Journal of Robotics and Control (JRC)
Volume 5, Issue 3, 2024
ISSN: 2715-5072, DOI: 10.18196/jrc.v5i3.21791

Robust Adaptive Iterative Learning Control for De-Icing Robot Manipulator

Thanh Quyen Ngo [1*], Thanh Hai Tran [2]
[1] Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Vietnam
[2] Office of Planning and Investment, Industrial University of Ho Chi Minh City, Vietnam
Email: [1] ngothanhquyen@iuh.edu.vn, [2] tranthanhhai@iuh.edu.vn
*Corresponding Author

***Abstract*—This paper introduces a new method of controlling uncertain robot using robust adaptive iterative learning control (RAILC) to track the trajectory in iterative operation mode. This method uses a PD controller combined with gain switching and forward learning techniques to predict the desired torque of the actuator. Using the Lyapunov method, this paper presents an RAILC control scheme for an uncertain robot system with structural and unstructured properties while ensuring the stability of the closed-loop system in the domain repeat. This study believes that this new control method can advance the field of robot control, especially in dealing with structured and unstructured uncertainties. It can help improve the flexibility and performance of robotic systems in real-world applications, such as automated manufacturing, transportation services, or healthcare. At the same time, providing simulation and test results demonstrates the effectiveness of the new control method in deicing high voltage power lines for robots.**

Keywords—PD Control; Learning Control; De-Icing Robot Manipulator; Adaptive Iterative Learning Control.

I. INTRODUCTION

Nonlinear adaptive control (AC) technology plays an important role in addressing the challenges of uncertain and variable-parameter dynamic systems [1]–[4]. However, the adaptive of AC depends heavily on the design of the learning rule and initial parameters. This creates many difficulties in designing a controller that can be applied to many different nonlinear objects. When the disadvantages of AC became a matter of concern for many scientists, the estimation method applied adaptive learning rule design to handle uncertain parameters and it became an effective means. In cases where the system does not meet the required conditions, iterative learning control (ILC) is a flexible solution. This is a particularly useful technique when dealing with processes that repeat tracking tasks at fixed intervals. ILC uses information from previous trials to shape subsequent control inputs, continuously improving tracking accuracy. Uniquely, ILC can be applied not only to reference trajectories that remain constant between trials but can also be adjusted to meet a more general goal without requiring tracking of a single target–specific static [5]. In general, the applications of ILC are not limited to specialized cases such as robots [6]–[7], high–speed trains [8]–[9], and subways [10]–[11], but can also be extended to apply to a wide range of nonlinear and parametrically variable dynamic systems. This opens opportunities to develop more flexible and precise control methods in various fields.

Since the 1980s, research on ILC algorithms in robot control has developed significantly, mainly based on the contraction mapping method [12]–[14]. ILC has proven effective in handling repetitive tasks within limited periods [15]–[20]. In the recent decade, the growing interest in robot manipulator adaptive ILC has made the field diverse. In some research, Tayebi [21] proposed three simple ILC schemes to solve robot trajectory tracking problems. Chien and colleagues [22] developed an adaptive learning rule combining the time domain and iteration domain to estimate uncertainty in robot control. He et al. proposed an adaptive ILC algorithm based on impulse neural networks to achieve high tracking performance for uncertain robotic systems [23]. In [24], Cao et al. developed an adaptive boundary ILC scheme for a two–link flexible controller. Li and colleagues' research [25] focuses on iterative learning impedance control for rehabilitation robots. Although the above ILC algorithms have achieved much progress, the original problem of ILC still needs to be solved. In practice, the required error–free initial reset in each iteration cycle is difficult due to the limitations of the actual reset mechanism. Studies have attempted to address this issue [26], and only a few results have been reported [27]–[29], including time–varying boundary layer, initial corrective, error–tracking, and more. Jin [30] used initial disciplinary action to face this problem, while Ouyang Zhang and colleagues [31] worked on designing error–tracking ILC for robots with initial errors depending on the idea.

Under certain conditions, artificial neural networks can approximate various nonlinear functions to any designed precision. Neural network controllers can perform well in cases where system dynamics information is known [32]–[36]. There have been many suitable results on adaptive neural network control for robotic systems through many years of effort and development [37]–[41]. In adaptive ILC, unknown parameter learning is performed in a finite period, and traditional adaptive neural networks are not suitable for direct application to such situations. To overcome the difficulty, neural networks based on adaptive differential learning have been developed to estimate uncertainties in finite–time operating systems [42]. In [42], Sun et al. propose a neural adaptive ILC scheme for continuous systems with arbitrary initial errors.

Under specific conditions, artificial neural networks are a powerful tool capable of approximating nonlinear functions accurately to any designed precision. Neural network–based controllers can perform better in the case of detailed system dynamics information [32]–[36]. Many studies have demonstrated the effectiveness of adaptive neural network control in robotics over the years [37]–[41]. However, in adaptive ILC, unknown parameter learning is usually performed in a finite period. Traditional adaptive neural networks are often not suitable for direct application in such situations. To overcome this challenge, a new approach has been developed that uses neural networks based on adaptive learning to estimate uncertainties in systems operating over finite periods [42]. In the work of Sun and colleagues [42], they proposed a neural network–based adaptive ILC scheme for a class of continuous systems with arbitrary initial errors. This represents a significant advance in applying artificial neural networks and ILC to face the challenges of dynamic systems with substantial uncertainties and require parameter tuning over short timescales.

ILC was initially developed as a direct feeding operation into open–loop systems [43]. Many research papers have proposed adaptive ILC methods to handle uncertain parameters and disturbances by adjusting the cybernetic gain [44]–[50]. In recent literature, many studies [51]–[57] have focused on iterative learning control design to track the trajectory of uncertain systems without requiring initial repositioning. Meanwhile, [56]–[57] developed a method to improve position–tracking accuracy through a minimum number of iterations. The notable point is that in [58], to achieve fast convergence of trajectory tracking in the initial iterations, it is impossible to increase the switching coefficient arbitrarily due to the limitation of the impact force, especially when the system has modeling errors or nonlinearities. The effectiveness of ILC depends on two factors. First, the design of the ILC must include a balance between accuracy and adaptability. Second, the ILC method is limited to large and complex problems. This depends on the updated ILC law, which has many elements that need improvement.

Developing ILC based on Lyapunov synthesis has introduced many advanced control technologies, such as robust or adaptive control, in the field of iterative learning control. A notable trend is the application of AC to design ILCs and vice versa to improve the dynamic quality of adaptive transitions [59]–[60]. Combining these two methods gave rise to adaptive iterative learning control (RAILC) theory, which overcomes characteristics that are difficult to achieve with a single control scheme. Therefore, RAILC is becoming a potential and promising research field in the future.

This article makes two important contributions:

- New control method: The paper has developed a new method that combines the advantages of several different control methods to create a new hybrid method to optimize tracking performance in repetitive tasks again. The proposed control method is not different from a conventional PD controller. Still, it has an adjustable ratio conversion technique and a forward learning term to predict the desired torque of the active mechanism. Using the Lyapunov method, a tuned iterative learning control scheme is introduced, ensuring the overall stability of the closed-loop system in the iterative domain.
- New mathematical model of robot controller: This article proposes a new mathematical model for deicing robot controllers and proves its effectiveness through experiments in industrial laboratories. This opens new perspectives in applying and improving robot control systems for repetitive tasks.

This paper is organized as follows: Section II describes a dynamic model of an n–link robot manipulator. In section III, RAILC is presented and its features are discussed. Lyapunov method is used to prove the asymptotic convergence of the proposed controller. Numerical simulation and experiment results of a three–link De–icing robot manipulator under the possible occurrence of uncertainties are provided to demonstrate the tracking control performance of the proposed RAILC system in section IV. Finally, conclusions are drawn in section V.

II. ROBOTIC DYNAMIC

In general, the dynamic of an n–link robot manipulator may be expressed in the following Lagrange form:

$$D(q^i(t))\ddot{q}^i(t) + C(q^i(t), \dot{q}^i(t))\dot{q}^i(t) + G(q^i(t), \dot{q}^i(t)) + \tau_a(t) = \tau^i(t) \tag{1}$$

With $t \in [0,\ t_f]$ denotes the time and $i \in N$ denotes the iteration, $q^i(t) \in R^n$, $\dot{q}^i(t) \in R^n$ and $\ddot{q}^i(t) \in R^n$ are the joint position, joint velocity, and joint acceleration variables vector respectively; $D(q^i(t)) \in R^{n\times n}$ is the inertia matrix, $C(q^i(t),\ \dot{q}^i(t)) \in R^n$ the coriolis–centripetal matrix; $G(q^i(t),\ \dot{q}^i(t)) \in R^n$ the gravity vector plus friction force vector, Bounded unknown disturbances are denoted by $\tau_a(t) \in R^n$ and the control input torque is $\tau^i(t) \in R^n$. In this paper, a new three–link De–icing robot manipulator, as shown in Fig. 1, is utilized to verify dynamic properties given in section IV.

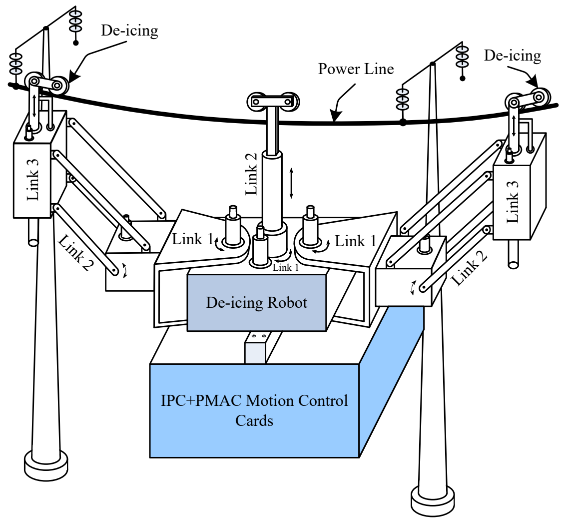


Fig. 1. Architecture of three–link De–icing robot manipulator

Property 1: The inertia matrix $D(q^i(t))$ is symmetric and positive definite. It is also bounded as a function of q: $m_1 I \le D(q^i(t)) \le m_2 I$. $m_1,\ m_2 > 0$

Property 2: $\dot{D}(q^i(t)) - 2C(q^i(t),\ \dot{q}^i(t))$ is a skew symmetric matrix. Therefore,

$y^T[\dot{D}(q^i(t)) - 2C(q^i(t),\ \dot{q}^i(t))]y = 0$ where y is a $n \times 1$ nonzero vector.

Assumption 1: The given desired joint trajectory $q_d(t)$ belongs to $C^2[0\ ,\ t_f]$, where $C^2[0\ ,\ t_f]$ is the set of twice continuously differentiable functions on$t \in [0,\ t_f]$.

Assumption 2: We also assume that time t is reset to zero at the starting point of each iteration. We assume that $q^i(0) = q_d(0)$ and $\dot{q}^i(0) = \dot{q}_d(0)$ for all $i \geq 1$.

III. RAILC Design

By linearzing the system (1) along the desired trajectory $q_d(t),\ \dot{q}_d(t),\ \ddot{q}_d(t)$ at the ith iterative, we obtain the following linear time–varying system according [21–22]:

$$\begin{aligned} D(t)\ddot{e}^i(t) + [C(t) + C_1(t)]\dot{e}^i(t) + R(t)e^i(t) \\ + n(\ddot{e}^i, \dot{e}^i, e^i, t) - \tau_a(t) \\ = S(t) - \tau^i(t) \end{aligned} \tag{2}$$

Where:

$$D(t) = D(q_d(t))$$

$$C(t) = C(q_d(t),\ \dot{q}_d(t))$$

$$C_1(t) = \left.\frac{\partial C}{\partial \dot{q}}\right|_{q_d(t),\ \dot{q}_d(t)} \dot{q}_d(t) + \left.\frac{\partial G}{\partial \dot{q}}\right|_{q_d(t),\ \dot{q}_d(t)}$$

$$R(t) = \left.\frac{\partial D}{\partial q}\right|_{q_d(t)} \ddot{q}_d(t) + \left.\frac{\partial C}{\partial q}\right|_{q_d(t),\ \dot{q}_d(t)} \dot{q}_d(t) + \left.\frac{\partial G}{\partial q}\right|_{q_d(t)}$$

$$\begin{aligned} S(t) = D\big(q_d(t)\big)\ddot{q_d}(t) + C\big(q_d(t),\quad \dot{q_d}(t)\big)q_d(t) \\ + G\big(q_d(t)\big) + \tau_a(t) \end{aligned}$$

$$e(t) = q_d(t) - q(t)$$

The term $n(\ddot{e}^i,\ \dot{e}^i,\ e^i,\ t)$ contains the higher order of terms $\ddot{e}^i(t)$, $\dot{e}^i(t)$ and $e^i(t)$, and it can be negligible. The control problem is to find a control law so that the position $q(t)$ can track specific commands $q_d(t)$. We construct controller as follows:

$$\tau^i = \tau_e^i + H^i \tag{3}$$

Where the first term $\tau_e^i = K_p^i(e^i(t)) + K_d^i(\dot{e}^i(t))$ is feedback PD control law with the following gain switching rule in [20]:

$$\begin{cases} K_p^{i+1} = \beta(i)K_p^i \\ K_d^{i+1} = \beta(i)K_d^i \qquad\qquad i = 0,\ 1,\ 2,\ \cdots\ ,\ N \\ \beta(i+1) > \beta(i) \end{cases} \tag{4}$$

Where K_p^i, K_d^i are the initial proportional and derivative control gain matrices that diagonal positive definite, K_p^{i+1}, K_d^{i+1} are the control gains of the ith iterative, $\beta(i) > 1$ is the gain switching factor. The gains adaptive laws in (4) are used to adjust the PD gains from iterative to iterative. And H^i is the initial predicted feed–forward control input to be computed at each iterative by a learning rule.

As it has been demonstrated in [20], the feedback PD control law with the gain switching factor in (4) plus the feed–forward learning control law with the input force profile, the convergence of system (2) is guaranteed. However, for the trajectory tracking convergence is fast in some initial iterative, we cannot increase the switching factor arbitrarily large because actuator forces are limited, especially when the system has modelling errors or nonlinearity. Hence, to deal this problem, we propose feed–forward control input $H^i(t)$ with a learning rule so that $H^i(t)$ converges to $R(t)$ for all $t \in [0,\ t_f]$ as follow:

$$H^{i+1}(t) = H^i(t) + \alpha\tau_e^i \tag{5}$$

At the initial stage of learning, the $H^i(t)$'s are set to zero. α is a positive constant often called a training factor. Therefore, for the ith and $(i+1)th$ iterations, applying the input (3) and (5) to system (2), we obtain an error equation as follows:

$$\begin{aligned} D(t)\ddot{e}^i(t) + [C(t) + C_1(t)]\dot{e}^i(t) + R(t)e^i(t) \\ = S(t) - \tau_e^i - H^i \end{aligned} \tag{6}$$

$$\begin{aligned} D(t)\ddot{e}^{i+1}(t) + [C(t) + C_1(t)]\dot{e}^{i+1}(t) \\ + R(t)e^{i+1}(t) \\ = S(t) - \tau_e^{i+1} - H^i - \alpha\tau_e^i \end{aligned} \tag{7}$$

To simplify the proof of stability, let $K_p^i = aK_d^i$ for the initial iteration, and define the filter errors as follow:

$$\tilde{x}^i(t) = \dot{e}^i(t) + ae^i(t) \tag{8}$$

Also, define $\delta\tilde{x}^i = \tilde{x}^{i+1} - \tilde{x}^i$ and $\delta e^i = e^{i+1} - e^i$. Then, from (8)

$$\delta\tilde{x}^i = \delta\dot{e}^i + a\delta e^i \tag{9}$$

From (4)–(8) and (9), one can obtain the following equation:

$$\begin{aligned} D\delta\dot{\tilde{x}}^i + (C + C_1 - aD + K_d^{i+1})\delta\tilde{x}^i + (R - a(C \\ + C_1 - aD))\delta e^i \\ = -(K_d^{i+1} + (\alpha - 1)K_d^i)\tilde{x}^i \end{aligned} \tag{10}$$

The following theorem can be proved.

Theorem: Consider an *n*–link robot manipulator dynamics represented by (1) satisfies property (1, 2) and assumption (1, 2). If the control laws of RAILC control system is designed as (3), the gain switching rule (4) and learning rule (5). The following should hold for all $t \in [0,\ t_f]$, we have

$$q^i(t) \xrightarrow{i\to\infty} q_d(t)$$

$$\dot{q}^i(t) \xrightarrow{i\to\infty} \dot{q}_d(t)$$

If the controller gains are selected so that the following relationships hold:

$$l_p = \lambda_{min}\Big((2-\alpha)K_d^i + 2C_i - 2aD\Big) > 0 \tag{11}$$

$$l_r = \lambda_{min}\left((2-\alpha)K_d^i + 2C + \frac{2R}{a} - 2\dot{C_1}/a\right) > 0 \tag{12}$$

$$l_p l_r \geq \left\|\frac{R}{a} - (C + C_1 - aD\right\|_{max}^2 \tag{13}$$

Where $\lambda_{min}(A)$ is the minimum eigenvalue of matrix A, and $\|M\|_{max} = max\|M(t)\|$ for $t \in [0, t_f]$. Here, $\|M\|$ represents the Euclidean norm of M.

Proof: We select a performance index $V^i(t)$ as follow:

$$V^i = \int_0^t e^{-\rho\tau}\, \tilde{x}^{i^T} Q \tilde{x}^i d\tau \geq 0 \tag{14}$$

Thus $\beta(i) > 1$ according (4) so we have $K_d^{i+1} > K_d^i$ and $\alpha \leq 1$ is a positive constant, so $Q = K_d^{i+1} + (\alpha - 1)K_d^i) > 0$. From the definition of V^i, for the $(i+1)th$ iteration, we can get

$$V^{i+1} = \int_0^t e^{-\rho\tau}\, \tilde{x}^{i+1^T} Q \tilde{x}^{i+1} d\tau \tag{15}$$

Let $\varDelta V^i = V^{i+1} - V^i$ then from (14), (15) and (10), we obtain:

$$\begin{aligned}\varDelta V^i &= \int_0^t e^{-\rho\tau}\,(\delta\tilde{x}^{i^T} Q \delta\tilde{x}^i + 2\delta\tilde{x}^{i^T} Q \tilde{x}^i) d\tau \\ &= \int_0^t e^{-\rho\tau}\,\delta\tilde{x}^{i^T} Q\delta\tilde{x}^i d\tau - 2\int_0^t e^{-\rho\tau}\,\delta\tilde{x}^{i^T} D\delta\dot{\tilde{x}}^i d\tau \\ &\quad - 2\int_0^t e^{-\rho\tau}\,\delta\tilde{x}^{i^T}((C + C_1 - aD + K_d^{i+1})\delta\tilde{x}^i + (R \\ &\quad - a(C + C_1 - aD))\delta e^i) d\tau\end{aligned} \tag{16}$$

By applying the partial integration, from assumption 2, and property 2, we have

$$\begin{aligned}\int_0^t e^{-\rho\tau}\,\delta\tilde{x}^{i^T} D\delta\dot{\tilde{x}}^i d\tau &= e^{-\rho\tau}\delta\tilde{x}^{i^T} D\delta\tilde{x}^i\Big|_0^t - \int_0^t (e^{-\rho\tau}\,\delta\tilde{x}^{i^T} D)'\delta\tilde{x}^i d\tau \\ &= e^{-\rho\tau}\delta\tilde{x}^{i^T}(t) D(t)\delta\tilde{x}^i(t) \\ &\quad + \rho\int_0^t e^{-\rho\tau}\,\delta\tilde{x}^{i^T} D\delta\tilde{x}^i d\tau - \int_0^t e^{-\rho\tau}\,\delta\tilde{x}^{i^T} D\delta\dot{\tilde{x}}^i d\tau \\ &\quad - 2\int_0^t e^{-\rho\tau}\,\delta\tilde{x}^{i^T} C\delta\tilde{x}^i d\tau\end{aligned} \tag{17}$$

Substituting (17) into (16) and $Q = K_d^{i+1} + (\alpha - 1)K_d^i)$ yields

$$\begin{aligned}\Delta V^i &= -e^{-\rho\tau}\delta\tilde{x}^{iT}(t) D(t)\delta\tilde{x}^i(t) \\ &\quad - \rho\int_0^t e^{-\rho\tau}\delta\tilde{x}^{iT} D\delta\tilde{x}^i d\tau - 2\int_t^t e^{-\rho\tau}\delta\tilde{x}^{iT}\big(R \\ &\quad - a(C + C_1 - aD)\big)\delta e^i d\tau \\ &\quad - \int_t^t e^{-\rho\tau}\delta\tilde{x}^{iT}(K_d^{i+1} - (\alpha - 1)K_d^i + 2C_1 \\ &\quad - 2aD)\delta\tilde{x}^i d\tau\end{aligned} \tag{18}$$

From (4), we have

$$\begin{aligned}\int_0^t e^{-\rho\tau}\,\delta\tilde{x}^{i^T} K_d^{i+1}\delta\tilde{x}^i d\tau &= \beta(i+1)\int_0^t e^{-\rho\tau}\,\delta\tilde{x}^{i^T} K_d^i\delta\tilde{x}^i d\tau \\ &\geq \int_0^t e^{-\rho\tau}\,\delta\tilde{x}^{i^T} K_d^i\delta\tilde{x}^i d\tau\end{aligned} \tag{19}$$

Substituting (9) into (18) and noticing (19), we obtain

$$\begin{aligned}\varDelta V^i &\leq -e^{-\rho\tau}\delta\tilde{x}^{i^T}(t) D(t)\delta\tilde{x}^i(t) - \rho\int_0^t e^{-\rho\tau}\,\delta\tilde{x}^{i^T} D\delta\tilde{x}^i d\tau \\ &\quad - \int_0^t e^{-\rho\tau}\,\delta\dot{e}^{i^T}((2\alpha)K_d^i + 2C_1 - 2aD)\delta\dot{e}^i d\tau \\ &\quad - 2a\int_0^t e^{-\rho\tau}\,\delta e^{i^T}((2-\alpha)K_d^i + 2C_1 \\ &\quad - 2aD)\delta\dot{e}^i d\tau - 2\int_0^t e^{-\rho\tau}\,\delta\dot{e}^{i^T}(R - a(C + C_1 \\ &\quad - aD)\delta e^i d\tau - a^2\int_0^t e^{-\rho\tau}\,\delta e^{i^T}((2-\alpha)K_d^i + 2C_1 \\ &\quad - 2aD)\delta e^i d\tau - 2a\int_0^t e^{-\rho\tau}\,\delta e^{i^T}(R - a(C + C_1 \\ &\quad - aD)\delta e^i d\tau\end{aligned} \tag{20}$$

By applying the partial integration the again gives

$$\begin{aligned}\int_0^t e^{-\rho\tau}\,\delta e^{i^T}\left((2-\alpha)K_d^i + 2C_1 - 2aD\right)\delta\dot{e}^i d\tau &= e^{-\rho\tau}\delta e^{i^T}((2-\alpha)K_d^i + 2C_1 - 2aD)\delta e^i\Big|_0^t \\ &\quad + \rho\int_0^t e^{-\rho\tau}\,\delta e^{i^T}((2-\alpha)K_d^i + 2C_1 - 2aD)\delta e^i d\tau \\ &\quad - \int_0^t e^{-\rho\tau}\,\delta\dot{e}^{i^T}((2-\alpha)K_d^i + 2C_1 - 2aD)\delta e^i d\tau \\ &\quad + 2\int_0^t e^{-\rho\tau}\,\delta e^{i^T}(a\dot{D} - \dot{C}_1)\delta e^i d\tau\end{aligned} \tag{21}$$

Therefore

$$\begin{aligned}\varDelta V^i &\leq -e^{-\rho\tau}\delta\tilde{x}^{i^T} D\delta\tilde{x}^i - \rho\int_0^t e^{-\rho\tau}\,\delta\tilde{x}^{i^T} D\delta\tilde{x}^i d\tau - ae^{-\rho\tau}\delta e^{i^T}((2 \\ &\quad - \alpha)K_d^i + 2C_1 - 2aD)\delta e^i - \rho a\int_0^t e^{-\rho\tau}\,\delta e^{i^T}((2 \\ &\quad - \alpha)K_d^i + 2C_1 - 2aD)\delta e^i d\tau - \int_0^t e^{-\rho\tau}\, w d\tau\} \\ &\leq -e^{-\rho\tau}\delta\tilde{x}^{i^T} D\delta\tilde{x}^i - ae^{-\rho\tau}\delta e^{i^T} l_p\delta e^i \\ &\quad - \rho\int_0^t e^{-\rho\tau}\,\delta\tilde{x}^{i^T} D\delta\tilde{x}^i d\tau \\ &\quad - \rho a\int_0^t e^{-\rho\tau}\,\delta e^{i^T} l_p\delta e^i d\tau - \int_0^t e^{-\rho\tau}\, w d\tau\end{aligned} \tag{22}$$

Where

$$\begin{aligned}w &= \delta\dot{e}^{i^T}\left((2-\alpha)K_d^i + 2C_1 - 2aD\right)\delta\dot{e}^i \\ &\quad + 2a\delta\dot{e}^{i^T}\left(\frac{R}{a} - (C + C_1 - aD)\right)\delta e^i \\ &\quad + a^2\delta e^{i^T}\left((2-\alpha)K_d^i + \frac{2R}{a} + 2C - \frac{2\dot{C}_1}{a}\right)\delta e^i\end{aligned} \tag{23}$$

Let $P = R/a - (C + C_1 - aD)$. Then from (11) and (12), we obtain

$$w \geq l_p\|\delta\dot{e}\|^2 + 2a\delta\dot{e}^T P\delta e + a^2 l_r\|\delta e\|^2 \tag{24}$$

By applying the Cauchy–Schwartz inequality, we obtain

$$\delta\dot{e}^T P\delta e \geq -\|\delta\dot{e}\|\|P\|_{max}\|\delta e\| \tag{25}$$

From (11)–(13) we obtain

$$w = l_p\|\delta\dot{e}\|^2 - 2a\|\delta\dot{e}\|\|P\|_{max}\|\delta e\| + a^2 l_r\|\delta e\|^2$$
$$= l_p\left(\|\delta\dot{e}\| - \frac{a}{l_p}\|P\|_{max}\|\delta e\|\right)^2 + a^2(l_p - \frac{1}{l_r}\|P\|_{max}^2)\|\delta e\|^2 \geq 0 \tag{26}$$

According to the properties 1 and (26), based on (22), it can be ensured that $\Delta V^i \leq 0$, therefore $V^{i+1} \leq V^i$. From the definition K_d^i is a positive definite matrix. From the definition of V^i, $V^i \geq 0$, and V^i is bounded. As a result, $\tilde{x}(t) \to 0$ when $i \to \infty$. Because $e^i(t)$ and $\dot{e}^i(t)$ are two independent variables, and a is a positive constant. Thus, if $i \to \infty, e^i(t) \to 0$ and $\dot{e}^i(t) \to 0$ for $t \in [0,\ t_f]$.

Finally, the following conclusions hold for $t \in [0,\ t_f]$

$$q^i(t) \xrightarrow{i\to\infty} q_d(t)$$
$$\dot{q}^i(t) \xrightarrow{i\to\infty} \dot{q}_d(t)$$

According to this analysis, we can be seen that the adaptive RAILC control method guarantee that the tracking errors converge arbitrarily close to zero as the number of iterations increases. The following case is demonstrated based on simulation and experimental results for this conclusion.

IV. Simulation and Experimental Results

A. Simulation Results

A three–link De–icing robot manipulator as shown in Fig. 1 is utilized in this paper to verify the effectiveness of the proposed control scheme. The detailed system parameters of this robot manipulator are given as: link mass m_1, m_2, m_3 (kg), lengths l_1, l_2 (m), angular positions q_1, q_2 (rad) and displacement position d_3 (m). The parameters for the equation of motion (1) can be represented as follows:

$$D(q) = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix}$$

$$D_{11} = \frac{9}{4m_1 l_1} + m_2\left(\frac{1}{4c_2 l_2} + l_1^2 + l_2 l_1(c_1^2 - s_1^2)\right) + m_3(c_2 l_2^2 + l_2^2 + 2c_2 l_1 l_2$$

$$D_{22} = 1/4m_2 l_2^2 + m_3 l_2^2 + 4/3m_1 l_1^2$$

$$D_{23} = M_{32} = m_3 c_2 l_2$$

$$D_{33} = m_3$$

$$D_{12} = D_{13} = D_{21} = D_{31} = 0$$

$$C(\dot{q}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}$$

$$C_{11} = -8m_2\, l_1\, l_2\, c_1\, s_1\,(q_1 + (-1/(2m_2\, s_2\, c_2\, l_2^2\,) + m_3\,(-2s_2\, c_2\, l_2^2 - 2s_2\, l_1\, l_2)\dot{q}_2$$

$$C_{21} = (-1/2m_2 s_2 c_2 l_2^2 + m_3(-2s_2 c_2 l_2^2 - 2s_2 l_1 l_2)\dot{q}_1$$

$$C_{22} = -m_3 s_2 l_2 \dot{d}_3$$

$$C_{23} = -2m_3 s_2 l_2 \dot{q}_2$$

$$C_{32} = -m_3 s_2 l_2 \dot{q}_2$$

$$C_{12} = C_{13} = C_{31} = C_{33} = 0$$

$$G(q) = \begin{bmatrix} (1/2c_1 c_2 l_2 + c_1 l_1)m_2 g \\ (-1/2s_1 s_2 l_2 m_2 + c_2 l_2 m_3)g \\ m_3 g \end{bmatrix} \tag{27}$$

Where $q \in R^3$and the shorthand notations $c_1 = cos(\,q_1)$, $c_2 = cos(\,q_2)$, $s_1 = sin(\,q_1)$ and $s_2 = sin(\,q_2)$ are used.

For the convenience of the simulation, the nominal parameters of the robotic system are given such as $m_1 = 3(kg)$, $m_2 = 2(kg)$, $m_3 = 2.5(kg)$, $l_1 = 0.14\ (m)$, $l_2 = 0.32\ (m)$, and $g = 9.8\ (m/s^2)$ and the initial conditions $q_1(0) = 0$, $q_2(0) = 1$, $d_3(0) = 0$, $\dot{q}_1(0) = 0$, $\dot{q}_2(0) = 0$, $\dot{d}_3(0) = 0$. The desired reference trajectories are $q_{d1}(t) = sin(\,2t)$, $q_{d2}(t) = cos(\,2t)$ and $d_{d2}(t) = sin(\,2t)$, respectively.

The most important parameters that affect the control performance of the robotic system are the bounded unknown disturbance $\tau_a(t)$ which consist of the external disturbance term $f_1(t)$ the friction term$f_2(\dot{q})$, in simulation, the external disturbance situation occurring at fifth the iteration are considered. The disturbance situation is that external forces are injected into the robotic system, and their shapes are expressed as follows:

$$f_1(t) = [5\,sin(\,5t) \quad 5\,sin(\,5t) \quad 5\,sin(\,5t)]^T \tag{28}$$

In addition, friction forces are also considered in this simulation and are given as

$$f_2(\dot{q}) = \left[20\dot{q}_1 + 0.8sgn(\dot{q}_1)4\dot{q}_2 + 2\,sgn(\dot{q}_2)4\dot{d}_3 + 2\,sgn(\dot{d}_3)\right]^T \tag{29}$$

In order to exhibit the superior control performance of RAILC system, the adaptive switching learning PD control system (ASL–PD) is represented in [20] for comparison.

The simulation results of ASL–PD system, the responses of position at first and fifteenth iteration and tracking error from iteration to iteration are depicted in Fig. 3 (a), (b), (c), Fig. 4 (a), (b), (c) and Fig. 5 (a), (b), (c). Now, the proposed RAILC control system depicted in Fig. 2 is applied to control the three–link De–icing robot manipulator for comparison. The simulation results of position responses and tracking error from iteration to iteration are depicted in Fig. 6 (a), (b), (c) and Fig. 7 (a), (b), (c). Table I show that the tracking performance of the proposed system from the initial iteration to fifteenth is obvious. Therefore, the comparison of their method and our method demonstrated fast convergence rate with the proposed control method.

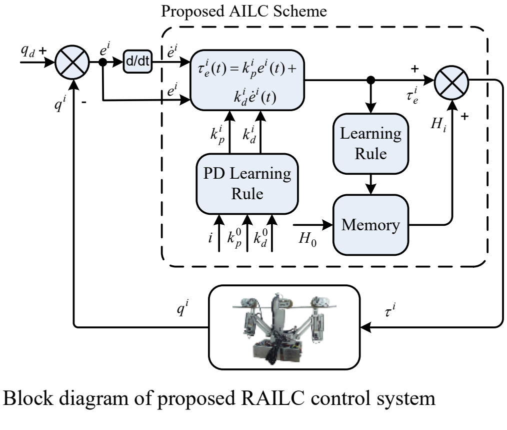


Fig. 2. Block diagram of proposed RAILC control system

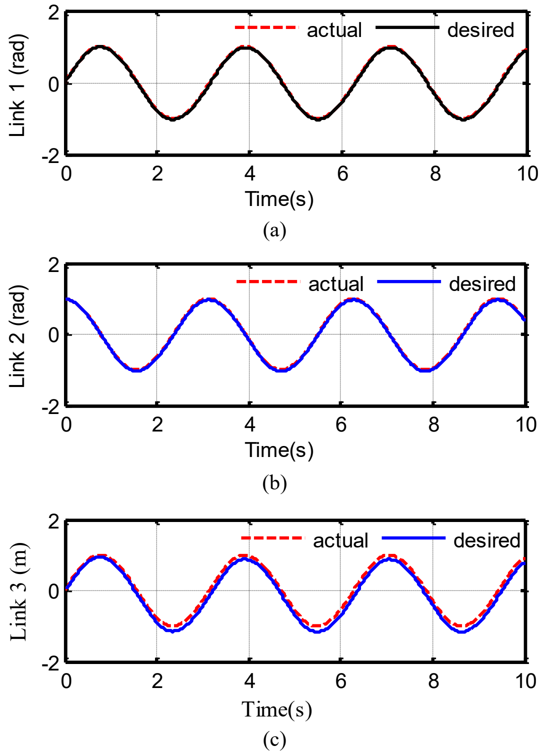


Fig. 3. Simulated position responses of the ASL–PD and proposed RAILC control system for joints 1, 2 and 3 at first iteration

In two simulation situations, The PD control gain was set to be the same as follows:

$$K_p^i = K_d^i = diag\{30,\ 30,\ 30\}$$

$$K_p^{i+1} = 2iK_p^i,\ K_d^{i+1} = 2iK_d^i,$$
$$\beta = 0.75,\ \alpha = 0.8$$

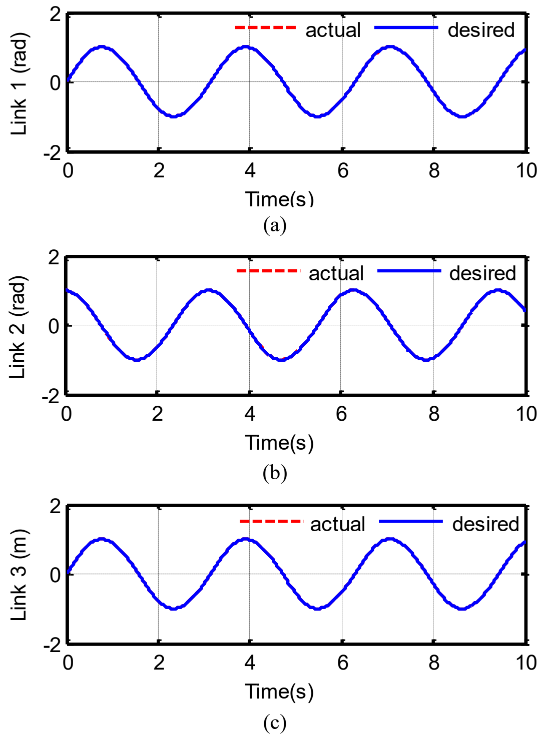


Fig. 4. Simulated position responses of the ASL–PD control system for joints 1, 2 and 3 at fifteenth iteration

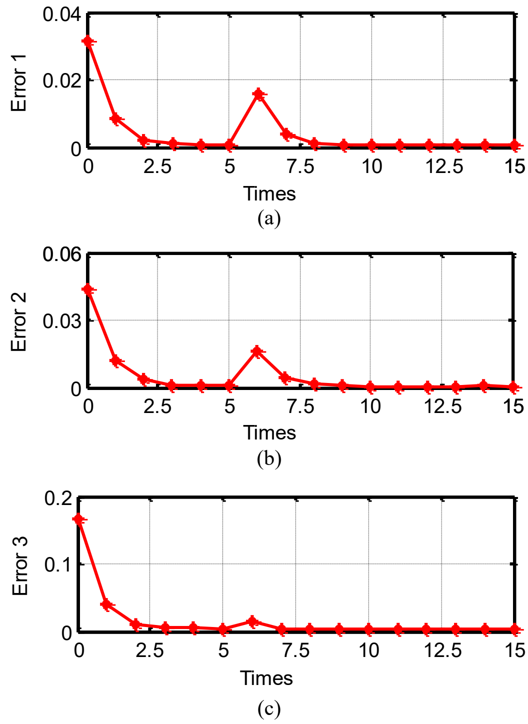


Fig. 5. Tracking error of ASL–PD control system from iteration to iteration

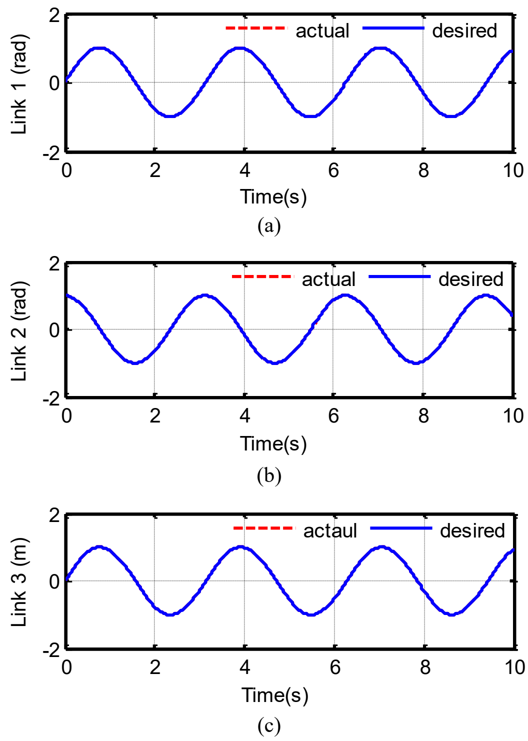
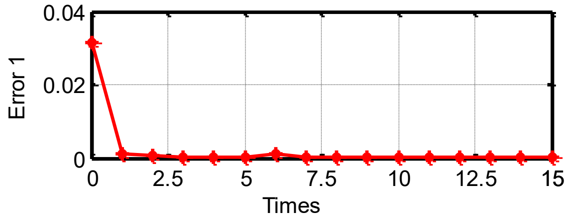


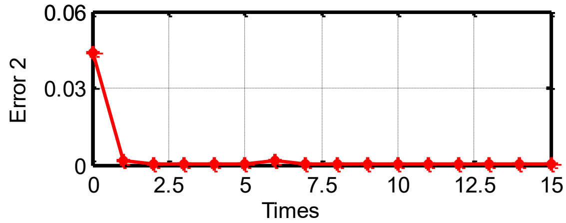
Fig. 6. Simulated position responses of the proposed RAILC control system for joints 1, 2 and 3 at fifteenth iteration

TABLE I. MAXIMUM TRACKING ERRORS FROM ITERATION TO ITERATION

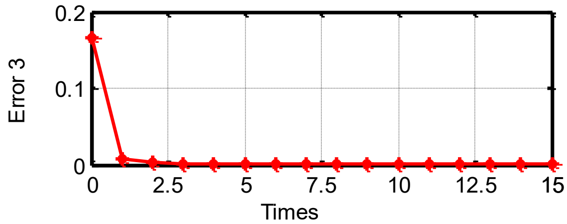
	Iterative	*0*	*5*	*10*	*15*
ASL–PD	Max$\lvert e_1^i\rvert$	0.0316	0.0007	0.0008	0.0007
	Max$\lvert e_2^i\rvert$	0.0437	0.0008	0.0005	0.0006
	Max$\lvert e_3^i\rvert$	0.1661	0.0040	0.0028	0.0023
RAILC	Max$\lvert e_1^i\rvert$	0.0316	0.0001	5.4x10^{-5}	3.4x10^{-5}
	Max$\lvert e_2^i\rvert$	0.0437	0.0002	2.5x10^{-5}	1.3x10^{-5}
	Max$\lvert e_3^i\rvert$	0.1661	0.0003	0.0001	5,6x10^{-5}



(a)



(b)



(c)

Fig. 7. Tracking error of proposed RAILC control system from iteration to iteration

Based on the data in the Table I, we can see the detailed differences between ASL-PD and RAILC methods in minimizing errors from one iteration to another.

- First, regarding the max tracking error rate (Max), ASL-PD usually has a more considerable Max value than RAILC at each iteration. For example, in the second iteration, the maximum value of ASL-PD ranges from 0.0008 to 0.0007, while RAILC only ranges from 5.4×10^{-5} to 3.4×10^{-5}. This means that RAILC tends to reduce errors more effectively than ASL-PD.
- Second, the fluctuations between iterations are also worth noting. While RAILC reduces error consistently with each iteration, ASL-PD can exhibit significant volatility. For example, in the third iteration, the maximum value of ASL-PD increases from 0.0005 to 0.0040, while RAILC only increases from 2.5×10^{-5} to 5.6×10^{-5}. This means ASL-PD may not be stable and may need tuning to achieve better performance.
- Finally, the convergence speed is also a notable point. RAILC tends to converge faster than ASL-PD, as shown by the rapid and stable error reduction over iterations. Meanwhile, ASL-PD may require more iterations to achieve similar error reduction.

In summary, RAILC is a more stable and effective method for minimizing errors from one iteration to another.

B. Experimental Results

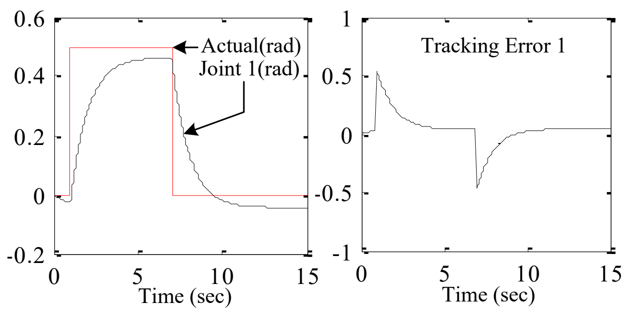
The experimental parameters of the proposed RAILC control system are selected:

$$K_p^i = K_d^i = diag\{100,\ 100,\ 100\}$$

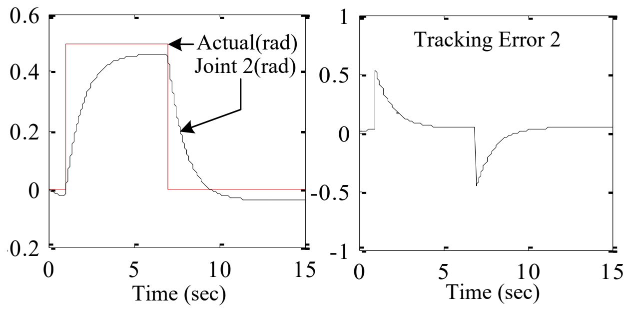
$$K_p^{i+1} = 2iK_p^i, K_d^{i+1} = 2iK_d^i$$

$$\beta = 0.75, \alpha = 0.8$$

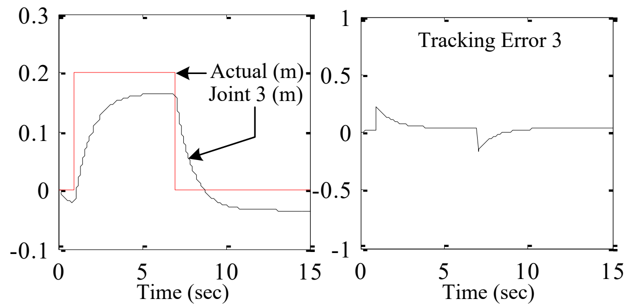
In this section, the control objective is to control the each joint angles of a three–link De–icing robot manipulator to move periodically for a periodic step commands and the initial conditions of system are given as $q_1(0) = 0(rad)$, $q_2(0) = 0(rad)$, $d_3(0) = 0(m)$. Finally, the experimental position and tracking error responses results of the proposed RAILC control system from first iteration to fifth iteration are depicted in Fig. 8 (a), (b) and (c), Fig. 9 (a), (b), and (c). According to these experimental and simulation results of proposed RAILC control system due to sinusoidal and periodic step reference commands indicate that the high accuracy tracking position responses can be achieved by using the proposed RAILC control system for difference reference commands and tracking errors responses of the RAILC scheme decrease with the increase of the iteration number under wide range of external disturbance.



(a)



(b)



(c)

Fig. 8. Experimental position responses and tracking error of the proposed RAILC control system at joints 1, 2 and 3 at first iteration

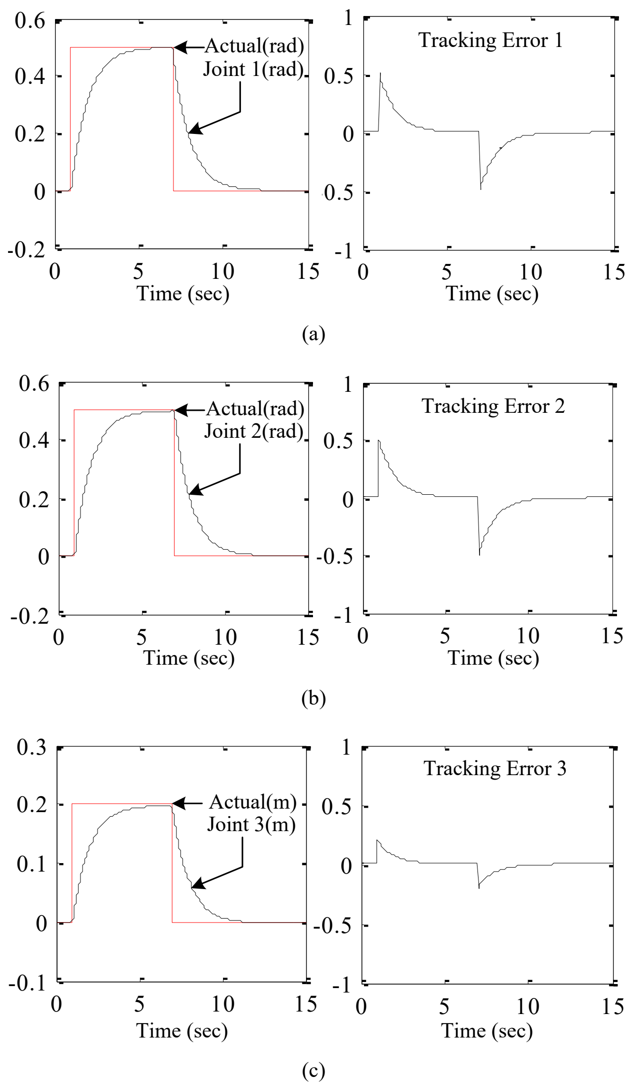


Fig. 9. Experimental position responses and tracking error of the proposed RAILC control system at joints 1, 2 and 3 at fifth iteration

The hardware block diagram of the control system is implemented to verify the effectiveness of the proposed methodologies and is shown in Fig. 10(a). Each joint of manipulator is derived by the "EC─**" type MAXON DC servo motors, which is designed by Switzerland Company, and each this motor contains an encoder. Digital filter and frequency multiplied by circuits are built into the encoder interface circuit to increase the precision of position feedback. The DCS303 is a digital DC servo driver developed with DSP to control the DC servo motor. The DCS303 is a micro–size brush DC servo drive. It is an ideal choice for this operating environment. Two DC servo motor motion control cards are installed in the industrial personal computer, in which, a 6–axis DC servo motion control card is used to control the joint motors and a 4–axis motion control card is used to control the drive motors. Each card includes multi–channels of digital/analog and encoder interface circuits. The name of model is DMC2610 with a PCI interface connected to the IPC. The DMC2610 implements the proposed program and execute in the real time. Considering that the control sampling rate $Ts = 1$ ms is too demanding for the hardware implementation, $Ts = 10$ ms is thus considered here.

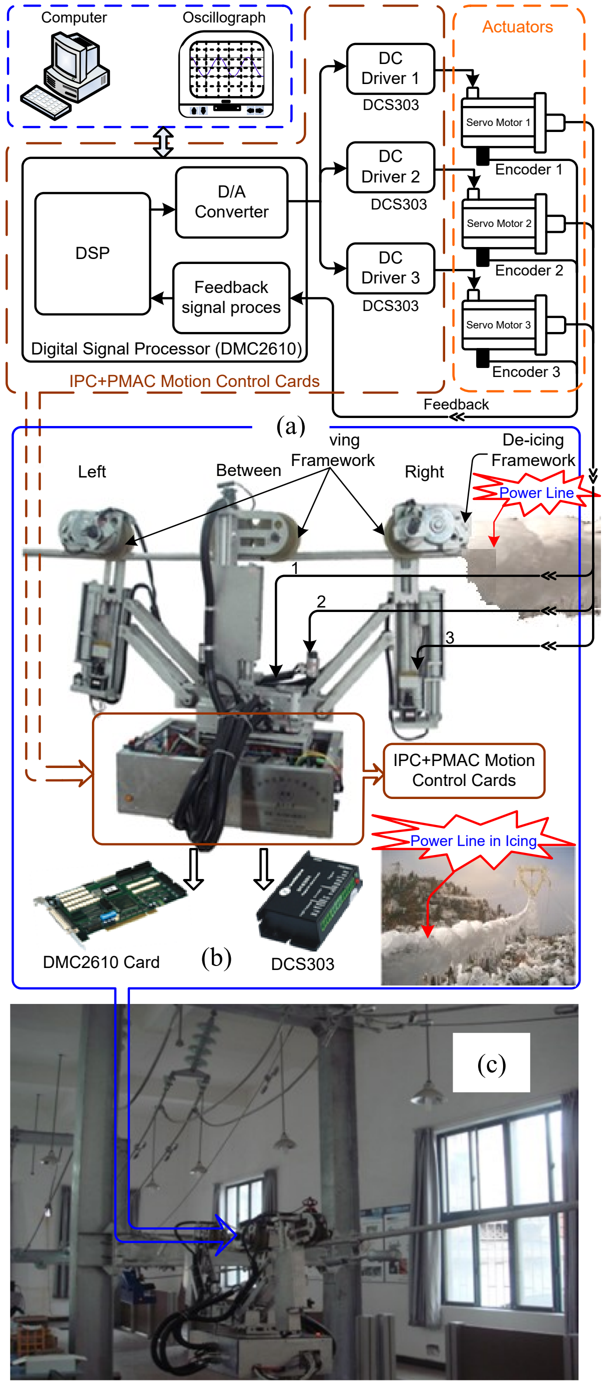


Fig. 10. IPC–based De–icing robot position control system a) Block diagram of three–link De–icing robot manipulator control system, b) image of practical control system, c) image of special robot laboratory of power industry

An image of a practical experiment control system for De–icing robot consists of three manipulators and is shown in Fig. 10(b). The left and right manipulators have three–link with two revolute joints and a prismatic joint. End–effectors of each manipulator have attached the motion structure to move the De–icing robot on the power line and the snow cleaning device. During normal operating conditions, the left and right manipulators are only in operation. The between manipulator has only two joints with a revolute joint and a prismatic joint. It only works when the De–icing robot voids obstacles on the power line. In general, the operation of De–

icing robot is very complex. In this paper, we consider only the three–link De–icing robot manipulator for proposed methodologies while the other manipulator is the same.

V. Conclusion

This paper has successfully implemented an RAILC scheme to control the position of three–link De–icing robot manipulator for achieving desired position control. All the dynamically system may be unknown. A new control method is a combination of advantages some other method into a hybrid one as explained above. By using Lyapunov theorem, the asymptotic convergence of the closed–loop control system can be ensured whether or not the uncertainties occur. Simulation and experimental results of a three–link De–icing robot manipulator via various existing control methods including ASL–PD and ILC control were also applied in this paper to compare and display the manipulative performance of the proposed control system. According to the these result, it is shown that the desired position tracking and tracking errors response of the RAILC scheme decrease with the increase of the iteration number under wide range of external disturbance. The main purpose of the paper is to construct a simple scheme, easy implementation, fast convergence.

In de-icing robotics, using RAILC offers excellent potential to improve the performance and stability of systems. Because the variable and complex actual of the ice environment in overhead power lines poses significant challenges, RAILC can help autonomous robots adapt, learn, and improve over time. However, RAILC still has the disadvantage of having fixed learning parameters. This means that each area of the robot needs to choose reasonable and suitable parameters for the changing conditions in that area. In the future, research and development in this field should focus on combining RAILC with control methods and parameter learning capabilities to enhance the effectiveness and reliability of de-icing robots. This will play an important role in developing intelligent automation systems that operate in harsh environments and contribute to the industry's sustainable development.

References

[1] J. Li, Z. Wu, C. Wen, and Y. Liu, "Stabilization involving condition-based adaptive control of an uncertain wave equation for the vibration of a flexible string," *Automatica*, vol. 161, p. 111488, March 2024.

[2] Y. Jiang, L. Liu, and G. Feng, "Fully distributed adaptive control for output consensus of uncertain discrete-time linear multi-agent systems," *Automatica*, vol. 162, p. 111531, April 2024.

[3] L. H. Wang and Y. Li, "Adaptive control of robot manipulators with closed architecture," *Automatica*, vol. 162, p. 111040, April 2024.

[4] Z. Liu, F. Wang, Y. Zhang, X. Chen, and C. L. P. Chen, "Adaptive Tracking Control for A Class of Nonlinear Systems With a Fuzzy Dead–Zone Input," in *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 1, pp. 193–204, Feb. 2015.

[5] X. Yu, Z. Hou, M. M. Polycarpou, and L. Duan, "Data–Driven Iterative Learning Control for Nonlinear Discrete–Time MIMO Systems," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 3, pp. 1136–1148, March 2021.

[6] L. Yang, Y. Li, D. Huang, J. Xia, and X. Zhou, "Spatial Iterative Learning Control for Robotic Path Learning," in *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 5789–5798, July 2022.

[7] D. X. Ba, N. T. Thien, and J. Bae, "A Novel Iterative Second–Order Neural–Network Learning Control Approach for Robotic Manipulators," in *IEEE Access*, vol. 11, pp. 58318–58332, 2023.

[8] D. Huang, Y. Chen, D. Meng, and P. Sun, "Adaptive iterative learning control for high–speed train: A multi–agent approach," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 7, pp. 4067-4077, 2021.

[9] Y. Chen, D. Huang, Y. Li, and X. Feng, "A Novel Iterative Learning Approach for Tracking Control of High–Speed Trains Subject to Unknown Time–Varying Delay," in *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 1, pp. 113–121, Jan. 2022.

[10] G. Liu and Z. Hou, "Adaptive Iterative Learning Control for Subway Trains Using Multiple–Point–Mass Dynamic Model Under Speed Constraint," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1388–1400, March 2021.

[11] J. Zheng and Z. Hou, "Model Free Adaptive Iterative Learning Control Based Fault–Tolerant Control for Subway Train With Speed Sensor Fault and Over–Speed Protection," *IEEE Transactions on Automation Science and Engineering*, vol.21, no.1, pp.168–180, 2024.

[12] S. Airmoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robotics by learning," *J. Robot. Syst*., vol. 12, no. 2, pp. 123–140, 1984.

[13] P. Bondi, G. Casalino, and L. Gambardella, "On the iterative learning control theory for robotic manipulators," *IEEE J. Robot. Autom*., vol. 4, no. 1, pp. 14–22, Feb. 1998.

[14] P. Jiang, L. C. A. Bamforth, Z. Feng, J. E. Baruch, and Y. Chen, "Indirect iterative learning control for a discrete visual servo without a camera–robot model," *IEEE Trans. Syst. Man Cybern. B Cybern*., vol. 37, no. 4, pp. 863–876, Aug. 2007.

[15] Y. Chen, D. Huang, N. Qin, and Y. Zhang, "Adaptive Iterative Learning Control for a Class of Nonlinear Strict–Feedback Systems With Unknown State Delays," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 9, pp. 6416–6427, Sept. 2023.

[16] Z. Chen, Y. Hou, R. Huang, and Q. Cheng, "Neural network compensator-based robust iterative learning control scheme for mobile robots nonlinear systems with disturbances and uncertain parameters," *Applied Mathematics and Computation*, vol. 469, p. 128549, 2024.

[17] L. Wu, Y. Yu, Z. Lin, Q. Yan, and X. Liu, "Time–Varying Boundary Layer Based Iterative Learning Control for Nonlinearly Parametric Time–Delay Systems With Arbitrary Initial Errors and Iteration–Varying Reference Trajectories," in *IEEE Access*, vol. 9, pp. 90642–90655, 2021.

[18] S. L. Chen, S. M. Hsieh, and T. Q. Ta, "Iterative learning contouring control for five-axis machine tools and industrial robots," *Mechatronics*, vol. 94, p. 103030, 2023.

[19] Y. Chen and C. T. Freeman, "Iterative learning control for piecewise arc path tracking with validation on a gantry robot manufacturing platform," *ISA Transactions*, vol. 139, pp. 650-659, 2023.

[20] Y. H. Lee, S. C. Hsu, T. Y. Chi, Y. Y. Du, J. S. Hu, and T. C. Tsao, "Industrial robot accurate trajectory generation by nested loop iterative learning control," *Mechatronics*, vol. 74, p. 102487, 2021.

[21] A. Tayebi, "Adaptive iterative learning control for robot manipulators," *Automatica*, vol. 40, no. 7, pp. 1195–1203, Jul. 2004.

[22] C.–J. Chien and A. Tayebi, "Further results on adaptive iterative learning control of robot manipulators," *Automatica*, vol. 44, no. 3, pp. 830–837, Mar. 2008.

[23] X. X. He, H. Zhuang, and D. Zhang, "Pulse neural network–based adaptive iterative learning control for uncertain robots," *Neural Comput. Appl*., vol. 23, no. 7, pp. 1885–1890, Dec. 2013.

[24] F. Cao and J. Liu, "An adaptive iterative learning algorithm for boundary control of a coupled ODE–PDE two–link rigid–flexible manipulator," *J. Franklin Inst*., vol. 354, no. 1, pp. 277–297, 2017.

[25] X. Li, Y.–H. Liu, and H. Yu, "Iterative learning impedance control for rehabilitation robots driven by series elastic actuators," *Automatica*, vol. 90, pp. 1–7, Apr. 2018.

[26] J.–X. Xu and R. Yan, "On initial conditions in iterative learning control," *IEEE Trans. Autom. Control*, vol. 50, no. 9, pp. 1349–1354, Sep. 2005.

[27] Y. Wang, W. Wu, X. Lou, and D. Görges, "Iterative learning control of an Euler-Bernoulli beam with time-varying boundary disturbance," *Computers & Mathematics with Applications*, vol. 162, pp. 145-154, 2024.

[28] L. Wang, Z. Huangfu, R. Li, X. Wen, Y. Sun, and Y. Chen, "Iterative learning control with parameter estimation for non-repetitive time-

varying systems," *Journal of the Franklin Institute*, vol. 361, no. 3, pp. 1455-1466, 2024.

[29] Y. Kong, X.-D. Li, and X. Li, "Iterative learning control approach for linear discrete time-varying systems with different initial time points," *Journal of the Franklin Institute*, vol. 361, no. 6, p. 106702, 2024.

[30] X. Jin, "Iterative learning control for non–repetitive trajectory tracking of robot manipulators with joint position constraints and actuator faults," *Int. J. Adapt. Control Signal Process*., vol. 31, no. 6, pp. 859–875, Jun. 2017.

[31] O. Zhang, W. Yao, D. Du, C. Wu, J. Liu, L. Wu, and Y. Sun, "Trajectory optimization and tracking control of free-flying space robots for capturing non-cooperative tumbling objects," *Aerospace Science and Technology*, vol. 143, p. 108718, 2023.

[32] Q. Yang, Q. Yan, J. Cai, J. Tian, and X. Guan, "Neural Network–Based Error–Tracking Iterative Learning Control for Tank Gun Control Systems With Arbitrary Initial States," in *IEEE Access*, vol. 8, pp. 72179–72187, 2020.

[33] G. Liu and Z. Hou, "RBFNN–Based Adaptive Iterative Learning Fault–Tolerant Control for Subway Trains With Actuator Faults and Speed Constraint," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 9, pp. 5785–5799, Sept. 2021.

[34] D. Li, C. L. P. Chen, Y.–J. Liu, and S. Tong, "Neural network controller design for a class of nonlinear delayed systems with time–varying fullstate constraints," *IEEE Trans. Neural Netw. Learn. Syst*., vol. 30, no. 9, pp. 2625–2636, Sep. 2019.

[35] L. Liu, Y.–J. Liu, and S. Tong, "Fuzzy–based multierror constraint control for switched nonlinear systems and its applications," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 8, pp. 1519–1531, Aug. 2018.

[36] T. Gao, Y.–J. Liu, L. Liu, and D. Li, "Adaptive neural network–based control for a class of nonlinear pure–feedback systems with time–varying full state constraints," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 5, pp. 923–933, Sep. 2018.

[37] L. Wu, Q. Yan, and J. Cai, "Neural Network–Based Adaptive Learning Control for Robot Manipulators With Arbitrary Initial Errors," in *IEEE Access*, vol. 7, pp. 180194–180204, 2019.

[38] X. Zhou, Y. Tian, and H. Wang, "Neural network state observer-based robust adaptive fault-tolerant quantized iterative learning control for the rigid-flexible coupled robotic systems with unknown time delays," *Applied Mathematics and Computation*, vol. 430, p. 127286, 2022.

[39] K. Xu, B. Meng, and Z. Wang, "Generalized regression neural networks-based data-driven iterative learning control for nonlinear non-affine discrete-time systems," *Expert Systems with Applications*, vol. 248, p. 123339, 2024.

[40] C. Yang, C. Chen, W. He, R. Cui, and Z. Li, "Robot learning system based on adaptive neural control and dynamic movement primitives," *IEEE Trans. Neural Netw. Learn. Syst*., vol. 30, no. 3, pp. 777–787, Mar. 2019.

[41] Y. Lv, X. Ren, J. Tian, and X. Zhao, "Inverse-model-based iterative learning control for unknown MIMO nonlinear system with neural network," *Neurocomputing*, vol. 519, pp. 187-193, 2023.

[42] M. Sun, T. Wu, L. Chen, and G. Zhang, "Neural AILC for error tracking against arbitrary initial shifts," *IEEE Trans. Neural Netw. Learn. Syst*., vol. 29, no. 7, pp. 2705–2716, Jul. 2018.

[43] Y. Chai, K. Liu, X. Duan, J. Yi, R. Sun, and J. Li, "Human–machine interaction controller of upper limb based on iterative learning method with zeroing neural algorithm and disturbance observe," *Engineering Applications of Artificial Intelligence*, vol. 122, p. 106108, 2023.

[44] Q. Yan, J. Cai, Y. Zhang, and Z. Yang, "Adaptive Iterative Learning Control for Robot Manipulators With Time–Varying Parameters and Arbitrary Initial Errors," in *IEEE Access*, vol. 9, pp. 166920–166931, 2021.

[45] X. Jin, "Iterative learning control for non–repetitive trajectory tracking of robot manipulators with joint position constraints and actuator faults," *Int. J. Adapt. Control Signal Process*., vol. 31, no. 6, pp. 859–875, Jun. 2017.

[46] Z. Chaochen, Y. Wang, H. Tan, J. Zhou, and Q. Lin, "Posture–based Leader–follower Formation for Multiple Mobile Robot Systems Using Data–based Adaptive Iterative Learning Control," *2022 China Automation Congress (CAC)*, pp. 6463–6468, 2022.

[47] J. Xu, D. Li, X. Ma, B. Zhang, and Q. Jin, "Study of an Adaptive Iterative Learning Control Based on Extended State Observer in Path Tracking of Double Joint Robot," *2020 39th Chinese Control Conference (CCC)*, pp. 2037–2042, 2020.

[48] Z. Cheng and Z. Zhuo, "Adaptive Iterative Learning Trajectory Tracking Control of SCARA Robot," *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 910–914, 2021.

[49] J. Li, R. Chi, and D. Yang, "High–order Model–free Adaptive Iterative Learning Control for Velocity Tracking of Wheeled Mobile Robots," *2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 388–392, 2020.

[50] L. Sun and W. Duan, "Decentralized Robust Adaptive Iterative Learning Control of Robot Manipulators," *2018 37th Chinese Control Conference (CCC)*, pp. 2685–2689, 2018.

[51] H. Zhang, Q. Yan, J. Cai, S. Gao, and Y. Zhang, "Initial–Rectification Neuro–Adaptive Iterative Learning Control for Robot Manipulators With Input Deadzone and Nonzero Initial Errors," in *IEEE Access*, vol. 11, pp. 22441–22449, 2023.

[52] Z. Zhu, X. Xu, Y. Zhu, A. Rodic, and P. B. Petrovic, "Research on Fuzzy Adaptive and PD–Type Iterative Learning Control for Robot Manipulator," *2021 6th IEEE International Conference on Advanced Robotics and Mechatronics (ICARM)*, pp. 487–491, 2021.

[53] D. X. Ba, N. T. Thien, and J. Bae, "A Novel Iterative Second–Order Neural–Network Learning Control Approach for Robotic Manipulators," in *IEEE Access*, vol. 11, pp. 58318–58332, 2023.

[54] D. X. Ba and T. T. Nguyen, "A new neural iterative learning control approach for position tracking control of robotic manipulators: Theory simulation and experiment," *Meas. Control Autom*., vol. 3, no. 2, pp. 66–74, 2022.

[55] C. Zhang and X. Tian, "Non–uniform trajectory tracking adaptive iterative learning control for nonlinear pure–feedback systems with initial state error based on RBF–neural network," *2021 40th Chinese Control Conference (CCC)*, pp. 532–539, 2021.

[56] G. Zhu and Q. Yan, "Error Tracking–Based Neuro–Adaptive Learning Control for Pneumatic Artificial Muscle Systems With Output Constraint," in *IEEE Access*, vol. 11, pp. 127479–127491, 2023.

[57] C. Zhang and J. Li, "Adaptive iterative learning control for nonlinear pure–feedback systems with initial state error based on fuzzy approximation," *Journal of the Franklin Institute*, vol. 351, no. 3, pp. 1483–1500, 2014.

[58] P. R. Ouyang, W. J. Zhang, and M. M. Gupta, "An adaptive switching learning control method for trajectory tracking of robot manipulators," *Mechatronics*, vol. 16, pp. 51–61, Apr. 2006.

[59] Q. Chen, H. Shi, and M. Sun, "Echo State Network-Based Backstepping Adaptive Iterative Learning Control for Strict-Feedback Systems: An Error-Tracking Approach," in *IEEE Transactions on Cybernetics*, vol. 50, no. 7, pp. 3009-3022, July 2020.

[60] Y. Guo, J. Ding, X. Feng, P. Sun, Q. Fang, and M. Wei, "Robust adaptive iterative learning control for high-speed trains under non-strictly repeated conditions," *Control Engineering Practice*, vol. 14, p. 105865, 2024.