stable if and only if Φ is Schur, the proof is complete. ■

2.4. Control algorithm and performance of closed-loop system

In order to facilitate the implementation of the proposed model-free controller (4), where $\underline{v}$ and $\underline{\eta}$ are obtained from (14) and (11), respectively, the following algorithm is established. In this control algorithm, each while-loop represents a trial, i.e. a working period T of repetitive robots.

Now, to complete this section, the output tracking performance of the closed-loop system illustrated in Fig. 1 will be investigated generally for the scenario of $\underline{\delta}_k(i)\neq\underline{0}$.

Theorem 3: *If* $\underline{d}$ *is continuous and bounded then the suggested model-free control framework in Fig. 1, including the feedback linearization block via disturbance compensator* (4), (11) *and the ILC block* (14), *drives the output tracking error* $|\underline{e}_k(\tau)|$ *of robot manipulators* (1) *to a* T_s- *dependent neighborhood* $\mathcal{O}$ *of origin*. *The smaller* T_s *is chosen, the smaller* $\mathcal{O}$ *will be.*

Proof: Due to the continuity and boundedness of $\underline{d}$, the total disturbance $\underline{\eta}$ is also continuous and bounded. Hence, $\underline{\delta}$ is also bounded. Denote the upper bound of $\underline{\delta}$ with Δ, then by Theorem 1 this upper bound Δ can be made arbitrarily small by reducing T_s.

Theorem 2 authenticated the rightness of

$$\dot{\underline{\omega}} = A\underline{\omega} + B\underline{v} \quad \text{with} \quad \underline{\omega} = \mathrm{vec}(\underline{r}, \dot{\underline{r}}) . \tag{17}$$

The subtraction (17) from (5) yields

$$\dot{\underline{\varepsilon}} = A\underline{\varepsilon} - B\underline{\delta} \quad \text{where} \quad \underline{\varepsilon} = \mathrm{vec}(\underline{e}, \dot{\underline{e}}) . \tag{18}$$

Since A is Hurwitz, the following Lyapunov equation $A^T P + PA = -Q$ with an arbitrarily chosen positive definite matrix Q always has a positive definite solution P. The usage of positive function $V(\underline{\varepsilon}) = \underline{\varepsilon}^T P \underline{\varepsilon}$ yields

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

where $\lambda_{\max}(Q)$ is the maximal eigenvalue of the positive definite matrix Q.

Since $\dot{V} < 0$ as long as

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

the vector of the tracking error $\underline{\varepsilon} = \mathrm{vec}(\underline{e}, \dot{\underline{e}})$ still tends to the origin until it reaches

$$\mathcal{O} = \left\{ \underline{\varepsilon} \in \mathbb{R}^{2n} \;\middle|\; \|\underline{\varepsilon}\| \le \frac{2\|PB\|\Delta}{\lambda_{\max}(Q)} \right\} \tag{19}$$

the point of proof. ■

Algorithm: Model-free output tracking control for robot manipulators with unknown parameters and matched disturbances.

---

1 Choose two matrices A_1, A_2 so that A given in (6) becomes Hurwitz and a sufficiently small constant $0 < T_s \ll 1$. Calculate $N = T/T_s$ and $\widehat{A}, \widehat{B}$ from A, B accordingly to (13). Chose arbitrarily $\hat{\underline{\eta}}$. Assign $\underline{v}(i) = \underline{r}(i),\ i = 0,1,\ \dots\ , N-1$ and $\underline{z} = \underline{0}$. Choose learning parameter K so that Φ given in (16) becomes Schur.

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

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

4   Send $\underline{u} = \underline{v}(i) - \hat{\underline{\eta}}$ to robot for a while of T_s. Measure $\underline{y}(i) = \underline{q}$ and $\underline{x} = \mathrm{vec}(\underline{q}, \dot{\underline{q}})$. Determine $\underline{e}(i) = \underline{r}(i) - \underline{y}(i)$.

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

6   **`end for`**

7   Establish $\underline{\mathbf{v}} = \mathrm{vec}\left(\underline{v}(0),\ \dots\ , \underline{v}(N-1)\right)$ and $\underline{\mathbf{e}} = \mathrm{vec}\left(\underline{e}(0),\ \dots\ , \underline{e}(N-1)\right)$.

```
    Set v ← v + Ke.
end while
```

3. SIMULATION RESULTS

3.1. Mathematical model of 2-DOF planar robot

To illustrate the performance of the proposed model-free controller, it is considered here a 2-DOF uncertain planar robot described by Euler-Lagrange model (1), where all mass, length and friction of robot arms are unknown. The model (1) of this uncertain robot consists of

$$\begin{aligned} M(\underline{q},\underline{\theta}) &= \begin{pmatrix} \theta_1+\theta_2\cos q_2 & \theta_3+\theta_4\cos q_2 \\ \theta_3+\theta_4\cos q_2 & \theta_5 \end{pmatrix} \\ C(\underline{q},\underline{\dot{q}},\underline{\theta}) &= \begin{pmatrix} \theta_6\dot{q}_2\sin q_2 & \dfrac{\theta_6}{2}\dot{q}_2\sin q_2 \\ \theta_7\dot{q}_1\sin q_2 & 0 \end{pmatrix} \\ \underline{g}(\underline{q},\underline{\theta}) &= \begin{pmatrix} \theta_8\cos q_1+\theta_9\cos(q_1+q_2) \\ \theta_{10}\cos(q_1+q_2) \end{pmatrix} \end{aligned} \tag{20}$$

It assumes additionally that the robot is repetitive with $T=10s$ and disturbed in inputs by

$$\underline{d}(t)=\begin{pmatrix} \theta_{11}\sin(\theta_{12}t) \\ \theta_{13}\cos(\theta_{14}t) \end{pmatrix}, \tag{21}$$

where all θ_i, $i=1\div 14$ are unknown constants. Note that the robot model (1) with parameters (20) and disturbances (21) will be utilized for the simulation to execute the robot dynamic only, not to design the controller.

3.2. Simulation results and discursions

For the simulation, there are assigned

$$\underline{r}(t)=\begin{pmatrix} \sin(\pi t/T)+0.2\sin(3\pi t/T) \\ 2\sin(\pi t/T)-0.5\sin(3\pi t/T) \end{pmatrix}$$

$$A_1=\begin{pmatrix} 31 & 0 \\ 0 & 13 \end{pmatrix},\ A_2=\begin{pmatrix} 12 & 0 \\ 0 & 8 \end{pmatrix},\ K=\begin{pmatrix} 0.85 & 0 \\ 0 & 0.3 \end{pmatrix} \text{ and } T_s=0.02s$$

After implementing the control algorithm presented in Subsection 2.4 as a Matlab program named VAST.m with the source code given in the Appendix, we obtain the simulation results exhibited in Figure 2 and Figure 3.

As seen in these visual tracking results, both joint variables converged on their desired references. Exactly after 200 trials the maximal value of the tracking error over the whole working period is approximately $\max|\underline{\mathbf{e}}_{200}(1)|\approx 0.05$ for the first joint variable and $\max|\underline{\mathbf{e}}_{200}(2)|\approx 0.04$ for the second joint variable, respectively. In addition, they also authenticated that the more the trials are executed, the smaller the tracking error will be. Hence, these simulation results completely satisfy all theoretical assertions obtained above.

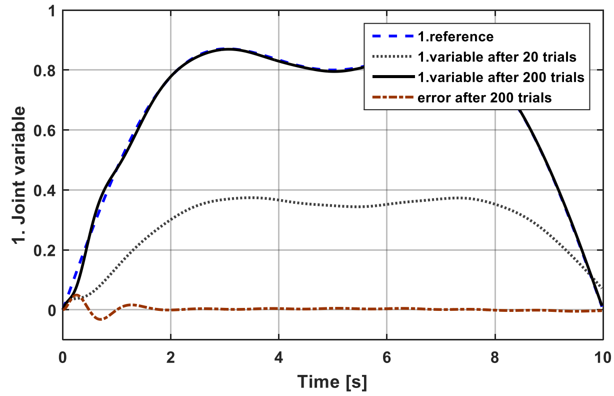

Figure 2. Output tracking of the first joint variable after 20 and 200 trials.

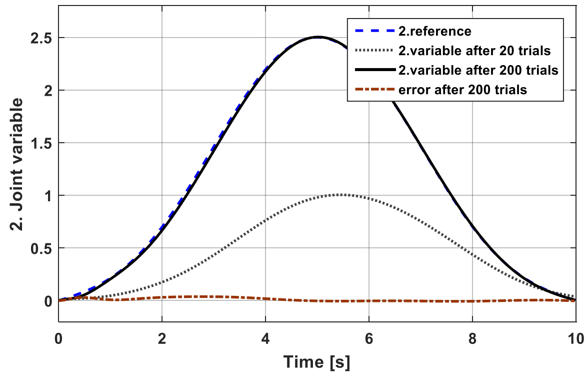

Figure 3. Output tracking of the second joint variable after 20 and 200 trials.

4. CONCLUSIONS

An intelligent controller for output tracking control of uncertain robot manipulators with matched disturbances is proposed in the article. This controller is established based on combining a summarized disturbances compensator acting as a model-free feedback linearization regulator and an output tracking controller which is created based on the ILC concept. This proposed model-free controller does not use the original Euler-Lagrange model (1). It needs only the measurement of $\underline{x} = \text{vec}(\underline{q}, \dot{\underline{q}})$ from robot manipulators for its operation. The simulation results have authenticated the intelligent adaptive performance of the proposed model-free controller as expected.

CRediT authorship contribution statement. Cao Thanh Trung: Methodology, Investigation, Funding acquisition, Supervision. Nguyen Hoai Nam: Formal analysis, Investigation. Nguyen Doan Phuoc: Methodology, Formal analysis, Supervision.

Declaration of competing interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

1 Frank L. Lewis, Darren M. Dawson, and Chaouki T. Abdallah - Robot manipulator control theory and practice. Marcel Dekker, Inc. 2004.

2 Craig J. J. – Introduction to Robotics, Mechanics and Control, 2a Edition, Addison-Wesley, 1989.

3 Sciavicco L., and Siciliano B. - Modeling and Control of Robot Manipulators, McGraw-Hill, 1996.

4 Abdelhamid Tayebi - Adaptive iterative learning control for robot manipulators **40** (7) (2004) 1195-1203.

5 Jean-Jacques Slotine, Weiping Li - Applied Nonlinear Control, Prentice-Hall, 1991.

6 Qu, Z. and Dawson, D. M. - Robust Tracking Control of Robot Manipulators, IEEE Press, 1996.

7 Feng Y., Zhou M., Yu X., and Han F. - Full order sliding mode control of rigid robotic manipulators, Asian Journal of Control. **21** (2019) 1228-1236.

8 Zhen S., Huang K., Sun H., Zhao H., and Chen Y. H. - Optimal robust control for rigid serial manipulators: A fuzzy approach, Asian Journal of Control **17** (2015) 2329-2344.

9 Long Jin, Shuai Li, Jiguo Yu, Jinbo He - Robot manipulator control using neural networks: A survey, Neurocomputing. **285** (2018) 23-34.

10 Wu Q. Yan and Cai J. - Neural network-based adaptive learning control for robot manipulators with arbitrary initial errors, IEEE Access. **7** (2019) 180194-180204.

11 Richard Lee, Liting Sun, Zining Wang, Masayoshi Tomizuka - Adaptive iterative learning control of robot manipulators for friction compensation, IFAC PapersOnLine. **52** (15) (2019) 75-180.

12 Farah Bouakrif - D-type iterative learning control without resetting condition for robot manipulators, Robotica. **29** (2011) 975-980.

13 Kevin L. Moore - Iterative learning control for deterministic systems, Springer, 1993.

APPENDIX

The source code of having used simulation program **VAST.m** is listed in detail as below.

```
global v i etah A1 A2 theta
for i=1:14  % uncertain model parameters
    theta(i)=2+random('norm',0,1);
end
% System parameters
T=10; Ts=0.02; N=T/Ts; % number of samples
A1=[31 0;0 13]; A2=[12 0;0 8]; K=[0.85 0;0 0.3];
A = eye(4)-Ts*[zeros(2) eye(2);-A1 -A2]; ti=[]; y=[];
for i=1:N  % Create desired references
    r(1,i)=sin(pi*(i-1)/N)+0.2*sin(3*pi*(i-1)/N);
    r(2,i)=2*sin(pi*(i-1)/N)-0.5*sin(3*pi*(i-1)/N);
    ti(i)=(i-1)*Ts;  % working time period
end
v=r; etah=zeros(2,1); t0=0; u=r; e=r; % initial values
M=200;    % setting trial number
for j=1:M
    x0=[0 0 1.6*pi/T 0.5*pi/T 0 0]; j
    for i=1:N
        [t,x]=ode45(@funVAST,[t0,t0+Ts],x0);
        % disturbances estimation
        etah = etah-v(:,i)+[zeros(2) eye(2)]*(A*x(end,1:4)'-x0(1:4)')/Ts;
        y(:,i)=[x0(1);x0(2)]; t0=t(end); x0=x(end,:);
    end
    e=r-y; v=v+K*e; % P learning
    if j==20; y20=y; end
end
figure(1); plot(ti,r(1,:),ti,y20(1,:),ti,y(1,:),ti,e(1,:));
legend('1.reference','1.variable after 20 trials','1.variable after 200
                                              trials','error after 200 trials');
figure(2); plot(ti,r(2,:),ti,y20(2,:),ti,y(2,:),ti,e(2,:));
legend('2.reference','2.variable after 20 trials','2.variable after 200
                                              trials','error after 200 trials');
```

The aforementioned simulation program used following subprogram named **funVAST.m** to execute the robot dynamic.

funVAST.m

```
function dx = funVAST(t,x)
% q-[x(1);x(2)]; q_dot=[x(3);x(4)]; q_ddot=[x(5);x(8)]
global v i etah A1 A2 theta
% System unknown inertia matrix
M1=theta(1)+theta(2)*cos(x(2)); M2=theta(3)+theta(4)*cos(x(2));
M3=theta(5); M=[M1 M2;M2 M3];
% System Coriolis and centrifugal terms
c11=theta(6)*x(4)*sin(x(2)); c12=c11/2;
c21=theta(7)*x(3)*sin(x(2)); c22=0; C=[c11 c12;c21 c22];
g1=theta(8)*cos(x(1))+theta(9)*cos(x(1)+x(2));
g2=theta(10)*cos(x(1)+x(2));
% Matched disturbances
d=[theta(11)*sin(theta(12)*t);theta(13)*cos(theta(14)*t)];
% Summarized disturbances
eta = d-((M-eye(2))*[x(5);x(6)]+C*[x(3);x(4)]+[g1;g2]);
% Control signals
u = v(:,i)-A1*[x(1);x(2)]-A2*[x(3);x(4)];
% Send to robot manipulator
dx = [zeros(2) eye(2) zeros(2);zeros(2) zeros(2) zeros(2);
           zeros(2) eye(2) zeros(2)]*x+[zeros(2);eye(2);zeros(2)]*(u+eta-etah);
end
```