Advanced open-closed-loop PIDD² /PID type ILC control of a robot arm

Mihailo P.Lazarević
Petar D.Mandić
Faculty of Mech. Eng.
University of Belgrade
Belgrade, Serbia
mlazarevic@mas.bg.ac.rs
pmandic@mas.bg.ac.rs

Boško Cvetković, Phd Student
University of Belgrade
Belgrade, Serbia
boskocvetkovic@gmail.com

Ljubiša Bučanović
Messer Tehnogas AD
Bor, Serbia
ljubisa.bucanovic@messer.rs

Mladen Dragović
Uno-lux processing,
Beograd, Srbija
dragovic.mladen@gmail.com

Abstract—In this paper, a new open-closed-loop PIDD² /PID type iterative learning control is studied for joint space trajectory tracking control of time-varying robotic manipulator. We propose intelligent control system which consists of a computed torque controller for linearizing robot dynamics. Then open-closed ILC is applied to the linearized system to further enhance tracking performance for repetitive tasks and deal with the model uncertainties. It is theoretically proven that the boundednesses of the tracking error are guaranteed in the presence of model uncertainty. Finally, a simulation example is presented to illustrate the feasibility and effectiveness of the proposed advanced open-closed iterative learning control scheme.

Keywords—iterative learning control; robot; computed torque; algorithms PID; PIDD²; trajectory tracking;

I. INTRODUCTION

In recent years, there has been increase in the interest and use of various advanced control techniques for engineering and scientific applications. An objective of modern control theory is to develop control design techniques that improve system performance for uncertain dynamic systems such as: parametric disturbances, unmodelled dynamics and external noises. Pressing demands of productivity and accuracy in today’s robotic applications have highlighted an urge to replace classical control strategies with their modern (intelligent) control counterparts [1,2]. Intelligent control is the control method which imitates human intelligence in decision making, learning, and problem solving, [3]. In the recent years several contributions have been produced on intelligent control and its applications on robotics, [4,5]. Based on the emulation of human learning, several control algorithms and methodologies have been developed which are classified as "learning control techniques" [6,7] as a special case of learning in general in the context of intelligent systems.

Also, in the last three decades, iterative learning control (ILC) has been extensively studied, achieves significant progress in both theory and application, and becomes one of the most active fields in intelligent control and system control, [4],[8-12]. ILC schemes were first proposed by Arimoto et al. [13], in the application of robot manipulators. Namely, some mechatronic systems, such as production machines or industrial robots, often perform the same task repeatedly. ILC is an intelligent control method for systems which perform tasks repetitively over a finite time interval where ILC approach is an imitation of a human learning process. Intelligent beings tend to learn by performing a trial (i.e. selecting a control input) and observing what was the end result of this control input selection. After that, they try to change their behavior in order to get an improved performance during the next trial. Emulating human learning, ILC uses knowledge obtained from the previous trial to adjust the control input for the current trial so that a better performance can be achieved, [14]. In general, ILC improves the tracking performance by a self-tuning process without using system model. The key purpose of using ILC is to achieve high performance after a few iterations. One advantage of ILC is that there is no requirement for the dynamic model of the controlled system. The basic idea of ILC schemes is to refine the control input to make better operation performance of the system on the next trial by use of updated data of the previous trial. These algorithms are generally based on the contraction mapping theory.

In this paper, robust intelligent control is suggested which combined a computed torque and advanced open-closed-loop ILC control algorithm PIDD²/PID type. Here, PID algorithm is used in closed-loop and PIDD² algorithm in open-loop. We can see later that proposed ILC algorithm achieves high performance after a few iterations. To the best of our knowledge, most of the existing ILC methods for robotic systems don’t include the first derivative of control in open loop ILC. In this contribution, we further investigate problem when matrix CB has not a full column rank which is a common assumption in ILC algorithms. Also, one of the applications of the learning control approach could be addressing safety issues such as those found in robotic-based manufacturing industries as well as in biomedical engineering. The authors believe the combination of the computed torque method and iterative learning control design approach for
robotic manipulators also has a potential for future enhancements.

The remainder of this paper is organized as follows. Section II presents some related work to the ILC algorithms that are considered in the paper. In Section III nonlinear and linearized mathematical model of the robot is given, and in Section IV presents suggested PIDD²/PID type ILC algorithm with some convergence results that will be used for the analysis. The results are illustrated using a numerical example in Section V, and finally Section VI gives some conclusions.

II. RELATED WORK

In terms of how to use tracking error signal of the previous iteration to form the control signal of the current iteration, the structures of ILC appeared as D-type, P-type, PD-type, and PID-type. In addition, ILC can be classified into two other kinds: off-line learning and on-line learning, [9]. Namely, a typical ILC in the time domain is a simple open-loop control (off-line ILC) and it cannot suppress unanticipated, non-repeating disturbances. Consequently, in a real application to overcome such drawbacks, an ILC scheme is usually performed together with a proper feedback controller for compensation where we often design a learning operator for the closed-loop (on-line ILC) systems that have achieved stability performance and convergence rate, [15], the combination of feedback control and ILC is a promising technique to achieve good tracking performance and to speed up the convergence process. During the past years, ILC has attracted considerable interests due to its simplicity and effectiveness of learning algorithm, and its ability to deal with problems with nonlinear, time-delay, uncertainties, as well as (non)singular systems integer or fractional order, [12],[16-19]. However, there are several critical requirements that limit the applications of ILC, especially to complex nonlinear (robotic) systems. Despite the widespread use of ILC in robot motion control, few attempts have been made to create an integrated design, [20].

Motivated by the mentioned investigations of ILC algorithms for robotic classical (non)singular systems, a new robust open-closed iterative learning control for a NeuroArm robotic manipulator is suggested in this paper. Therefore, we propose a joint space trajectory tracking control system consisting of a computed torque controller to linearize the robot dynamics. Then open-closed ILC is applied to the linearized system to further enhance tracking performance for repetitive tasks and deal with the uncertainty and disturbance and stability performance.

A. The $\lambda$-norm, maximum norm, the induced norm

For later using in proving the convergence of proposed learning control, the following norms are introduced [4] for $n$-dimensional Euclidean space $\mathbb{R}^n$:

the sup-norm $\|x\|_\infty = \sup_{1 \leq i \leq n} |x_i|$ $x = [x_1, x_2, \ldots, x_n]^T$,

and maximum norm $\|x\|_\infty = \max_{0 \leq t \leq T} |x(t)|$ $x = [x_1, x_2, \ldots, x_n]^T$;

the matrix norm as $\|A\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^{n} |a_{ij}|$, $A = [a_{ij}]_{m \times n}$

and the $\lambda$-norm for a real function:

$h(t)$, $t \in [0, T]$), $h : [0, T] \rightarrow \mathbb{R}^n$

$\|h(t)\|_{\lambda} = \sup_{t \in [0, T]} e^{-\lambda t} \|h(t)\|$, $\lambda > 0$ (1)

The induced norm of a matrix A is defined as:

$\|A\| = \sup \left\{ \frac{\|Ax\|}{\|x\|} : x \in X \text{ with } \|x\| \neq 0 \right\}$, (2)

where $\|\cdot\|$ denotes an arbitrary vector norm. In case $\|\cdot\|_\infty$ we have

$\|Ax\|_{\infty} \leq \|A\| \|x\|_{\infty}$, (3)

where $\|\cdot\|_{\infty}$ denotes the maximum value of matrix A. For the previous norms, note that

$\|h(t)\|_{\lambda} \leq \|h(t)\|_{\infty} \leq e^{\lambda T} \|h(t)\|_{\lambda}$.

The $\lambda$-norm is thus equivalent to the $\infty$-norm. For simplicity, in applying the norm $\|\cdot\|_{\infty}$, it will be omitted index $\infty$.

III. MATHEMATICAL MODEL OF ROBOT: NONLINEAR AND LINEARIZED CASE TRANSFORMATION OF SYSTEM MODEL

A. Nonlinear Mathematical Model of Robot

Robots today are making a considerable impact on many aspects of modern life, from manufacturing to health care, [21]. Therefore, robotic systems are more and more ubiquitous in the field of direct interactions with humans, in a so-called friendly home environment. One of these robotic systems capable of operating in such environments is NeuroArm robotic system. It is an integral part of the Laboratory of Applied Mechanics, at Faculty of Mechanical Engineering in Belgrade, Fig. 1. From the mechanical point of view, NeuroArm robotic manipulator has seven degrees of freedom. Schematic view of a NeuroArm robot is given in Fig. 2. Accurate robot control and realistic robot simulation require an accurate dynamic robot model, [22]. Hence, by applying the Rodrigues approach, [23], a detailed mathematical model of a NeuroArm robot will be given first.
The mechanical structure of a robot manipulator consists of a sequence of rigid bodies (or links) interconnected by means of one-degree-of-freedom joints forming kinematical pairs of the fifth class, [23]. The open chain system of rigid bodies \((V_1, V_2, \ldots, V_n)\) is shown in Fig. 3. Two neighboring bodies \((V_i, V_{i+1})\) are connected with a joint \((i)\), which allows translation or rotation of body \((V_i)\) in respect to the body \((V_{i+1})\). The values \(q^i, i = 1, \ldots, n\) represent generalized coordinates and define a configuration of the mechanical model, where \(n\) is a number of bodies in the system.

The reference frame \(Oxyz\) is the inertial Cartesian frame, and the reference frame \(Oxyz_i\) is a local body–frame which is associated with the body \((V_i)\). At an initial time, the corresponding axes of reference frames were parallel, i.e. all the variables \(q_i^i, i = 1, \ldots, n\) are zero and the robotic system is in reference configuration (position). Parameters \(\xi_i, \xi_j = 1 - \xi_i\) denote parameters for recognizing joints between bodies \((V_i, V_{i+1})\) and \((V_i, V_{i+2})\), \(\xi_j = 1 - \xi_j\), \(\xi_i = 1\)-prismatic, \(0\)-cylindrical joint). The geometry of the system is defined by the unit vectors \(\vec{e}_i\) and the position vectors \(\vec{p}_i\) and \(\vec{p}_{ii}\) expressed in local coordinate systems \(C_i\), \(\xi_i, \eta_i, \zeta_i\) are connected to mass centers of bodies in a multibody system [23]. Unit vectors \(\vec{e}_i, i = 1, 2, \ldots, n\) are describing the axis of rotation (translation) of the \(i\)-th segment with respect to the previous segment, and \(\vec{p}_{ii} = O_iO_{i+1}\) denotes a vector between two neighboring joints in a multi body system, while the position of the center of mass of \(i\)-th segment is expressed by vectors \(\vec{p}_i = O_iC_i\). For the entire determination of this mechanical system, it is necessary to specify masses \(m_i\) and tensors of inertia \(J_{C_i}\) expressed in local coordinate systems. Dynamic equations of motion for the robot system can be obtained by applying Lagrange equations of the second kind in the covariant form as follows:

\[
\sum_{\alpha=1}^{n} a_{\alpha\alpha} \ddot{q}_\alpha + \sum_{\alpha=1}^{n} \sum_{\beta=1}^{n} \Gamma_{\alpha\beta\gamma} \dot{q}_\alpha \dot{q}_\beta = Q_\gamma, \quad \gamma = 1, 2, \ldots, n \tag{5}
\]

The coefficients \(a_{\alpha\alpha} = a_{\alpha\beta}\) are the covariant coordinates of the basic metric tensor \([a_{\alpha\beta}]_{\in R^m}\) and \(\Gamma_{\alpha\beta\gamma}\) present Christoffel symbols of the first kind. Coefficients of the metric tensor are defined as, [23]:

\[
a_{\alpha\beta} = \sum_{i=1}^{n} m_i \left( \vec{T}_{ai(i)} \right) \left( \vec{T}_{bi(i)} \right) \left( \vec{F}_{ai(i)} \right) \left( \vec{F}_{bi(i)} \right) \left( \vec{H}_{ai(i)} \right) \left( \vec{H}_{bi(i)} \right), \tag{6}
\]

where quasi-base vectors \(\vec{T}_{ai(i)}\) and \(\vec{F}_{ai(i)}\) are

\[
\vec{T}_{ai(i)} = \begin{cases} \frac{\xi_\alpha \bar{e}_\alpha}{\xi_i} \left( \sum_{k=1}^{n} \left( \vec{p}_{ik} + \xi_i \bar{e}_k q^k \right) \right) + \bar{p}_i, & \alpha \leq i, \\ 0, & \forall \alpha > i, \end{cases} \tag{7}
\]

\[
\vec{F}_{ai(i)} = \begin{cases} \frac{\xi_\alpha \bar{e}_\alpha}{\xi_i}, & \forall \alpha \leq i, \\ 0, & \forall \alpha > i, \end{cases} \tag{8}
\]

and Cristoffel symbols are

\[
\Gamma_{\alpha\beta\gamma} = \frac{1}{2} \left( \frac{\partial a_{\alpha\beta}}{\partial q_\gamma} + \frac{\partial a_{\alpha\gamma}}{\partial q_\beta} - \frac{\partial a_{\beta\gamma}}{\partial q_\alpha} \right), \quad \alpha, \beta, \gamma = 1, \ldots, n. \tag{9}
\]

The generalized forces \(Q_\gamma\) can be presented in the following expression (6), wherein \(Q_\gamma^g, Q_\gamma^c, Q_\gamma^v, Q_\gamma^s, Q_\gamma^f\) denote the generalized control, gravitational, viscous, spring and friction forces, respectively.

\[
Q_\gamma = Q_\gamma^g + Q_\gamma^c + Q_\gamma^v + Q_\gamma^s + Q_\gamma^f, \quad \gamma = 1, 2, \ldots, n \tag{10}
\]

**B. Linearized Model of Robot, Transformation of System Model**

The robot arm dynamics can be written in compact matrix form as (where in our case \(Q^g, Q^c, Q^f = 0\)):

\[
A(q) \ddot{q} + \left( C(q, \dot{q}) - Q^g \right) = A(q) \ddot{q} + n(q, \dot{q}) = \nu \tag{11}
\]

where \(A(q)\) represents basic metric tensor (or inertia matrix), \(C(q, \dot{q})\) is a matrix that includes centrifugal and Coriolis forces, respectively.
effects, and \( Q^a = v \), respectively, [24]. The state vector of the nonlinear robot arm system is introduced as follows:

\[
\dot{x} = \begin{bmatrix} \dot{x}_1, \dot{x}_2 \end{bmatrix}^T = (q, \dot{q})^T \in \mathbb{R}^{2n}
\]

so one can obtain (11) in state space form:

\[
\begin{align*}
\dot{x}(t) &= f(\dot{x}(t)) + g(\dot{x}(t)) v(t) \\
y(t) &= h(\dot{x}(t)) = [1 \quad 0] \dot{x}(t)
\end{align*}
\]

Using the information about the dynamic model, a nonlinear technique known as computed torque control is implemented, with the aim to simplify equations of motion. Therefore, a feedback linearization control technique is applied on a given robot arm which is known as computed torque control, [25]. It is a special application of feedback linearization technique used in nonlinear control systems, [26]. Computed torque controllers can be very effective, since they provide us independent joint control, which can then be used together with some classical and modern design techniques, as we will see in the rest of this paper. A schematic diagram of a feedback linearization technique is illustrated in Fig. 4 where a nonlinear controller will be realized as:

\[
v = A(q) \cdot u + n(q, \dot{q})
\]

The nonlinear transformation (13) has converted a complicated nonlinear controls design problem into a simple design problem for a linear system consisting of \( n \) decoupled subsystems. First, it is necessary to verify if the dynamic can be linearized in terms of input to state variables. This is possible if and only if both following conditions are simultaneously satisfied, [27].

a) vectors of a matrix

\[
G(x) = \begin{bmatrix} g(x), ad f g(x), \ldots, ad f^{n-1} g(x) \end{bmatrix}
\]

are of full rank, where \( r \) indicates the relative order (number) of the system and \( ad f g(x) \) denotes the Lie bracket derivative which is defined as:

\[
ad f g(x) = \nabla g \cdot f - \nabla f \cdot g, \quad \nabla g = \frac{\partial g}{\partial x}, \quad \nabla f = \frac{\partial f}{\partial x}
\]

b) distribution of

\[
\text{span} \begin{bmatrix} g(x), ad f g(x), \ldots, ad f^{r-1} g(x) \end{bmatrix}
\]

is involutive. In our case, we have the case \( r = n \), [28], i.e. the linearization of the input-output coincides with the linearization of the input-state and there is no internal dynamics of the system. Also, taking into account that matrix \( A(q) \) is nonsingular and (9), in the ideal case, one can linearize the dynamics as follows:

\[
\ddot{q}(t) = u(t)
\]

Under the influence of model uncertainties \( \eta_i = \eta_i(t), i = 1, 2, \ldots, n \) we have:

\[
\ddot{q}(t) = u(t) + \eta(t)
\]

or in state-space

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t) + D\eta(t) \\
A &= \begin{bmatrix} 0_{n \times n} & I_{n \times n} \\ 0_{n \times n} & 0_{n \times n} \end{bmatrix}, \quad B = D = \begin{bmatrix} 0_{n \times n} \\ I_{n \times n} \end{bmatrix}
\end{align*}
\]

IV. PIDD\(^2\)/PID TYPE ILC DESIGN

A. Advanced Open-Closed Loop ILC

Fig. 4. Block diagram of the open-closed PIDD\(^2\)/PID type of ILC for a robotic system

Resulting control scheme appears in Fig. 4. We investigate the problem of robust tracking of a repeated trajectory in the joint space of linearized system (17) over a time interval \([0, T], T \in \mathbb{R}\), where \( T \) is the time duration, \( i \) denotes the iteration index or the operation number, \( x_i \in \mathbb{R}^n, u_i \in \mathbb{R}^m, y_i \in \mathbb{R}^r \) are the state, control input and output of the system, respectively. In other words, the robot manipulator must repeatedly follow the desired trajectory \( q_d(t) \in \mathbb{R}^n, t \in J \) in the joint space under the influence of model uncertainties, \( \eta_i(t) \in \mathbb{R}^n \). For the linearized dynamics of the robot arm (17), the open-closed ILC algorithm is suggested which comprises two types of control laws: a feed-forward PIDD2 control law and a PID feedback law, see Fig. 4.

\[
u_{i+1}(t) = u_{ff+1}(t) + u_{fd+1}(t)
\]

Therefore, open-closed loop PIDD\(^2\)/PID-type iterative learning algorithm is introduced as:
\[ u_{i+1}(t) = u_i(t) + \Pi \cdot \dot{u}_i(t) + \]
\[ + \Gamma_1 \left[ \hat{e}_i(t) + R_1 \hat{e}_i(t) + R_2 \epsilon_i(t) + R_3 \int_0^t \epsilon_i(\tau) d\tau \right] + E \hat{x}_d \]
\[ + \Gamma_2 \left[ \hat{e}_{i+1}(t) + Q_1 \epsilon_{i+1}(t) + Q_2 \int_0^t \epsilon_{i+1}(\tau) d\tau \right] \] (19)

where \( e_i(t) = y_d(t) - y_i(t) \) is the trajectory tracking error in \( i \)-th iteration, \( e_{i+1}(t) = y_d(t) - y_{i+1}(t) \) is the trajectory tracking error in \( i+1 \)-th iteration, \( y_d(t) \) denotes desired output trajectory. \( \Gamma_1, \Gamma_2, R_1, R_2, R_3, Q_1, Q_2 \in \mathbb{R}^{n \times n} \) are open-loop positive-definite diagonal learning matrices and \( \Pi \) is a constant weighting matrix, i.e.

\[ R_1 = \text{diag}\{r_{11}, r_{22}, \ldots, r_{nn}\}, \quad Q_1 = \text{diag}\{q_{11}, q_{22}, \ldots, q_{nn}\}, \] (20)

\[ \Gamma_1 = \text{diag}\{\gamma_{11}, \gamma_{22}, \ldots, \gamma_{nn}\}. \]

For the first time, we introduce in ILC control scheme first derivative of control \( \dot{u}_i(t) \).

The following assumptions on the system (17) are imposed.

A1. The desired trajectories \( y_d(t), x_d(t) \) are continuously differentiable on \([0, T]\).

A1 is a reasonable assumption which makes possible calculating \( \dot{e}_i = \dot{y}_d - \dot{y}_i, \quad \ddot{e}_i = C \ddot{x}_d = C (\ddot{x}_i - \ddot{x}_d). \)

A2. The system (17a),(17b) is causal. Specifically, for a given desired output trajectory \( y_d(t) \), there exists a unique bounded control input \( u_d(t) \) such that the system has a unique bounded state \( x_d(t) \) and \( y_d(t) \), i.e:

\[ \dot{x}_d(t) = Ax_d(t) + Bu_d(t), \quad y_d(t) = Cx_d(t), \] (21)

By taking the proposed control law and applying (25),(26) one obtains:

\[ \begin{align*}
\dot{e}_i &= x_d(t) - x_i(t), \quad \ddot{e}_i = \dot{x}_d(t) - \dot{x}_i(t) \\
\dot{u}_i &= u_d(t) - u_i(t), \quad \ddot{u}_i = \dot{u}_d(t) - \dot{u}_i(t)
\end{align*} \] (25)

Also, the tracking error and its derivatives can be presented as:

\[ \begin{align*}
\hat{e}_i &= C \delta x_i = C A \delta x_i + C \delta u_i - C \delta y_i \\
\hat{\epsilon}_i &= C \delta x_i = C A^2 \delta x_i + C A \delta u_i + C \delta \dot{u}_i - C \delta \dot{y}_i
\end{align*} \] (26)

Proof: Let

\[ \begin{align*}
\delta x_i &= x_d(t) - x_i(t), \quad \delta \dot{x}_i = \dot{x}_d(t) - \dot{x}_i(t) \\
\delta u_i &= u_d(t) - u_i(t), \quad \delta \dot{u}_i = \dot{u}_d(t) - \dot{u}_i(t)
\end{align*} \]

A3. The initial resetting conditions hold for all iterations, i.e.

\[ x_i(0) = x_d(0), \quad i = 0, 1, 2, \ldots, \] (23)

A4. The uncertainties \( \eta_i(t) \in \mathbb{R}^p, \tilde{\eta}_i(t) \in \mathbb{R}^q \), and terms \( \hat{u}_d(t), \dot{x}_d(t) \) are uniformly bounded. In the sequel, we use positive constants \( d_u, d_x, d_w, d_x \) to denote the upper bounds for \( \eta_i(t), \tilde{\eta}_i(t), \hat{u}_d(t), \dot{x}_d(t) \) i.e., \( \forall i \in [0, T] \) and

\[ \forall i \rightarrow \|\eta_i(t)\| \leq d_w, \quad \|\tilde{\eta}_i(t)\| \leq d_w, \quad \|\hat{u}_d(t)\| \leq d_u, \quad \|\dot{x}_d(t)\| \leq d_x. \]

Assumption 4. puts the boundedness restrictions of the \( \eta_i(t), \tilde{\eta}_i(t), \hat{u}_d(t), \dot{x}_d(t) \) on given time interval \( \forall t \in [0, T] \).

Based on contraction mapping, all existing ILC methods require Assumption 4. The reason is as follows. ILC methodology tries to use as little system prior knowledge as possible in its design, and the lack of such system knowledge, however, gives rise to a difficulty in designing a suitable (stable) closed-loop controller.

A sufficient condition for convergence of a proposed open-closed loop ILC is given by the Theorem 1 and proved as follows.

**B. Convergence Analysis**

**Theorem 1:** Let system (17a)-(17b) satisfies assumptions (A1)-(A4). If updating law (19) is applied together with learning gain matrices \( \Gamma_1, \Gamma_2, R_i \) is designed such that

\[ \left\| (I + \Gamma_2 CB)^{-1} \left( I - \Gamma (CAB + R CB) \right) \right\| \leq \rho < 1 \] (24)

In addition, a gain matrix \( \Gamma_2 \) is such that \( (I + \Gamma_2 CB) \) is invertible then, when \( i \rightarrow \infty \) bounds of the tracking errors \( \|x_d(t) - x_i(t)\|, \|y_d(t) - y_i(t)\|, \) and \( \|\dot{u}_d(t) - u_i(t)\| \)

require Assumption 4. The reason is as follows. ILC methodology tries to use as little system prior knowledge as possible in its design, and the lack of such system knowledge, however, gives rise to a difficulty in designing a suitable (stable) closed-loop controller.

A sufficient condition for convergence of a proposed open-closed loop ILC is given by the Theorem 1 and proved as follows.

**Proof:** Let

\[ \begin{align*}
\delta x_i &= x_d(t) - x_i(t), \quad \delta \dot{x}_i = \dot{x}_d(t) - \dot{x}_i(t) \\
\delta u_i &= u_d(t) - u_i(t), \quad \delta \dot{u}_i = \dot{u}_d(t) - \dot{u}_i(t)
\end{align*} \]

Also, the tracking error and its derivatives can be presented as:

\[ \begin{align*}
\dot{e}_i &= C \delta x_i = C A \delta x_i + C \delta u_i - C \delta y_i \\
\dot{\epsilon}_i &= C \delta x_i = C A^2 \delta x_i + C A \delta u_i + C \delta \dot{u}_i - C \delta \dot{y}_i
\end{align*} \]

By taking the proposed control law and applying (25),(26) one obtains:

\[ \begin{align*}
\delta u_{i+1} &= (I + \Gamma_2 CB) \delta u_i - \left[ (I - \Gamma_1 (CAB + R CB)) \right] \delta u_i - \Gamma_1 \delta \dot{u}_d - E \dot{x}_d \\
&= -\Gamma_1 \left( CA^2 + R_1 CA + R_2 C \right) \delta x_i - \Gamma_1 R_3 C \int_0^t \delta x_i(\tau) d\tau \\
&= -\Gamma_1 \left( CA + Q \right) \delta x_i - \Gamma_1 Q \int_0^t \delta x_i(\tau) d\tau \\
&+ \Gamma_1 \left( R_1 CD + CD \right) \eta_i + \Gamma_1 C \delta \dot{y}_d
\end{align*} \] (27)

One can choose gain matrix \( \Gamma_2 \) is such that \( (I + \Gamma_2 CB) \) is invertible as well as \( \Pi, \Gamma_1 \) such that \( \Pi - \Gamma_1 CB = 0 \), and taking the norms \( \|\cdot\|_\delta \) of (27), and by using the condition of Theorem 1, it follows, (see Appendix):
\[ \| \delta u_{i+1} \|_2 \leq \rho \| \delta u_i \|_2 + (\beta_1 + O(\lambda^{-1}) \beta_2) \| \delta x_i \|_2 + (\beta_2 + O(\lambda^{-1}) \beta_4) \| \delta x_{i+1} \|_2 + \gamma_1 d_\eta + \gamma_2 d_\eta + \gamma_3 d_m + \gamma_4 d_u + \gamma_5 d_x \]

(28)

Also, we have

\[ \delta x_i = \int_0^t \left( A \delta x(\tau) + B \delta u_i(\tau) + D \eta_i(\tau) \right) d\tau \]

(29)

and taking the norm \( \| \cdot \| \) on the both sides of (29), then applying Grownwall-Bellman lemma and multiplying by \( e^{-\lambda_1 t} \), we obtain (see Appendix):

\[ \| \delta x_i \|_2 \leq d_\eta T + bO(\lambda^{-1}) \| \delta u_i \|_2, \]

(30)

and

\[ \| \delta x_{i+1} \|_2 \leq d_\eta T + bO(\lambda^{-1}) \| \delta u_{i+1} \|_2, \]

(31)

Now, substituting (31) and (32) into (29), we can get

\[ \| \delta u_{i+1} \|_2 \leq \tilde{\rho} \| \delta u_i \|_2 + \tilde{\varepsilon} \]

(32)

where one can make by using a sufficiently large \( \tilde{\lambda} \) so that \( \tilde{\rho} < 1 \). Therefore, according to Lemma 1, [1], it can be concluded that:

\[ \lim_{i \to \infty} \| \delta u_i \|_2 \leq \frac{1}{1 - \tilde{\rho}} \tilde{\varepsilon} \].

(33)

This completes the proof of Theorem 1. In our linearized case of a matrix \( CB \) has not a full column rank, as well as matrix \( CAB \) has a full column rank, so we obtain the following important corollary.

**Corollary 1.** Let system (17a)–(17b) satisfies assumptions (A1)–(A4), and the matrix \( CAB \) has a full column rank and the matrix \( CB \) has not a full column rank. If updating law (19) is applied together with learning gain matrices \( \Gamma_i \) is designed such that

\[ \| (I - \Gamma_i CAB) \| \leq \rho < 1 \]

(34)

and, when \( i \to \infty \) bounds of the tracking errors \( \| x_{d}(t) - x_i(t) \|, \| y_{d}(t) - y_i(t) \|, \) and \( \| u_{d}(t) - u_i(t) \| \) converge asymptotically into the specified bounds.

**Proof.** A proof is similar to the proof of Theorem 1 and it is omitted here.

Applying proposed intelligent ILC control we improved ILC control for robotic manipulators, because it does not have to be a fulfilled a next constraint (the matrix \( CB \) has a full column rank which is a common assumption in ILC algorithms).

**V. SIMULATION RESULTS**

For simplicity, we used a NeuroArm robotic manipulator with revolute joints, with first three active DOFs, Fig. 2, to solve the trajectory tracking problem in joint space. For the simulation, we use the next model parameters of robot arm \( m_1 = 4 \text{ kg}, m_2 = 6 \text{ kg}, m_3 = 4 \text{ kg} \) \( L_1 = 0.3 \text{ m}, L_2 = 0.6 \text{ m}, L_3 = 0.4 \text{ m}, \) [29]. Numerical simulations were carried out to demonstrate the feasibility and effectiveness of the proposed advanced ILC PID/PID type. The desired trajectories are given as

\[ q_{d1}(t) = 0.5 \sin t, \quad q_{d2}(t) = 0.1 \sin t + 0.1, \quad q_{d3}(t) = 1 - e^{-t}, \quad \forall t \in [0, T], \quad T = 3 \text{ sec}. \]

(35)

Model of feedback linearized robotic system in state space is given as:

\[ \dot{x}_k(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_k(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} B u_k(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \eta_k(t), \quad k = 1, 2, 3 \]

\[ y_k(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k(t) \]

where \( \eta_k(t), k = 1, 2, 3 \) are model uncertainties:

\[ \eta_1(t) = 0.2 \cdot \sin(\pi t), \quad \eta_2(t) = 0.1 \cdot (1 - e^{-t}), \quad \eta_3(t) = 0.2 \cdot \sin(\pi t), \quad t \in [0, 3] \]

(36)

For the elements of learning gain matrices, \( \Gamma_1, \Gamma_2 \), the following values are adopted:

\[ \Gamma_1 = \text{diag} \{0.1, 0.1, 0.1\}, \quad \Gamma_2 = \text{diag} \{55, 5, 5\}, \]

(37)

The values of learning gain matrix \( \Gamma_1 \) need to satisfy the condition \( 0 < \Gamma_1 \leq 1 \). Also, in this case, one can easily check that

\[ \| (I - \Gamma_i CAB) \| \leq 0.9 < \rho < 1 \]

(38)

Learning gain matrices \( R_1, R_2, R_3, Q_1, Q_2 \) can be chosen by trial and error, taking into account the fact that the other components in the ILC updating law are related to the stability of the desired trajectory as well as robustness performance of ILC. In the numerical simulation, we select the following gain matrices:

\[ Q_1 = \text{diag} \{150, 90, 3\}, Q_2 = \text{diag} \{5, 10, 5\}, \]

\[ R_1 = \text{diag} \{5, 40, 40\}, \]

\[ R_2 = \text{diag} \{0.5, 10, 10\}, R_3 = \text{diag} \{50, 20, 20\}. \]

(39)

Fig. 5 shows that the ILC control law drives the considered robotic system output on the interval \( t \in [0, 3] \) through the desired trajectory as closely as possible.
It is clear that the trajectory tracking error decreases through the iterations. Also, we can find (see Fig. 6), that proposed requirement of tracking performance is achieved at the sixth iteration.

VI. CONCLUSION

In this paper, we studied the tracking problem of robot manipulators with revolute joints via intelligent control which includes advanced ILC control. First, a feedback linearization control technique (computed torque control) is applied on a given robot arm. Then, the proposed intelligent control algorithm takes the advantages offered by closed-loop control PID type and open-loop control PIDD2 type of ILC. Suggested robust ILC algorithm is applied to the linearized system to further enhance tracking performance for repetitive tasks and deal with the model uncertainties. Sufficient conditions for guaranteeing the convergence and robustness of proposed open-closed-loop ILC system are presented. Finally, a simulation example is presented to illustrate the effectiveness of the proposed robust ILC scheme for a robot arm. Future work will be focused on implementing the intelligent optimization method(s) to tune learning gain matrices of proposed ILC control.

ACKNOWLEDGMENT

Authors gratefully acknowledge the support of Ministry of Education, Science and Technological Development of the Republic of Serbia under the project TR 35006 (M.P.L.) as well as supported by project TR 33047 (P.D.M.).
From assumption A4, we can get:

\[
\beta_k = \left\| \left( I + \Gamma_1 C A + R C A + R_2 C \right)^{-1} \Gamma_1 \right\|.
\]  \hspace{1cm} (A1)

\[
\beta_1 = \left\| \left( I + \Gamma_1 C A + R C A + R_2 C \right)^{-1} \Gamma_1 \right\|.
\]

\[
\gamma_1 = \left\| \left( I + \Gamma_1 C A + R C A + R_2 C \right)^{-1} \Gamma_1 \right\|.
\]

\[
\gamma_2 = \left\| \left( I + \Gamma_1 C A + R C A + R_2 C \right)^{-1} \Gamma_1 \right\|.
\]

\[
\gamma_3 = \left\| \left( I + \Gamma_1 C A + R C A + R_2 C \right)^{-1} \Gamma_1 \right\|.
\]

\[
\gamma_4 = \left\| \left( I + \Gamma_1 C A + R C A + R_2 C \right)^{-1} \Gamma_1 \right\|.
\]

\[
a = \|A\|, \quad b = \|B\|, \quad \beta_1 = \|D\|.
\]

\[
O(\lambda^{-1}) = (1 - e^{-\lambda T}) / \lambda \leq 1 / \lambda, \quad O(\lambda^{-1}) = 1 / (\lambda - a)
\]  \hspace{1cm} (A2)

\[
\alpha_1 = \left( \beta_1 + O(\lambda^{-1}) \beta_2 \right) \left\| d_1 d_y T + b O(\lambda^{-1}) \right\|.
\]

\[
\alpha_2 = \left( \beta_1 + O(\lambda^{-1}) \beta_2 \right) \left\| d_1 d_y T + b O(\lambda^{-1}) \right\|.
\]

\[
\delta_2 = \gamma_3 + \gamma_4 + \gamma_5 + \gamma_6 d_y + \gamma_7 d_x
\]

\[
\bar{e} = \delta_2 / (1 - \alpha_2), \quad \bar{\rho} = (\rho + \alpha_1) / (1 - \alpha_2)
\]

From assumption A4, we can get:

\[
d_1 \int_0^\infty \left\| \eta_1 (\tau) \right\| d\tau \leq d_1 \cdot d_y T
\]  \hspace{1cm} (A4)

REFERENCES


[28] Batalov S., Non-linear robust control of a multivariable robotic system using fractional calculus and genetic algorithms, Diploma work, Faculty of Mech. Eng, University of Belgrade, 2011.
