Artificial Chemical Reaction Optimization Algorithm and Neural Network Based Adaptive Control for Robot Manipulator

Thuy Van Tran*, YaoNan Wang**

**College of Electrical and Information Engineering, Hunan University, Changsha, Hunan, 410082, China; and Faculty of Electrical Engineering, Hanoi University of Industry, Hanoi, 10000, Vietnam (e-mail: tranthuyvan.haui@gmail.com).*
***College of Electrical and Information Engineering, Hunan University, Changsha, Hunan, 410082, China (e-mail: yaonan@hnu.edu.cn).*

Abstract: This paper presents an artificial chemical reaction optimization algorithm (ACROA) and neural network based adaptive control scheme for robot manipulator (RM) to attain the desired trajectory tracking. Radial basis function neural network (RBFNN) is applied to approximate the uncertainties in robot dynamics. The network parameters in initial stage are optimized by utilizing ACROA. The RBFLN weights are determined based on adaptive tuning strategy in Lyapunov theory. Thus, the convergence and stability of whole system are guaranteed, and the tracking performance of RM is improved. The simulation results of two-link RM are represented to validate the efficiency of the proposed method.

Keywords: Artificial chemical reaction optimization algorithm, radial basis function neural network, network parameter optimization, adaptive control, robot manipulator.

1. INTRODUCTION

In past decades, neural network control (NNC) method (Omidvar and Elliott, 1997) has been widely applied to control the motion of robot manipulators (Ge et al., 1997; Cheng et al., 2009; Wang et al., 2015). Generally, the control performance of robot manipulators are influenced by external disturbances and various uncertainties such as friction forces, unmodeled dynamics, and parameter variations (Spong et al., 2005). It is hard to specifically attain the mathematical models in the design process for the control systems. Hence, the usual requirement of the NNC method is that it can decrease the influence of the external disturbances and the dynamic uncertainties on the system performance by utilizing its learning capability without detailed knowledge of robot manipulator. Especially, by using Gaussian function to replace sigmoid function in multilayer perceptron neural network, radial basis function neural network (RBFNN) has more compact topology, faster learning ability (Park and Sandberg, 1991; Billings et al., 2007), and more extensive application (Tang et al., 2013; Mahjoub et al., 2014; Han and Lee, 2015). Nevertheless, the RBFNN only have expected performance after learning for some time (Mu, 2010). In order to achieve expected performance in initial stage, the parameters of RBFNN need be optimized in advance.

Recently, a novel metaheuristic optimization method was suggested by Alatas, namely artificial chemical reaction optimization algorithm (ACROA) (Alatas, 2011). The ACROA is developed based on the chemical reactions of molecules and the second law of thermodynamics, so a system tends to the lowest enthalpy and the highest entropy (Moran et al., 2014). In the ACROA, enthalpy or entropy can be used as objective function for minimization problem or maximization problem. ACROA is different from other optimization algorithms such as genetic algorithm (GA) (Whitley, 1994) and particle swarm optimization (PSO) (Poli et al., 2007) in a solution technique of optimization and search problem. In addition, the ACROA method has fewer parameters and is more robust than that of used in other optimization methods. Thus, ACROA is adapted to solve the optimization problems. The successful application of the ACROA for the mining of classification rules can be indicated in (Alatas, 2012).

In this paper, an ACROA and RBFNN based adaptive control (ARNAC) system for robot manipulator is proposed to obtain the trajectory tracking performance with high accuracy. In the ARNAC scheme, firstly, the RBFNN with powerful learning capability is applied to approximate the robot dynamics function including the uncertainties. Secondly, a robust term is added into tracking control law to eliminate the external disturbances and functional approximation errors. Thirdly, by utilizing the ACROA to optimize the parameters of RBFLN in initial stage, the convergence rate and the tracking performance of the robot system are improved. Fourthly, an adaptive tuning strategy for the weights of RBFLN is derived from Lyapunov stability theory to guarantee the network convergence and the stable control performance. Finally, the numerical simulation results of two-link robot manipulator are represented to validate the robustness and the superiority of the proposed adaptive control method in comparison with other existing methods.

The remainder of the paper is organized as follows. The preliminaries are described in section 2. The design procedure of the proposed control system is provided in section 3. Section 4 represents the numerical simulation results. Finally, section 5 shows the conclusions.

2. PRELIMINARIES

2.1 Dynamics Model of Robot Manipulator

Generally, the dynamics of an n-link robot manipulator can be presented under the Lagrange equation (Lewis et al., 2003):

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{F}(\dot{\boldsymbol{q}}) + \boldsymbol{u} = \boldsymbol{\tau} \quad (1)$$

where inertial matrix, Coriolis-centripetal matrix, gravity vector, and friction forces vector are denoted by $\boldsymbol{M}(\boldsymbol{q}) \in R^{n\times n}$, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in R^{n\times n}$, $\boldsymbol{G}(\boldsymbol{q}) \in R^n$, and $\boldsymbol{F}(\dot{\boldsymbol{q}}) \in R^n$, respectively. $\boldsymbol{q}(t)$, $\dot{\boldsymbol{q}}(t)$, and $\ddot{\boldsymbol{q}}(t) \in R^n$ are joint position vector, corresponding velocity vector, and corresponding acceleration vector, respectively. External disturbances are expressed by the vector $\boldsymbol{u} \in R^n$, and $\boldsymbol{\tau} \in R^n$ is a control vector of joint torques.

In the robot dynamics, the following fundamental properties hold (Lewis et al., 2003):

Property 1: $\boldsymbol{M}(\boldsymbol{q})$ is a symmetric positive definite matrix and satisfies

$$\rho_1 \boldsymbol{I}_n \le \boldsymbol{M}(\boldsymbol{q}) \le \rho_2 \boldsymbol{I}_n; \quad \forall \boldsymbol{q} \in R^n \quad (2)$$

where $\boldsymbol{I}_n \in R^{n\times n}$ is the identity matrix, and ρ_1 and ρ_2 are positive constants.

Property 2: The matrix $[\dot{\boldsymbol{M}}(\boldsymbol{q}) - \boldsymbol{2C}(\boldsymbol{q},\dot{\boldsymbol{q}})]$ is skew-symmetric and satisfies

$$\boldsymbol{\mu}^{\boldsymbol{T}}\left[\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\right]\boldsymbol{\mu} = 0; \quad \forall \boldsymbol{\mu} \in R^n \quad (3)$$

Property 3: The following inequalities are satisfied: $\|\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\| \le k_c\|\dot{\boldsymbol{q}}\|$; $\|\boldsymbol{G}(\boldsymbol{q})\| \le k_g$; $\|\boldsymbol{u}\| \le k_u$; where $\|\,.\,\|$ is Euclidean norm, and k_c, k_g, and k_u are positive constants.

2.2 Structure of RBFNN

As illustrated in Figure 1, the structure of RBFNN includes p input variables, r hidden neurons, and n output variables. For each layer in the RBFNN, input-output variables and radial basis functions are detailed as follows:

Layer 1 (Input Layer): Consider a vector of input variables as $\boldsymbol{\vartheta} = [\vartheta_1, \vartheta_2, \dots, \vartheta_p]^T \in R^p$, the input variables will be directly transmitted to the hidden layer by the input neurons.

Layer 2 (Hidden Layer): Each neuron in this layer is represented by a Gaussian radial basis function (GRBF) as

$$\varphi_j(\boldsymbol{\vartheta}) = e^{-\frac{\|\vartheta - z_j\|^2}{2b_j^2}}; \quad j = \{1, 2, \dots, r\} \quad (4)$$

where $b_j \in R$ and $\boldsymbol{z}_j = [z_{j1}, z_{j2}, \dots, z_{jp}]^T \in R^p$ are the variance and the centre vector of the j^{th} GRBF, respectively. Then, the variance vector is denoted by $\boldsymbol{b} = [b_1, b_2, \dots, b_r]^T \in R^r$, and the centre matrix is expressed by $\boldsymbol{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_r]^T \in R^{r\times p}$.

Layer 3 (Output Layer): Each output variable is an output neuron, and is calculated by the weighted sum technique as follows:

$$y_i = \sum_{j=1}^{r} w_{ij}\varphi_j; \quad i = \{1, 2, \dots, n\} \quad (5)$$

where w_{ij} denotes a weight between the j^{th} hidden neuron and the i^{th} output neuron. Then, the vector form of output variables can be described as

$$\boldsymbol{y} = \boldsymbol{W}^T\boldsymbol{\varphi} = [y_1, y_2, \dots, y_n]^T \quad (6)$$

where $\boldsymbol{\varphi} = [\varphi_1, \varphi_2, \dots, \varphi_r]^T \in R^r$ is the output vector of hidden layer, and the weight matrix between the hidden layer and the output layer is expressed by

$$\boldsymbol{W} = \begin{bmatrix} w_{11} & w_{21} & \cdots & w_{n1} \\ w_{12} & w_{22} & \cdots & w_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1r} & w_{2r} & \cdots & w_{nr} \end{bmatrix} \in R^{r\times n} \quad (7)$$

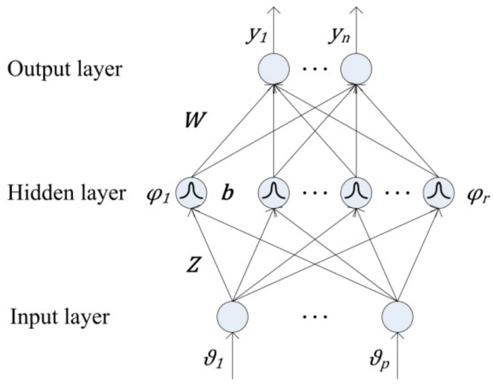


Fig. 1. Structure of RBFNN.

2.3 ACROA Method

ACROA is a stochastic and adaptive search method. Its optimization is based on a chemical reaction process that leads to the transformation of chemical substances into another. The principle of ACROA contains five steps (more details about five steps can be found in (Alatas, 2011; Alatas, 2012).

Step 1: Optimization problem and initial parameters

The optimization problem is defined as

$$\text{minimize}\{f(\boldsymbol{\alpha})\}; \ \alpha_k \in H_k = \left[\theta_k^l, \theta_k^u\right]; \ k = \{1, 2, \dots, N\} \quad (8)$$

where $f(\boldsymbol{\alpha})$ is a objective function, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_N]$ is a decision variables vector, H_k is the feasible range of values for the k^{th} decision variable, N is the number of decision variables, and θ_k^u and θ_k^l are the upper and lower bounds of the k^{th} decision variable, respectively. The different encoding type of molecules is appropriately used for each optimization problem. Furthermore, the parameter $ReacNum$, is the initial population of reactants, is assigned in this step.

Step 2: Initialization and evaluation for reactants

The reactants are uniformly initialized in the possible solution region. The association rules are represented, and the value of objective function is evaluated.

Step 3: Application of elementary reactions

In the ACROA, there are five elementary reactions, namely decomposition reaction, redox1 reaction, synthesis reaction, displacement reaction, and redox2 reaction.

Step 4: Updating reactants

The chemical equilibrium is tested, and the new reactants are updated by evaluating the objective function values.

Step 5: Checking termination criterion

Step 3 and step 4 will be repeated until the termination criterion is met.

3. DESIGN OF ARNAC

3.1 Defined Control Law

Given a desired trajectory $\boldsymbol{q}_d(t) \in R^n$, with $\boldsymbol{q}_d(t)$, $\dot{\boldsymbol{q}}_d(t)$, and $\ddot{\boldsymbol{q}}_d(t)$ are uniformly continuous and bounded, the tracking error between $\boldsymbol{q}_d$ and $\boldsymbol{q}$ is

$$\boldsymbol{\varepsilon}(t) = \boldsymbol{q}_d(t) - \boldsymbol{q}(t) \tag{9}$$

Then the filtered tracking error is represented as follows:

$$\boldsymbol{\delta}(t) = \dot{\boldsymbol{\varepsilon}} + \boldsymbol{\Lambda\varepsilon} \tag{10}$$

where $\boldsymbol{\Lambda}$ expresses a diagonal and positive definite constant matrix.

By differentiating (10) and using (9), the dynamics of (1) can be obtained as

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\delta}} = -\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{\delta} - \boldsymbol{\tau} + \boldsymbol{y}(\boldsymbol{\vartheta}) + \boldsymbol{u} \tag{11}$$

where $\boldsymbol{y}(\boldsymbol{\vartheta})$ is a nonlinear dynamics function, and it is presented as

$$\boldsymbol{y}(\boldsymbol{\vartheta}) = (\ddot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\dot{\boldsymbol{\varepsilon}})\boldsymbol{M} + (\dot{\boldsymbol{q}}_d + \boldsymbol{\Lambda\varepsilon})\boldsymbol{C} + \boldsymbol{G} + \boldsymbol{F} \tag{12}$$

The vector $\boldsymbol{\vartheta}$, is entailed to determine $\boldsymbol{y}(\boldsymbol{\vartheta})$, is specified as

$$\boldsymbol{\vartheta} = [\, \boldsymbol{\varepsilon}^T,\ \dot{\boldsymbol{\varepsilon}}^T,\ \boldsymbol{q}_d^T,\ \dot{\boldsymbol{q}}_d^T,\ \ddot{\boldsymbol{q}}_d^T \,]^T \tag{13}$$

Consider a control input torque which is defined as follows:

$$\boldsymbol{\tau}_d = \hat{\boldsymbol{y}}(\boldsymbol{\vartheta}) + \boldsymbol{K\delta} \tag{14}$$

where gain matrix $\boldsymbol{K} = \boldsymbol{K}^T > 0$, and $\hat{\boldsymbol{y}}(\boldsymbol{\vartheta})$ is the estimate of the nonlinear function $\boldsymbol{y}(\boldsymbol{\vartheta})$. Then functional estimate error is determined by

$$\tilde{\boldsymbol{y}}(\boldsymbol{\vartheta}) = \boldsymbol{y}(\boldsymbol{\vartheta}) - \hat{\boldsymbol{y}}(\boldsymbol{\vartheta}) \tag{15}$$

Hence, the dynamics of the closed-loop system become

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\delta}} = -[\boldsymbol{K} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})]\boldsymbol{\delta} + \boldsymbol{\beta}_d \tag{16}$$

where $\boldsymbol{\beta}_d$ is defined by $\boldsymbol{\beta}_d = \tilde{\boldsymbol{y}}(\boldsymbol{\vartheta}) + \boldsymbol{u}$. This is a system of the errors, wherein $\dot{\boldsymbol{\varepsilon}}$ and $\boldsymbol{\varepsilon}$ are driven by $\tilde{\boldsymbol{y}}(\boldsymbol{\vartheta})$.

From (14), the proportional derivative (PD) control is incorporated in the control input torque $\boldsymbol{\tau}_d$, i.e., $\boldsymbol{K\delta} = \boldsymbol{K}(\boldsymbol{\Lambda\varepsilon} + \dot{\boldsymbol{\varepsilon}}) \equiv \boldsymbol{K}_P\boldsymbol{\varepsilon} + \boldsymbol{K}_D\dot{\boldsymbol{\varepsilon}}$, where $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ denote the diagonal and positive definite constant matrices. Moreover, in order to guarantee the stability for $\boldsymbol{\delta}(t)$, the dynamics as (16) is utilized to select an adaptive tuning strategy for the network parameters. Thus, the stability of a system which comprises the input as $\boldsymbol{\delta}(t)$ and the output as $\boldsymbol{\varepsilon}(t)$ in (10) can be ensured, and $\boldsymbol{\varepsilon}(t)$ demonstrates stable behaviour (Liu and Lewis, 1992). Essentially, $\|\dot{\boldsymbol{\varepsilon}}\|_2 \le \|\boldsymbol{\delta}\|_2$, and $\|\boldsymbol{\varepsilon}\|_2 \le \|\boldsymbol{\delta}\|_2/\lambda_{min}$, where $\|\,.\,\|_2$ demotes Euclidean norm, and λ_{min} is the minimum singular value of the matrix $\boldsymbol{\Lambda}$.

Property 4: A system with the dynamics as (16), which includes the input as $\boldsymbol{\beta}_d(t)$ and the output as $\boldsymbol{\delta}(t)$, is the state-strict passive system and validates the equality of a power form (Lewis et al., 2003).

3.2 RBFNN Based Adaptive Control (NNBC)

In this control scheme, an RBFNN is utilized to approximate the nonlinear function of the unmodeled dynamics. The input of the RBFNN is considered as $\boldsymbol{\vartheta}$ in (13), and the output of the RBFNN is considered as $\boldsymbol{y}(\boldsymbol{\vartheta})$ in (12). Then, there exists an ideal matrix $\boldsymbol{W}^*$ of RBFNN such that:

$$\boldsymbol{y}(\boldsymbol{\vartheta}) = \boldsymbol{W}^{*T}\boldsymbol{\varphi} + \boldsymbol{\sigma} \tag{17}$$

where $\boldsymbol{\sigma}$ expresses the functional approximate error of the RBFNN, and it is arbitrarily decreased if the neuron number in the hidden layer is increased (i.e., $\|\boldsymbol{\sigma}\| \le k_\sigma$, with k_σ is a known positive constant). Suppose that the norm of ideal matrix is bounded by a known positive real value ($\|\boldsymbol{W}^*\| \le W_{max}$), and $\boldsymbol{W}^*$ is determined by known value (Spooner et al., 2001):

$$\boldsymbol{W}^* = \arg\min_{\boldsymbol{W}\in S_W}\left[\sup_{\boldsymbol{\vartheta}\in S_\vartheta}\|\boldsymbol{y}(\boldsymbol{\vartheta}) - \boldsymbol{W}^T\boldsymbol{\varphi}\|\right] \tag{18}$$

where S_ϑ and S_W denote the compact sets of $\boldsymbol{\vartheta}$ and $\boldsymbol{W}$, respectively.

Actually, it is hard to exactly achieve the weight matrix $\boldsymbol{W}$ that the approximation for $\boldsymbol{y}(\boldsymbol{\vartheta})$ is the best. Hence, the output vector of RBFNN, as in (6), is applied to estimate $\boldsymbol{y}(\boldsymbol{\vartheta})$ in (17). Then, the RBFNN functional estimate, $\hat{\boldsymbol{y}}$, is presented as

$$\hat{\boldsymbol{y}} = \widehat{\boldsymbol{W}}^{\boldsymbol{T}}\boldsymbol{\varphi} \tag{19}$$

where $\widehat{\boldsymbol{W}}$ is the estimative matrix of $\boldsymbol{W}$, and it is provided by an adaptive tuning strategy. So the deviation of weight matrix, is denoted by $\widetilde{\boldsymbol{W}}$, is determined as $\widetilde{\boldsymbol{W}} = \boldsymbol{W}^* - \widehat{\boldsymbol{W}}$.

Based on the definition of $\boldsymbol{\tau}_d$ as in (14) and the presentation of $\hat{\boldsymbol{y}}$ as in (19), an tracking control law is designed as follows:

$$\boldsymbol{\tau} = \widehat{\boldsymbol{W}}^{\boldsymbol{T}}\boldsymbol{\varphi} + \boldsymbol{K\delta} + (k_\sigma + k_u)\mathrm{sgn}(\boldsymbol{\delta}) \tag{20}$$

where $\widehat{\boldsymbol{W}}^{\boldsymbol{T}}\boldsymbol{\varphi}$ is applied to approximate the unmodeled dynamics, $\boldsymbol{K\delta}$ is used to ensure the stability of control system, and $\boldsymbol{\omega} \equiv (k_\sigma + k_u)\text{sgn}(\boldsymbol{\delta})$ is robust term for eliminating the functional reconstruction errors and the external disturbances.

Thus, the dynamics of the closed-loop control system obtain

$$\boldsymbol{M}\dot{\boldsymbol{\delta}} = -(\boldsymbol{K} + \boldsymbol{C})\boldsymbol{\delta} + \widetilde{\boldsymbol{W}}^{\boldsymbol{T}}\boldsymbol{\varphi} + \boldsymbol{\sigma} + \boldsymbol{u} - (k_\sigma + k_u)\text{sgn}(\boldsymbol{\delta}) \quad (21)$$

Theorem 1: Consider an n-link robot manipulator system which comprises the dynamics as described in (1) and (21), if the tracking control law is designed as (20), and the weight matrix of RBFNN is updated by the adaptive tuning strategy as

$$\dot{\widehat{\boldsymbol{W}}} = \boldsymbol{X}\boldsymbol{\varphi}\boldsymbol{\delta}^{\boldsymbol{T}} \quad (22)$$

where $\boldsymbol{X}$ is a diagonal and positive definite constant matrix, then the control system is convergent and stable.

Proof: Consider a Lyapunov function as

$$\mathcal{L} \equiv \mathcal{L}\big(\boldsymbol{\delta}(t), \widetilde{\boldsymbol{W}}\big) = \tfrac{1}{2}\boldsymbol{\delta}^T\boldsymbol{M}\boldsymbol{\delta} + \tfrac{1}{2}tr(\widetilde{\boldsymbol{W}}^T\boldsymbol{X}^{-1}\widetilde{\boldsymbol{W}}) \quad (23)$$

where $tr(\,.\,)$ expresses the trace operator. The time derivative of Lyapunov function is then:

$$\dot{\mathcal{L}} = \boldsymbol{\delta}^T\boldsymbol{M}\dot{\boldsymbol{\delta}} + \tfrac{1}{2}\boldsymbol{\delta}^T\dot{\boldsymbol{M}}\boldsymbol{\delta} + tr\left(\widetilde{\boldsymbol{W}}^T\boldsymbol{X}^{-1}\dot{\widetilde{\boldsymbol{W}}}\right) \quad (24)$$

Based on the Properties, (21), and (22), it is concluded that

$$\dot{\mathcal{L}} = -\boldsymbol{\delta}^T\boldsymbol{K}\boldsymbol{\delta} + \tfrac{1}{2}\boldsymbol{\delta}^T\big(\dot{\boldsymbol{M}} - 2\boldsymbol{C}\big)\boldsymbol{\delta} + tr\widetilde{\boldsymbol{W}}^T\left(\boldsymbol{X}^{-1}\dot{\widetilde{\boldsymbol{W}}} + \boldsymbol{\varphi}\boldsymbol{\delta}^T\right) + \boldsymbol{\delta}^T(\boldsymbol{\sigma} + \boldsymbol{u}) - \boldsymbol{\delta}^T(k_\sigma + k_u)\text{sgn}(\boldsymbol{\delta}) = -\boldsymbol{\delta}^T\boldsymbol{K}\boldsymbol{\delta} + \boldsymbol{\delta}^T(\boldsymbol{\sigma} + \boldsymbol{u}) - \|\boldsymbol{\delta}\|(k_\sigma + k_u) \quad (25)$$

Thus, $\dot{\mathcal{L}}\big(\boldsymbol{\delta}(t), \widetilde{\boldsymbol{W}}\big) \le -\boldsymbol{\delta}^T\boldsymbol{K}\boldsymbol{\delta} \le 0$, and $\boldsymbol{\mathcal{L}}\big(\boldsymbol{\delta}(t), \widetilde{\boldsymbol{W}}\big) \le \boldsymbol{\mathcal{L}}\big(\boldsymbol{\delta}(0), \widetilde{\boldsymbol{W}}\big)$. It means that $\boldsymbol{\delta}(t)$ and $\widetilde{\boldsymbol{W}}$ are uniformly bounded. Let a function as $\gamma(t) = \boldsymbol{\delta}^T\boldsymbol{K}\boldsymbol{\delta} \le -\dot{\mathcal{L}}$. By integrating $\gamma(t)$ with respect to time, it can be attained

$$\int_0^t \gamma(t)dt \le \boldsymbol{\mathcal{L}}\big(\boldsymbol{\delta}(0), \widetilde{\boldsymbol{W}}\big) - \boldsymbol{\mathcal{L}}\big(\boldsymbol{\delta}(t), \widetilde{\boldsymbol{W}}\big) \quad (26)$$

Since $\boldsymbol{\mathcal{L}}\big(\boldsymbol{\delta}(0), \widetilde{\boldsymbol{W}}\big)$ is bounded, and $\boldsymbol{\mathcal{L}}\big(\boldsymbol{\delta}(t), \widetilde{\boldsymbol{W}}\big)$ is bounded and non-increasing, so $\lim_{t\to\infty}\int_0^t \gamma(t)dt < \infty$, and $\dot{\gamma}(t)$ is bounded. According to Barbalat's Lemma in (Astrom and Wittenmark, 2008), it can be proved that $\lim_{t\to\infty}\gamma(t) = 0$. In other words, $\boldsymbol{\delta}(t) \to 0$ when $t \to \infty$. This completes the proof.

3.3 ACROA Based Optimized RBFNN Parameters

In order to improve the convergence rate and achieve the expected performance of RBFNN based control system in initial stage, the parameters of RBFNN need to be optimized in advance (Mu, 2010). The optimization of RBFNN parameters (i.e., $\boldsymbol{b}$, $\boldsymbol{Z}$, and $\boldsymbol{W}$) by ACROA, is described as in Figure 2, contains the following steps:

Step 1: The reactants are randomly initialized in the search region. Each molecule is encoded as a vector including all the elements of $\boldsymbol{b}$, $\boldsymbol{Z}$, and $\boldsymbol{W}$.

Step 2: Each reactant corresponds to an RBFNN structure in the model-based system. The objective function value of reactants is determined by $f = \|\boldsymbol{\varepsilon}\|^2/2$.

Step 3: The evaluation is implemented for the objective function values of reactants. A new reactant will be included when its objective function value is lowest, and a worse reactant will be excluded.

Step 4: The optimization process is repeated until the termination criterion is met. Then, the best reactant corresponds to the optimal parameters of RBFNN (i.e., $\boldsymbol{b}^0$, $\boldsymbol{Z}^0$, and $\boldsymbol{W}^0$).

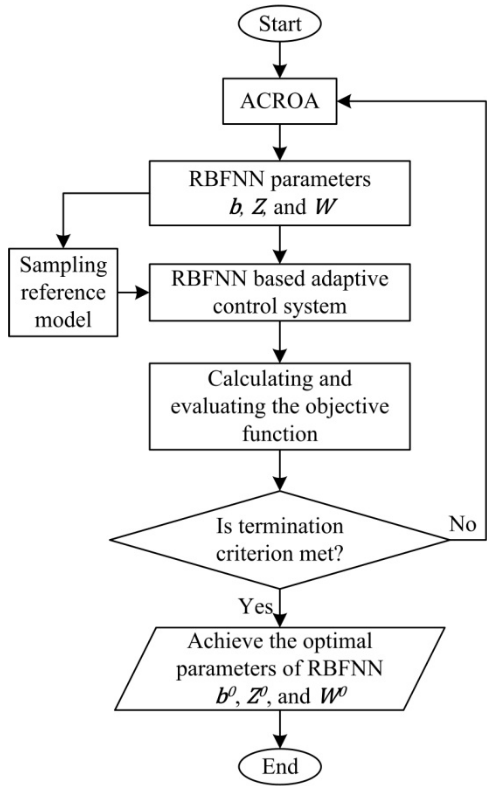


Fig. 2. Flow chart of RBFNN parameters optimization based on ACROA.

3.4 Proposed ARNAC system

Herein, an ARNAC system is proposed as in Figure 3 for tracking the desired trajectory of robot manipulator. The input of RBFNN, as in (13), includes the desired trajectories and the filtered tracking errors. The output of RBFNN, is the RBFNN functional estimate as in (19), is used to approximate the nonlinear dynamics function as in (12). The control input torque of robot manipulator is designed as (20).

According to Theorem 1, the closed-loop dynamics system as (21) is convergent and stable. Additionally, as mentioned above, the optimal RBFNN parameters (i.e., $\boldsymbol{b}^0$, $\boldsymbol{Z}^0$, and $\boldsymbol{W}^0$) will be used as initial parameters in the system instead of the random generation of them. Therefore, the convergence rate of system is improved, and the tracking performance can be obtained with high accuracy.

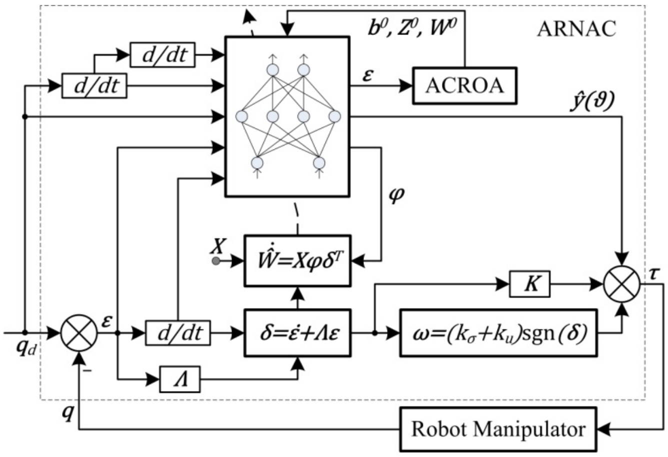


Fig. 3. Structure of ARNAC system.

4. NUMERICAL SIMULATION RESULTS

In this simulation, a two-link robot manipulator, as illustrated in Figure 4, is considered to validate the efficiency and robustness of the proposed ARNAC system. The dynamics parameters of two-link robot manipulator are detailed as follows:

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix};\ M_{22} = m_2 l_2^2;\ M_{11} = (m_1 + m_2) l_1^2 + m_2 l_2^2 + 2 m_2 l_1 l_2 \cos(q_2);\ M_{21} = M_{12} = m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2);\ \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix};\ C_{22} = 0;\ C_{11} = -m_2 l_1 l_2 \dot{q}_2 \sin(q_2);\ C_{21} = m_2 l_1 l_2 \dot{q}_1 \sin(q_2);\ C_{12} = -m_2 l_1 l_2 (\dot{q}_1 + \dot{q}_2) \sin(q_2);\ \boldsymbol{G}(\boldsymbol{q}) = \begin{bmatrix} G_1 \\ G_1 \end{bmatrix};\ G_1 = g(m_1 + m_2) l_1 \cos(q_2) + g m_2 l_2 \cos(q_1 + q_2);\ G_2 = g m_2 l_2 \cos(q_1 + q_2) \quad (27)$$

where $g = 9.81$ (m/s^2). The nominal parameters of the robot system are given as in Tables 1, and the desired trajectories are chosen by: $q_{d1}(t) = 0.3\cos(t) + 0.7\sin(2t)$; $q_{d2}(t) = 0.3\sin(t) + 0.7\cos(2t)$.

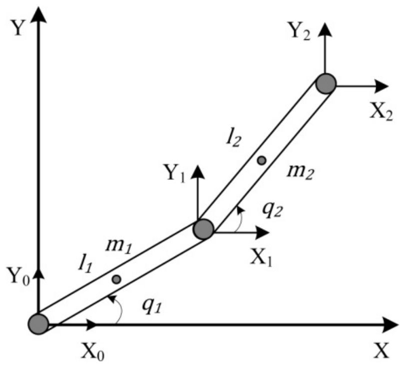


Fig. 4. Structure of two-link robot manipulator.

Table 1. The nominal parameters of the robot system.

Link	Mass m (kg)	Length l (m)	Initial position $\boldsymbol{q}(0)$ (rad)	Initial velocity $\dot{\boldsymbol{q}}(0)$ (rad/s)
Link 1	2	0.8	0.6	0
Link 2	1	1	0.4	0

In order to demonstrate the robustness and the superior control performance of the ARNAC system, not only the ARNAC system but also computed torque control (CTC) system, PD control system (Haddad and Chellaboina, 2008), and NNBC are simulated for comparison. On the other hand, the external disturbances and the various uncertainties which influence the control performance of system are also considered, where the conditions are the same for all control systems.

Firstly, the friction forces term, at the beginning, is formulated as follows (Shang et al., 2009):

$$\boldsymbol{F}(\dot{\boldsymbol{q}}) = [f_{v1}\dot{q}_1 + f_{c1}\text{sgn}(\dot{q}_1),\quad f_{v2}\dot{q}_2 + f_{c2}\text{sgn}(\dot{q}_2)]^T \quad (28)$$

where the viscous friction coefficients and the Coulomb friction forces of links 1 and 2, respectively, are given as $f_{v1} = 10$, $f_{c1} = 0.4$, $f_{v2} = 3$, $f_{c2} = 0.1$.

Secondly, the parameter variation, arising at 3 seconds, is that a weight m_w is added to the mass of second link:

$$m_2 = 1 + m_w = 1 + 0.5 = 1.5 \quad (29)$$

Thirdly, the external disturbances, occurring at 6 seconds, are considered as the external forces which are inserted into the robot system:

$$\boldsymbol{u} = [2\cos(10t),\quad 10\sin(10t)]^T \quad (30)$$

In addition, root mean square error (RMSE), is one of the most common measures, is utilized to exhibit the dominance of the ARNAC system distinctly.

$$RMSE_i = \sqrt{\frac{1}{T_s}\sum_{s=1}^{T_s}[q_{di}(s) - q_i(s)]^2};\quad i = \{1, 2\} \quad (31)$$

where $q_{di}(s)$ is the s^{th} element in q_{di}, $q_i(s)$ is the s^{th} element in q_i, and T_s expresses the total sampling instants.

In this paper, the simulation is carried out using the Windows Matlab.

4.1 Computed Torque Control for Robot Manipulator

Figure 5 describes the computed torque control (CTC) system, with the control input torque $\boldsymbol{\tau}_{ct}$ as

$$\boldsymbol{\tau}_{ct} = (\boldsymbol{K}_c\boldsymbol{\varepsilon} + \boldsymbol{K}_t\dot{\boldsymbol{\varepsilon}} + \ddot{\boldsymbol{q}}_d)\boldsymbol{M} + \dot{\boldsymbol{q}}\boldsymbol{C} + \boldsymbol{G} \quad (32)$$

where $\boldsymbol{K}_c$ and $\boldsymbol{K}_t$ are gain matrices, and they are given by:

$$\boldsymbol{K}_c = \begin{bmatrix} 60 & 0 \\ 0 & 80 \end{bmatrix};\ \boldsymbol{K}_t = \begin{bmatrix} 15 & 0 \\ 0 & 35 \end{bmatrix} \quad (33)$$

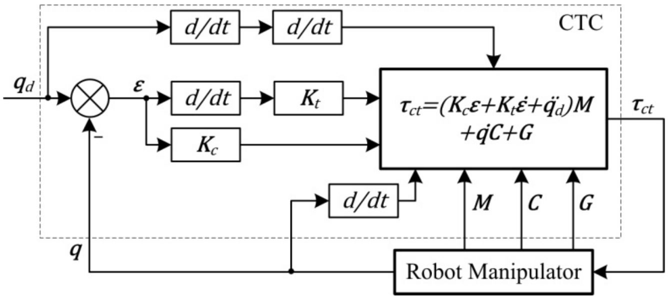
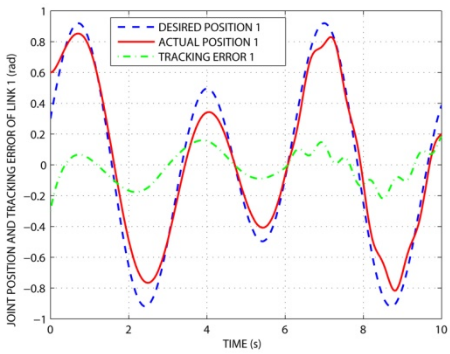
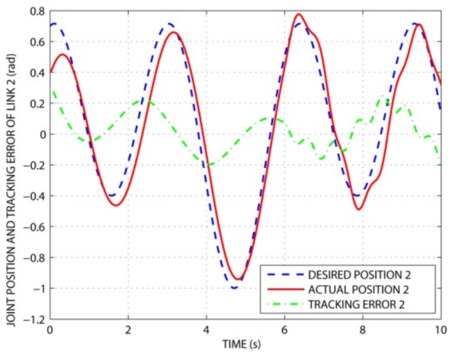


Fig. 5. Structure of CTC System.

The determination of $\boldsymbol{K}_c$ and $\boldsymbol{K}_t$ is based on the roots of $(\boldsymbol{K}_c\boldsymbol{\varepsilon} + \boldsymbol{K}_t\dot{\boldsymbol{\varepsilon}} = 0)$ strictly lie in the complex-plane left-haft, i.e., $\lim_{t\to\infty} \boldsymbol{\varepsilon}(t) = 0$. It implies that the CTC system is asymptotically and globally stable when the robot dynamics are considered without the uncertainties and the external disturbances. Nevertheless, when the system is disturbed by the uncertainties and the external disturbances, the stability of control system can be demolished. The simulation responses containing joint position, tracking error, RMSE, and control input torque for the CTC system are shown in Figures 6-7.



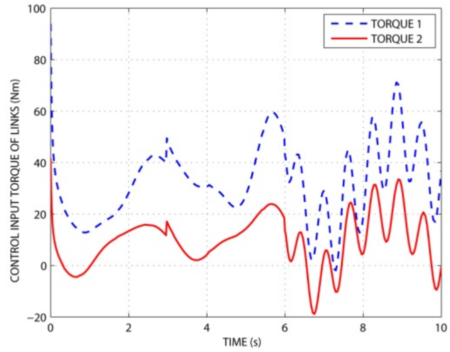
(a)



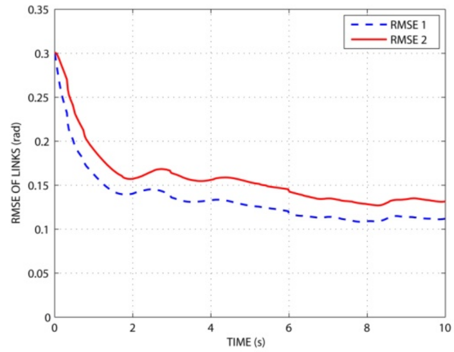
(b)

Fig. 6. Joint position and tracking error in the CTC control system: (a) Joint position and tracking error of link 1, (b) Joint position and tracking error of link 2.

From Figures 6-7, the beneficial tracking responses are only achieved in the nominal condition. Unfavourable tracking responses appear when the friction forces, the parameter variation, and the external disturbances occur. Actually, because the occurrence of the uncertainties and the external disturbances in the system, it is difficult to determine $\boldsymbol{K}_c$ and $\boldsymbol{K}_t$ reasonably.



(a)



(b)

Fig. 7. Control input torque and RMSE in the CTC control system: (a) Control input torque of links, (b) RMSE of links.

4.2 PD Control for Robot Manipulator

Figure 8 illustrates a PD control system, with the control input torque $\boldsymbol{\tau}_{pd}$ is defined as

$$\boldsymbol{\tau}_{pd} = \boldsymbol{K}_p\boldsymbol{\varepsilon} + \boldsymbol{K}_d\dot{\boldsymbol{\varepsilon}} \tag{34}$$

where $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ are gain matrices, and are given as

$$\boldsymbol{K}_p = \begin{bmatrix} 1400 & 0 \\ 0 & 800 \end{bmatrix};\ \boldsymbol{K}_d = \begin{bmatrix} 35 & 0 \\ 0 & 20 \end{bmatrix} \tag{35}$$

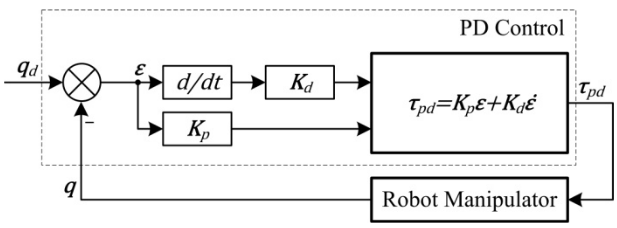


Fig. 8. Structure of PD System.

The determination of $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ is derived from the step responses of the controlled plant according to the tuning rules of Ziegler Nichols (Astrom and Wittenmark, 2008). Generally, $\boldsymbol{K}_p$ is chosen based on the satisfaction of steady state. For selecting $\boldsymbol{K}_d$, when its magnitude is increased, the noises of high frequency are amplified in the system. The simulation responses for the PD system are displayed in Figures 9-10.

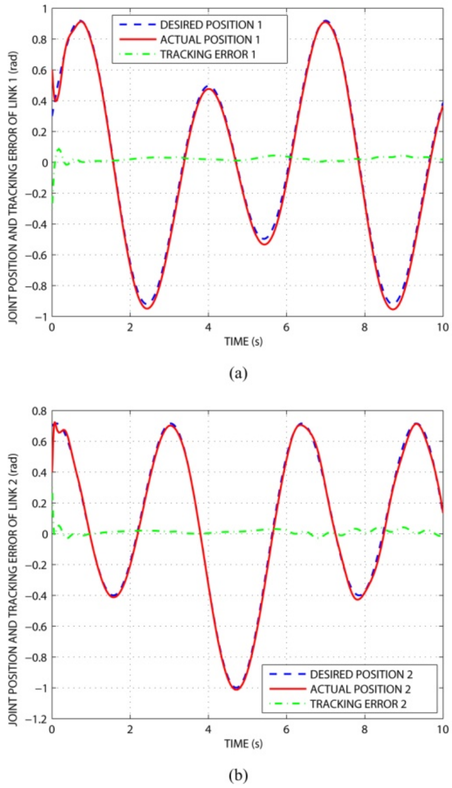


Fig. 9. Joint position and tracking error in the PD control system: (a) Joint position and tracking error of link 1, (b) Joint position and tracking error of link 2.

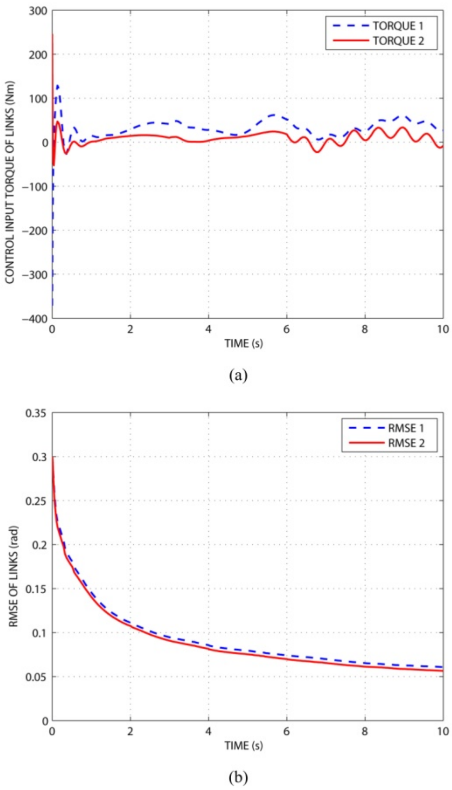


Fig. 10. Control input torque and RMSE in the PD control system: (a) Control input torque of links, (b) RMSE of links.

According to the simulation results in Figures 9-10, the system is stable, and the tracking responses are enhanced. However, the convergences of the control performance are still sluggish.

4.3 NNBC for Robot Manipulator

Herein, the NNBC system can be considered as the ARNAC system without ACROA. In this NNBC system, the values of the parameters are fixed as follows:

$$\boldsymbol{\Lambda} = \begin{bmatrix} 10 & 0 \\ 0 & 5 \end{bmatrix};\ \boldsymbol{K} = \begin{bmatrix} 55 & 0 \\ 0 & 40 \end{bmatrix};\ \boldsymbol{X} = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix};\ r = 10;$$
$$k_u = k_\sigma = 0.5 \tag{36}$$

The elements of $\boldsymbol{b}$, $\boldsymbol{Z}$, and $\boldsymbol{W}$ are randomly selected in ranges of $[-10,\ \ 10]$, $[-1,\ \ 1]$, and $[-5,\ \ 5]$, respectively. The simulation results comprising joint position, tracking error, RMSE, and control input torque for ARNAC system are depicted in Figures 11-12.

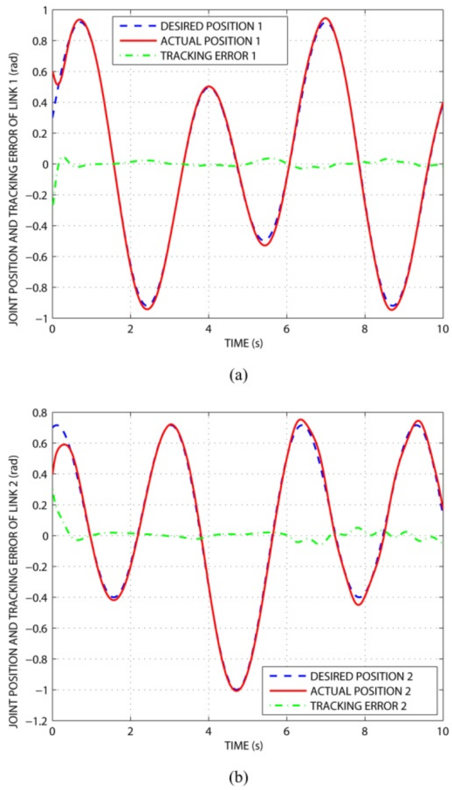


Fig. 11. Joint position and tracking error in the NNBC system: (a) Joint position and tracking error of link 1, (b) Joint position and tracking error of link 2.

Figures 11-12 show that the response of joint position is controlled to track the desired trajectory. Also, under the presence of the uncertainties and the external disturbances, position tracking error is regularly decreased due to learning capability of RBFNN. Nonetheless, the RBFNN parameters are selected roughly, there inescapably exists non-optimal parameters. Therefore, the convergence rate of system and the tracking errors can get better when the network parameters are optimized.

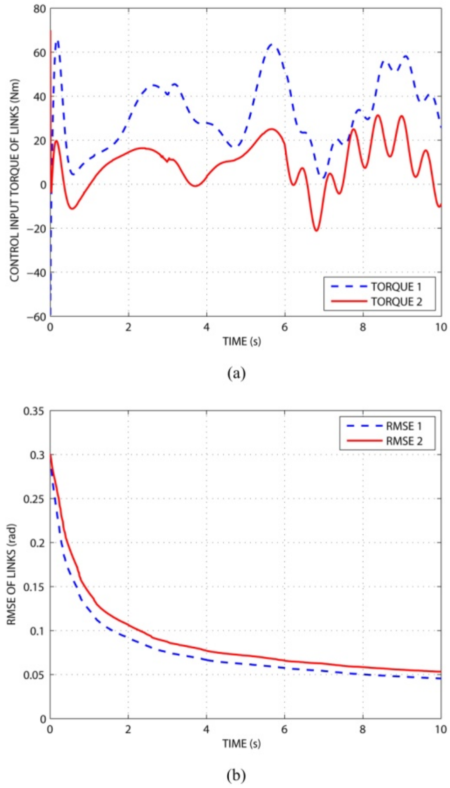


Fig. 12. Control input torque and RMSE in the NNBC system: (a) Control input torque of links, (b) RMSE of links.

4.4 ARNAC for Robot Manipulator

As described in Figure 3, the ARNAC is utilized to compare with CTC, PD control, and NNBC. Similarly, the detailed parameters of ARNAC system are determined as in (36). In the implementation of ACROA, the initial population is set by $ReacNum = 50$, and the termination criterion is considered as 100 iterations. The initial reactants, include all elements of $\boldsymbol{b}$, $\boldsymbol{Z}$, and $\boldsymbol{W}$, are generated over the search regions of $[-10,\ 10]$, $[-1,\ 1]$, and $[-5,\ 5]$, respectively. The simulation results for the ARNAC system are represented in Figures 13-14. Furthermore, the approximate error of the function $\boldsymbol{y}(\boldsymbol{\vartheta})$ for the NNBC system and the ARNAC system are illustrated in Figure 15, and the values of RMSE of four above control systems are also displayed in Table 2 for comparison.

In Figures 13-14, the joint positions can be controlled to closely track the desired trajectory with high accuracy, and the tracking error is improved considerably. Although in the occurrence of the external disturbances and the uncertainties, the convergence and the robustness of the ARNAC system are ensured. In comparison with the results in Figures 6-7 and Figures 9-12, the results in Figures 13-14 indicate that the ARNAC system achieves better-quality tracking performance without the chattering. Additionally, as observed from Figure 15, the ARNAC system can reach lower approximate error than the NNBC system.

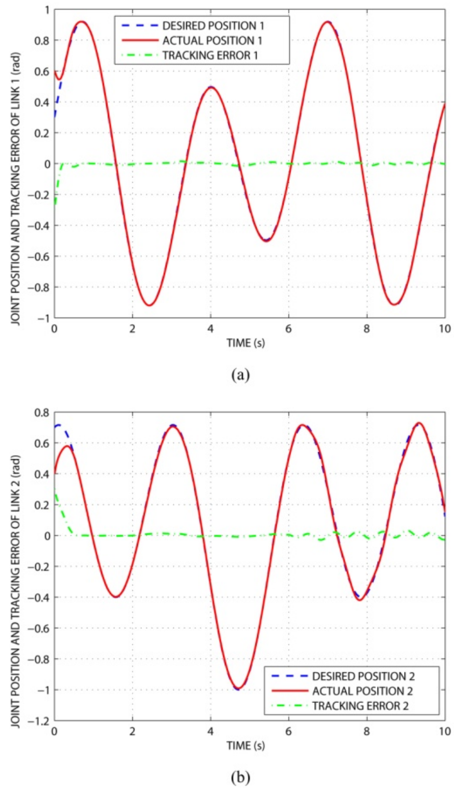


Fig. 13. Joint position and tracking error in the ARNAC system: (a) Joint position and tracking error of link 1, (b) Joint position and tracking error of link 2.

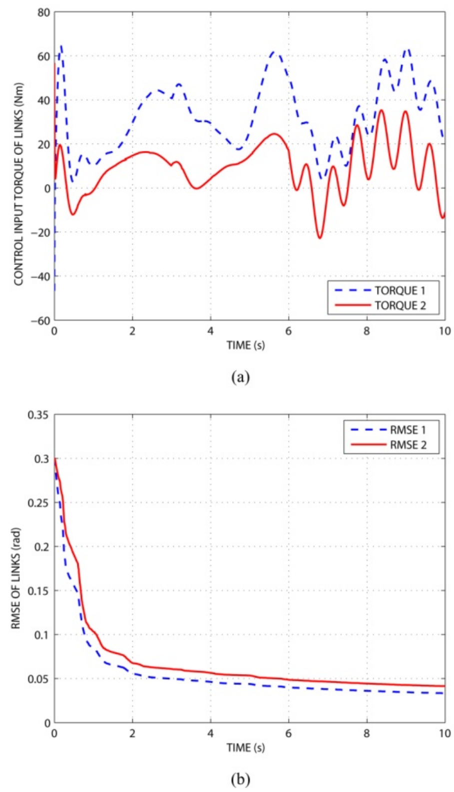


Fig. 14. Control input torque and RMSE in the ARNAC system: (a) Control input torque of links, (b) RMSE of links.

Table 2. RMSE of four control systems.

RMSE	CTC	PD Control	NNBC	ARNAC
$RMSE_1$ (rad)	0.112	0.061	0.046	0.033
$RMSE_2$ (rad)	0.132	0.057	0.053	0.041

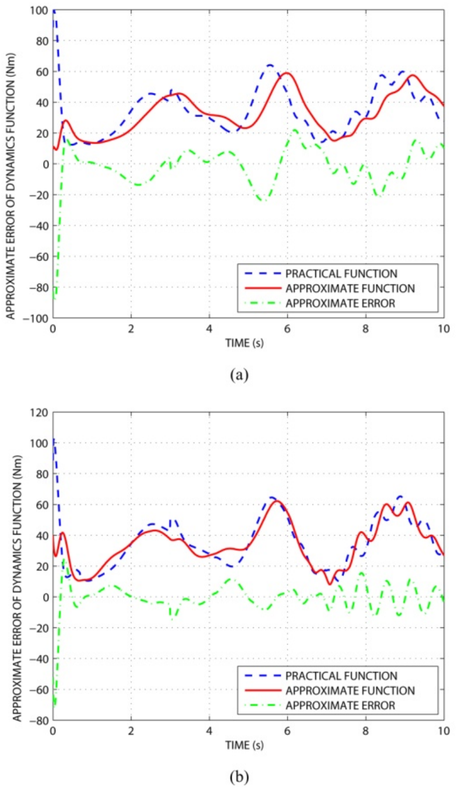


Fig. 15. Approximate error of function $\boldsymbol{y}$: (a) Approximate error of $\boldsymbol{y}$ for the NNBC system, (b) Approximate error of $\boldsymbol{y}$ for the ARNAC system.

The simulation results in Figures 6-7, Figures 9-15, and Table 2 demonstrate that the proposed ARNAC system attains smaller RMSE and tracking errors, while the convergence performance is better than the NNBC system, the PD control system, and the CTC system.

5. CONCLUSIONS

This paper has contributed a new method, namely ARNAC, to robotic control domain in the term of desired trajectory tracking. In the ARNAC scheme, both the tracking control law and the adaptive tuning strategy are derived from Lyapunov theory to guarantee the stability and robustness of whole system as well as the network convergence. Moreover, by utilizing the ACROA to achieve the optimal parameters of RBFLN in initial stage, the convergence rate and the tracking performance are improved significantly. Consequently, the proposed system attains the high-accuracy tracking performance without the detailed knowledge of the robot dynamics. Finally, the numerical simulation results of two-link robot manipulator exhibit that the joint positions can closely track the desired positions although in the appearance of the external disturbances and the various uncertainties. In comparison with other existing control methods containing CTC, PD Control, and NNBC, the ARNAC obtains lower tracking errors and faster convergence rates.

ACKNOWLEDGMENT

The authors are thankful to the editor and the reviewers for their invaluable comments and constructive suggestions with regard to the revision of the paper.

REFERENCES

Alatas, B. (2011). ACROA: Artificial Chemical Reaction Optimization Algorithm for global optimization. *Expert Systems with Applications,* 38 (10), 13170-13180.

Alatas, B. (2012). A novel chemistry based metaheuristic optimization method for mining of classification rules. *Expert Systems with Applications,* 39 (12), 11080-11088.

Astrom, K.J. and Wittenmark, B. (2008). *Adaptive Control: Second Edition*, 590. Dover Publications, Mineola, New York.

Billings, S.A., Wei, H.-L., and Balikhin, M.A. (2007). Generalized Multiscale Radial Basis Function Networks. *Neural Networks,* 20 (10), 1081-1094.

Cheng, L., Hou, Z.-G., and Tan, M. (2009). Adaptive Neural Network Tracking Control for Manipulators with Uncertain Kinematics, Dynamics and Actuator Model. *Automatica,* 45 (10), 2312-2318.

Ge, S.S., Hang, C.C., and Woon, L.C. (1997). Adaptive Neural Network Control of Robot Manipulators in Task Space. *IEEE Transactions on Industrial Electronics,* 44 (6), 746-752.

Haddad, W.M. and Chellaboina, V. (2008). *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*, 944. Princeton University Press, Princeton, New Jersey.

Han, S.-I. and Lee, J.-M. (2015). Decentralized neural network control for guaranteed tracking error constraint of a robot manipulator. *International Journal of Control, Automation and Systems,* 13 (4), 906-915.

Lewis, F.L., Dawson, D.M., and Abdallah, C.T. (2003). *Robot Manipulator Control: Theory and Practice*, 638. CRC Press, Boca Raton, Florida.

Liu, K. and Lewis, F.L. (1992). Robust Control Techniques for General Dynamic Systems. *Journal of Intelligent and Robotic Systems,* 6 (1), 33-49.

Mahjoub, S., Mnif, F., Derbel, N., and Hamerlain, M. (2014). Radial-Basis-Functions Neural Network Sliding Mode Control for Underactuated Mechanical Systems. *International Journal of Dynamics and Control,* 2 (1), 1-9.

Moran, M.J., Shapiro, H.N., Boettner, D.D., and Bailey, M.B. (2014). *Fundamentals of Engineering Thermodynamics*, 1056. John Wiley & Sons, Inc., Hoboken, New Jersey.

Mu, X. (2010). Fuzzy Neural Sliding Mode Control Based on Genetic Algorithm for Multi-Link Robots. *2010 Chinese Control and Decision Conference (CCDC)*, 1766-1770.

Omidvar, O. and Elliott, D.L. (1997). *Neural Systems for Control*, 358. Academic Press, Cambridge, Massachusetts.

Park, J. and Sandberg, I.W. (1991). Universal Approximation Using Radial-Basis-Function Networks. *Neural Computation,* 3 (2), 246-257.

Poli, R., Kennedy, J., and Blackwell, T. (2007). Particle swarm optimization. *Swarm Intelligence,* 1 (1), 33-57.

Shang, W.-w., Cong, S., and Jiang, S.-l. (2009). Dynamic model based nonlinear tracking control of a planar parallel manipulator. *Nonlinear Dynamics,* 60 (4), 597-606.

Spong, M.W., Hutchinson, S., and Vidyasagar, M. (2005). *Robot Modeling and Control*, 496. John Wiley & Sons, Inc., Hoboken, New Jersey.

Spooner, J.T., Maggiore, M., Ordonez, R., and Passino, K.M. (2001). *Stable Adaptive Control and Estimation for Nonlinear Systems: Neural and Fuzzy Approximator Techniques*, 550. John Wiley & Sons, Inc., Hoboken, New Jersey.

Tang, L., Liu, Y.-J., and Tong, S. (2013). Adaptive neural control using reinforcement learning for a class of robot manipulator. *Neural Computing and Applications,* 25 (1), 135-141.

Wang, C., Zhao, Y., Chen, Y., and Tomizuka, M. (2015). Nonparametric statistical learning control of robot manipulators for trajectory or contour tracking. *Robotics and Computer-Integrated Manufacturing,* 35, 96-103.

Whitley, D. (1994). A Genetic Algorithm Tutorial. *Statistics and Computing,* 4 (2), 65-85.