*Journal of **AI** and **D**ata **M**ining*
Vol 5, No 1, 2017, 137-147

Direct adaptive fuzzy control of flexible-joint robots including actuator dynamics using particle swarm optimization

M. Moradi Zirkohi* and S. Izadpanah

Department of Electrical Engineering, Behbahan Khatam Alanbia University of Technology, Behbahan, Iran.

Received 05 May 2015; Accepted 18 August 2016
**Corresponding author: moradi@bkatu.ac.ir (M. Moradi).*

Abstract

In this paper, a novel direct adaptive fuzzy system is proposed to control flexible-joint robots including actuator dynamics. This design includes two interior loops. The inner loop controls the motor position using the proposed approach, while the outer one controls the joint angle of robots using a proportional-integral-derivative (PID) control law. One novelty of this paper is the use of a particle swarm optimization (PSO) algorithm for optimizing the control design parameters in order to achieve the desired performance. It is worthy of note that to form the control law by considering practical considerations, just the available feedbacks are used. It is beneficial for industrial applications, where the real-time computation is costly. The proposed control approach has a fast response with a good tracking performance under the well-behaved control efforts. The stability is guaranteed in the presence of both the structured and unstructured uncertainties. As a result, all the system states remain bounded. The results of the simulation conducted on a two-link flexible-joint robot show the efficiency of the proposed scheme.

Keywords: *Fuzzy System, Particle Swarm Optimization, Flexible-Joint Robot, Actuator Dynamics.*

1. Introduction

Due to the non-linearities and coupling effects, the trajectory tracking control of robot manipulators with joint flexibilities is a challenging problem. Compared with the rigid robots, the number of degrees of freedom becomes twice the number of control actions due to flexibility in the joints, and the matching property between the non-linearities and inputs is lost [1]. As a result, to improve the performance and to avoid the unwanted oscillations for practical applications, joint flexibility must be taken into account in both modeling and control. However, to simplify the complexity of the controller design, most controllers for industrial robots are designed based on the rigid-robot assumption [2]. As a result of considering the actuator dynamics and joint flexibility, the controller design would become extremely complex. Therefore, the modeling and control of the flexible-joint robots are more difficult than those of the rigid robots [3].

The torques are the inputs to the system equations. However, in many papers, such as the feedback linearization method [4], the adaptive sliding mode technique [5], and the proportional-derivative control approach [6] dynamics of the actuators for providing the desired torques are excluded [7]. It has been shown that actuator dynamics form an important part of the complete robot dynamics, especially in the cases of high-velocity movement and highly-varying loads [8]. One of the drawbacks of these previously-published results is that they require velocity measurements. Moreover, in practical robotic systems, the velocity measurements obtained through tachometers are contaminated by noise [9-11].

More specifically, the major limitation associated with the mentioned control schemes is that these schemes assume that torques can be directly applied to the robot links, i.e. the actuator dynamics is ignored and the control is designed at a dynamic level with the torque as input. Researchers often refer to this method as the torque-control strategy.

To solve the aforementioned problems, the voltage-control strategy was proposed [12]. In this strategy, the actuator dynamics is also taken into account, and the voltages of motors are considered as the inputs of the robotic system including the actuators and robot manipulators. Recently, robust control [13] and non-linear adaptive control [14] of flexible-joint robots have been developed using the voltage-control strategy. On the other hand, including the electrical sub-system of the actuator dynamics causes several challenging problems. Some of these reasons are as follow [15]: 1) The electrical sub-system increases the complexity of the system model such that a 5th order non-linear differential equation should be employed for describing a single link flexible joint robot [16]. 2) Any practical control system is subject to some upper and lower bounds that limit the actuator input command. In addition, there are unwanted non-linearities, which come from dynamical effects such as deadzone, backlash, and hysteresis. These constraints make the control design problem extremely difficult.

Alternatively, fuzzy control, as a model-free approach, can be easily designed to control non-linear uncertain systems [17]. So far, fuzzy control of robot manipulators has received considerable attention for overcoming uncertainty, non-linearity, and coupling [17-19]. The fuzzy adaptive control approaches are classified into two categories, direct adaptive fuzzy control and indirect adaptive fuzzy control algorithms [17]. In the direct adaptive fuzzy control, the fuzzy controller is a single fuzzy system constructed (initially) from the control knowledge. On the other hand, in the indirect adaptive fuzzy control, the fuzzy controller comprises a number of fuzzy systems constructed (initially) from the plant knowledge.

This paper discusses the problem of designing a novel direct adaptive fuzzy control for a class of flexible-joint robotic manipulators including actuator dynamics in the presence of uncertainties associated with both the robot and motor dynamics. An advantage is that it uses the voltage-control strategy instead of the torque-control strategy, which is simpler, less computational, and more effective than the torque-control strategy. An electrically-driven manipulator is then controlled via its motors as individual single-input/single-output systems. The design includes two interior loops: the inner loop controls the motor position using the proposed approach, while the outer loop controls the joint angle of the robot using a proportional-integral-derivative (PID) control law. In addition, performance of the control system is improved by optimizing the PID gains.

Another novelty of this paper is the use of a particle swarm optimization (PSO) algorithm for optimizing the control design parameters in order to achieve the desired performance. It is worthy of note that one of the advantages of the proposed method is that there is no need for a velocity measurement. Based on the Lyapunov stability theorem, the stability analysis was presented. Finally, simulations were conducted on two-link robotic manipulators to show the effectiveness of the proposed control scheme. As a result, the advocated design methodology not only assures a closed-loop stability but also a desired tracking performance can be achieved for the overall system.

The rest of the paper is organized as follows: Section 2 presents modeling of the flexible-joint robots. Section 3 introduces PSO. Section 4 develops the proposed method. Section 5 presents the simulation results, and finally, Section 6 concludes the paper.

2. Electrically-driven fexible-joint robot dynamics

Consider a flexible-joint robot, which is driven by geared permanent magnet dc motors. If the joint flexibility is modeled by a linear torsional spring, the dynamic equations of motion can be expressed as follow [13, 14]:

$$D(\theta)\ddot{\theta}+C(\theta,\dot{\theta})\dot{\theta}+g(\theta)+\tau_d = K(r\theta_m-\theta) \tag{1}$$

$$J\ddot{\theta}_m+B\dot{\theta}_m+rK(r\theta_m-\theta)=\tau \tag{2}$$

where, $\theta\in R^n$ is a vector of joint angles, $\theta_m\in R^n$ is a vector of rotor angles, and $\tau_d\in R^n$ denotes unknown disturbances including unstructured dynamics and unknown payload dynamics. Thus this system possesses $2n$ coordinates as $[\theta\ \theta_m]$. The matrix $D(\theta)$ is an $n\times n$ matrix of manipulator inertia, $C(\theta,\dot{\theta})\dot{\theta}\in R^n$ is the vector of centrifugal and Coriolis forces, $g(\theta)\in R^n$ is a vector of gravitational forces, and $\tau\in R$ is a torque vector of motors. The diagonal matrices J,B and r represent the coefficients of the motor inertia, motor damping, and reduction gear, respectively. The diagonal matrix K represents the lumped flexibility provided by the joint and reduction gear. To simplify the model, both the joint stiffness and gear coefficients are assumed constant. The vector of gravitational forces $g(\theta)$ is assumed a function of only the joint positions as used in the simplified model [20]. Note that the vector and matrix are represented in the bold form for clarity.

System (1-2) is highly non-linear, extensively computational, heavily coupled, and a multi-input/multi-output system with $2n$ coordinates. Complexity of the model has been a serious challenge in robot modeling and control in the literature. It is expected to face a higher complexity if the proposed model includes the actuator dynamics. In order to obtain the motor voltages as inputs, consider the electrical equation of the geared permanent magnet DC motors in the following matrix form:

$$u = RI_a + L\dot{I}_a + K_b\dot{\theta}_m \qquad (3)$$

where $u \in R^n$ is a vector of motor voltages, $I_a \in R^n$ is a vector of motor currents, and $\dot{\theta}_m$ is a vector of rotor velocities. The diagonal matrices R, L and K_b represent the coefficients of armature resistance, armature inductance, and back-emf constant, respectively. The motor torque τ as an input for dynamic equation (2), is produced by the motor currents as:

$$K_m I_a = \tau \qquad (4)$$

where, K_m is a diagonal matrix of the torque constants. Equations (1-4) form the robotic system such that the voltage vector u is the input vector, and the joint angle vector θ is the output vector.

The dynamics of the electrical robot (1)-(4) in the state space is formed as:

$$\dot{X} = f(X) + bu + \vartheta \qquad (5)$$

where:

$$f(X) = \begin{bmatrix} X_2 \\ D^{-1}(X_1)(-g(X_1) - KX_1 - C(X_1, X_2)X_2 \\ +Krx_3) \\ J^{-1}(rKX_1 - r^2KX_3 - BX_4 + K_mX_5) \\ -L^{-1}(K_bX_4 + RX_5) \end{bmatrix}$$

$$b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ L^{-1} \end{bmatrix}, X = \begin{bmatrix} \theta \\ \dot{\theta} \\ \theta_m \\ \dot{\theta}_m \\ I_a \end{bmatrix}, \vartheta = \begin{bmatrix} 0 \\ -D^{-1}(X_1)\tau_d \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where, $X_1 = \theta, X_2 = \dot{\theta}, X_3 = \theta_m, X_4 = \dot{\theta}_m$ and $X_5 = I_a$.

Equation (5) shows a highly-coupled non-linear large system, where the state vector X includes vectors of position and velocity to the motor and the joint and the motor current.

3. Particle swarm optimization

The particle swarm optimization (PSO) algorithm is a population-based search algorithm based on the simulation of the social behavior of birds within a flock [21]. This algorithm optimizes a problem by having a population of candidate solutions and moving these particles around in the search-space according to simple mathematical formulae over the particle position and velocity. The position of each particle is changed based on the experiences of the particle itself and those of its neighbors [22,23]. Consequently, the particles tend to fly towards the better searching areas over the searching space [24,25].

The velocity of the ith particle (v_i) is calculated as follows [23,26]:

$$\begin{aligned} v_i(t+1) &= w\,v_i(t) + c_1r_1(pbest_i(t) - \\ &x_i(t)) + c_2r_2(gbest(t) - x_i(t)) \end{aligned} \qquad (6)$$

where, in the ith iteration, x_i is the particle position, $pbest_i$ is the previous best particle position, $gbest$ is the previous global best position of particles, w is the inertia weight, c_1 and c_2 are the acceleration coefficients namely the cognitive and social scaling parameters, respectively, and r_1 and r_2 are two random numbers in the range of $[0\;1]$.

The new position of the ith particle is then calculated as [26]:

$$x_i(t+1) = x_i(t) + v_i(t+1) \qquad (7)$$

The PSO algorithm is performed repeatedly until the goal is achieved. The number of iterations can be set to a specific value as a goal of optimization. In addition, to enhance the performance of PSO based on the experimental results, the inertia weight was proposed to control the velocity, as [27]:

$$w = (w_1 - w_2)\left(\frac{iter_{\max} - iter}{iter_{\max}}\right)^n + w_2 \qquad (8)$$

where, n is the non-linear modulation index, w decreases from a higher value w_1 to a lower value w_2 and $iter_{\max}$ is the maximum iteration number. Moreover, proper fine-tuning of the parameters c_1 and c_2 in (6) may result in faster convergence of the algorithm and alleviation of the local minima [28]. Hence, c_1 and c_2 are given as:

$$c_1 = (c_{1i} - c_{1f})(\frac{iter_{\max} - iter}{iter_{\max}}) + c_{1f}$$
$$c_2 = (c_{2i} - c_{2f})(\frac{iter_{\max} - iter}{iter_{\max}}) + c_{2f} \qquad (9)$$

where, c_{1i} and c_{2i} are the initial values for the acceleration coefficient c_1 and c_2 and c_{1f} and c_{2f} are the final values for the acceleration coefficients c_1 and c_2, respectively [28, 29].

Simulations were carried out with various constraint optimization problems to find out the best ranges of values for c_1 and c_2. From the results, it was observed that the best solutions were determined when changing c_1 from 2.5 to 0.5 and changing c_2 from 0.5 to 2.5 over the full range of search [29].

4. Proposed control law

To control such a complicated system, a novel simple controller was proposed using a voltage-control strategy. The design includes two interior loops: the inner loop controls the motor position, while the outer loop controls the joint angle of the robot manipulator. The outer loop provides the desired trajectory for the inner loop.

4.1. Designing inner loop

In order to design the inner loop to control the motor position, the electrical equation for a permanent magnet dc motor is written as:

$$u = RI_a + L\dot{I}_a + k_b\dot{\theta}_m + \vartheta \qquad (10)$$

Where R, L and k_b denote the armature resistance, inductance, and back emf constant, respectively, u is the motor voltage, I_a is the motor current, θ_m is the rotor position, and ϑ represents the external disturbance (assumed to be bounded).

The motor angle θ_m, as an output, can be controlled via the voltage u, as an input. From (10) we have:

$$\dot{\theta}_m = \frac{1}{k_b}u - \frac{RI_a + L\dot{I}_a}{k_b} - \frac{\vartheta}{k_b} \qquad (11)$$

Using feedback linearization by assuming $\vartheta = 0$, a perfect control law can be obtained as:

$$u_{eq} = k_b(\frac{RI_a + L\dot{I}_a}{k_b} + \dot{\theta}_{md} + \alpha(\theta_{md} - \theta_m)) \qquad (12)$$

where, α is a positive gain, and θ_{md} is a desired motor angle.

Substituting (12) into (11), and after some manipulation, yields:

$$\dot{e} + \alpha e = 0 \qquad (13)$$

where, e is the tracking error, expressed by $e = \theta_{md} - \theta_m$. It can be concluded that the error approaches zero using the control law (12).

Feedback linearization is one of the popular techniques used in the non-linear control approaches. Feedback linearization can convert a multi-input/multi-output non-linear system to single-input/single-output linear decoupled systems. However, feedback linearization suffers from some problems such as model uncertainty and additional computations in control efforts. In fact, a perfect model is required to apply feedback linearization, while a perfect model is not available. Therefore, performances of control strategies are dependent on the model used in the feedback linearization [30].

From (12), one can note that the feedback linearization controller requires an exact cancellation of non-linearities to achieve the desired performance. In the presence of uncertainties, the non-linearities may not get canceled exactly, which may result in a poor performance, and thus, it is necessary to compensate for the effects of the uncertainties. To overcome this drawback, the following control law is proposed:

$$u = u_D + k_b u_s \qquad (14)$$

where, u_D is the output of an adaptive fuzzy system, and u_s can be considered as an extra term to overcome uncertainties. Substituting (14) into (11) gives:

$$\dot{\theta}_m = \frac{1}{k_b}u_D + u_s - \frac{RI_a + L\dot{I}_a}{k_b} - \frac{\vartheta}{k_b} + \frac{1}{k_b}u_{eq} \\ -\frac{1}{k_b}u_{eq} \qquad (15)$$

By substituting (12) into (15), one can obtain:

$$\dot{\theta}_m = \frac{1}{k_b}u_D + u_s - \frac{RI_a + L\dot{I}_a}{k_b} - \frac{\vartheta}{k_b} + \\ \frac{RI_a + L\dot{I}_a}{k_b} + \dot{\theta}_{md} + \alpha(\theta_{md} - \theta_m) - \frac{1}{k_b}u_{eq} \qquad (16)$$

After some simple manipulations, we have:

$$\dot{\theta}_m = \dot{\theta}_{md} + \alpha(\theta_{md} - \theta_m) + \frac{1}{k_b}(u_D \\ - u_{eq}) + u_s - \varphi \qquad (17)$$

where, $\varphi = \frac{\vartheta}{k_b}$. The role of u_s is presented by (17). It is quite obvious from (17) that u_s is employed to attenuate the external disturbance. Further simplification gives:

$$\dot{e} = -\alpha e + \frac{1}{k_b}(u_{eq} - u_D) + \varphi - u_s \qquad (18)$$

where, e is $\theta_{md} - \theta_m$.

We design a fuzzy controller using two variables as the inputs to the fuzzy controller namely the tracking error e and its derivative $\dot{e}$. If three membership functions are given to each fuzzy input, the whole control space is covered by nine fuzzy rules. The linguistic fuzzy rules are proposed in the form of:

$$R^l : if\ e\ is\ A^l\ and\ \dot{e}\ is\ B^l\ then\ y = y^l \qquad (19)$$

where, R^l denotes the lth fuzzy rule for $l = 1,\ldots,9$ In the lth rule, y^l is a crisp output, and A^l and B^l are the fuzzy membership functions belonging to the fuzzy variables e and $\dot{e}$, respectively. Three Gaussian membership functions, namely Positive (P), Zero (Z), and Negative (N) are defined for input e in the operating range of the manipulator. They are expressed as:

$$\mu_N(e) = \begin{cases} 1 & e \le -1 \\ 1-2(e-1)^2 & -1 \le e \le -.5 \\ 2e^2 & -.5 \le e \le 0 \\ 0 & e \ge 0 \end{cases}$$

$$\mu_p(\dot{e}) = \begin{cases} 0 & \dot{e} \le 0 \\ 2\dot{e}^2 & 0 \le \dot{e} \le .5 \\ 1-2(\dot{e}+1)^2 & .5 \le \dot{e} \le 1 \\ 1 & \dot{e} \ge 1 \end{cases} \qquad (20)$$

$$\mu_z(e) = \exp(-\frac{e^2}{2\sigma^2}),\ \sigma = .5$$

The membership functions of $\dot{e}$ are given the same as e. If we use the singleton fuzzifier and the center average defuzzifier, u_D is calculated by:

$$u(e|\Theta) = \sum_{l=1}^{9} y^l \xi_l(e,\dot{e}) = \Theta^T \xi \qquad (21)$$

where $\Theta^T = [y^1 \quad y^2 \quad \ldots \quad y^9], \xi = [\xi_1 \quad \xi_2 \quad \ldots \quad \xi_9]$ and $e = [e \ \ \dot{e}]$. In the meantime, ξ_1 is expressed as

$$\xi_l(e,\dot{e}) = \frac{\mu_{A^l(e)}\mu_{B^l(\dot{e})}}{\sum_{l=1}^{l=9}\mu_{A^l(e)}\mu_{B^L(\dot{e})}} \qquad (22)$$

where, $\mu_{A^l(e)}, \mu_{B^l(\dot{e})} \in [0\,1]$. The parameter Θ in (21) is determined by the adaptive rule afterward.

According to the universal approximation theorem, there exists an optimal fuzzy system $u_D^*(\mathbf{x}|\Theta)$ in the form of (21) such that:

$$u_D^*(e,\dot{e}) = \Theta^{*T}\zeta + \varepsilon \qquad (23)$$

where, ε is the approximation error, assumed to be bounded by $|\varepsilon| \le \rho$, where ρ is a positive scalar. Employing a fuzzy system to approximate u_D yields:

$$u_D(e|\Theta) = \Theta^{*T}\zeta \qquad (24)$$

where, $\hat{\Theta}$ is the estimated vector of Θ. One can obtain:

$$u_D^*(e|\Theta) - u_D(e|\Theta) = \tilde{\Theta}^T\zeta + \varepsilon \qquad (25)$$

where, $\tilde{\Theta} = \Theta^* - \hat{\Theta}$.

Using (18), and by adding and subtracting $u_D^*(e|\Theta)$, one can obtain:

$$\dot{e} = -\alpha e + \frac{1}{k_b}(u_D^*(e|\Theta) - u_D(e|\Theta)) - \frac{1}{k_b}w + \varphi - u_s \qquad (26)$$

where, $w = u_D^*(e|\Theta) - u_D$. Substituting (25) into (26) yields:

$$\dot{e} = -\alpha e + \frac{1}{k_b}(\tilde{\Theta}^T\zeta) - \frac{1}{k_b}w + \varphi + \in - u_s \qquad (27)$$

where, $\in = \frac{\varepsilon}{k_b}$. To establish convergence of the error, a lyapunove function is defined as:

$$V = 0.5e^2 + \frac{1}{2\gamma_1}\tilde{\Theta}^T\tilde{\Theta} \qquad (28)$$

where, γ_1 is a positive constant. Taking the time derivative of the above equation yields:

$$\dot{V} = e\dot{e} + \frac{1}{\gamma_1}\tilde{\Theta}^T\dot{\tilde{\Theta}} \qquad (29)$$

Substituting (27) into (29) yields:

$$\dot{V} = e(-\alpha e + \frac{1}{k_b}(\tilde{\Theta}^T\zeta) - \frac{1}{k_b}w + \varphi + \in - u_s) + \frac{1}{\gamma_1}\tilde{\Theta}^T\dot{\tilde{\Theta}} \qquad (30)$$

After some simple manipulations, one can obtain:

$$\dot{V} = -\alpha e^2 + \tilde{\Theta}^T(\frac{e}{k_b}\zeta + \frac{1}{\gamma_1}\dot{\tilde{\Theta}}) - \frac{e}{k_b}w + e(\varphi + \in - u_s) \qquad (31)$$

Suppose that $u_s = e$. Thus:

$$\dot{V} = -e^2(\alpha + 1) + \tilde{\Theta}^T(\frac{e}{k_b}\zeta + \frac{1}{\gamma_1}\dot{\tilde{\Theta}}) - \frac{e}{k_b}w + e(\varphi + \in) \qquad (32)$$

To establish the convergence, the following adaptive law is given:

$$\dot{\hat{\Theta}} = \gamma_1 e \zeta / k_b \tag{33}$$

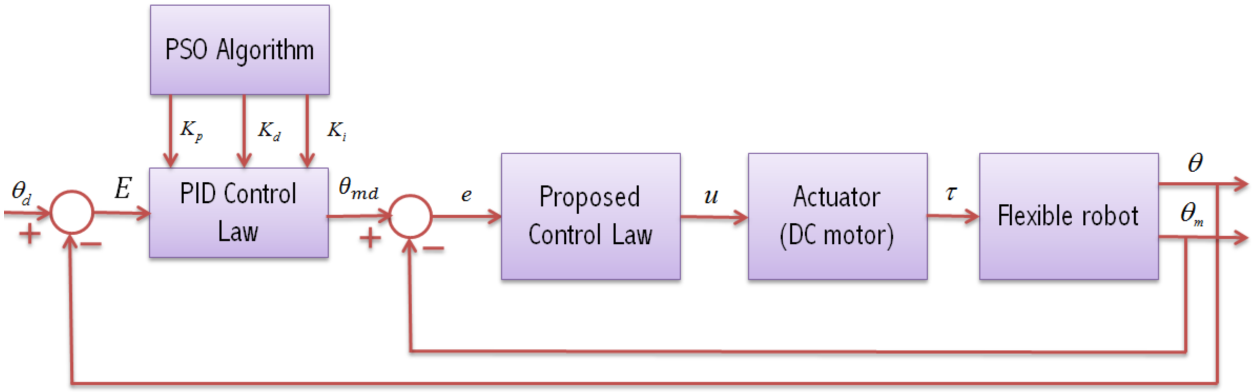


Figure 1. Block diagram of control system.

In order to evaluate the adaptive law (33), we substitute it into (32) to obtain:

$$\dot{V} = -e^2(\alpha+1) - \frac{e}{k_b}w + e(\varphi + \epsilon) \tag{34}$$

For achieving $\dot{V} \le 0$, (34) can be recast as:

$$\dot{V} \le -e^2(\alpha+1) - \frac{e}{k_b}w + |e||(\varphi + \epsilon)| \tag{35}$$

For achieving $\dot{V} \le 0$, it is required that:

$$e(\varphi + \epsilon) - \frac{e}{k_b}w \le 0 \tag{36}$$

From the universal approximation theorem, it can be expected that the terms ϵ and w should be very small, if not equal to zero, in the adaptive fuzzy system. Hence, using the Cauchy–Schwartz inequality, and by assuming that $|(\varphi + \epsilon)| \le \rho_1$, where ρ_1 is a positive scalar, we can obtain:

$$e(\varphi + \epsilon) \le |e|\rho_1 \tag{37}$$

Thus in order to satisfy (36), we suggest:

$$|e|\rho_1 = \frac{e}{k_b}w \tag{38}$$

Substituting for $\rho_1 = \frac{e}{|e|k_b}w$ in (35) gives:

$$\dot{V} \le -e^2(\alpha+1) \tag{39}$$

Therefore, $e \in L_\infty$, and $\int_0^\infty e^2 \le -(\alpha+1)^{-1}\int_0^\infty \dot{V} < \infty$ implies $e \in L_2$. In addition, we have already assumed that $\varphi \in L_\infty$. As a result, (27) implies $\dot{e} \in L_\infty$. Hence, asymptotic convergence of error can be concluded using the Barbalat's lemma. Namely, $\lim_{t\to\infty}|e(t)| = 0$.

4.2. Designing outer loop

The outer loop is designed to control the joint angle and providing the desired θ_{md} to the inner loop using a PID ordinary control law as:

$$\theta_{md} = k_d\dot{E} + k_p E + k_i \int E\, dt \tag{40}$$

where, k_p, k_d, and k_i are positive constant gains. $E = \theta_d - \theta$ denotes the joint tracking error, θ is the actual joint angle, and θ_d is the desired joint angle. To enhance the control system performance, the gains of PID control law are optimized using PSO. In addition, according to the proof given by [14], the robotic system is stable as well. As a conclusion, based on the stability analysis, all the signals required to form (14) are bounded.

To clarify the proposed control algorithm, a block diagram of the control system is depicted in figure 1.

In addition, to summarize the above analysis, a design procedure for the proposed approach is proposed as follows:

Step 1: Construct membership function for e and $\dot{e}$.

Step 2: Designing inner loop: Specify the desired coefficients γ_1 and α.

Step 3: Designing outer loop: Apply PSO algorithm, and find optimal specify coefficients k_p, k_d and k_i.

Step 4: Obtain the control law in (14), and apply it to the electrically-driven flexible-joint robot.

5. Simulation results

In this section, the proposed approach is applied to control a two-link flexible-joint robotic manipulator, as shown in figure 2, as described by [11,31]:

Table 1. Motor parameters.

Motors	R	k_b	L	J	B	r	K
1,2	1.26	0.26	0.001	0.0002	0.001	0.01	500

$$D(\theta) = \begin{bmatrix} m_1 l_1^2 + m_2(l_1^2 + l_2^2 + 2l_1 l_2 c_2) & m_2 l_2^2 + m_2 l_1 l_2 c_2 \\ m_2 l_2^2 + m_2 l_1 l_2 c_2 & m_2 l_2^2 \end{bmatrix} \quad (41)$$

$$C(\theta,\dot{\theta}) = \begin{bmatrix} -2m_2 l_1 l_2 s_2 \dot{\theta}_2 & m_2 l_1 l_2 s_2 \dot{\theta}_2 \\ m_2 l_1 l_2 s_2 \dot{\theta}_2 & 0 \end{bmatrix} \quad (42)$$

$$g(\theta) = \begin{bmatrix} m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1 \\ m_2 l_2 g c_{12} \end{bmatrix} \quad (43)$$

where, m_1 and m_2 are the masses of links 1 and 2, respectively; l_1 and l_2 are the lengths of links 1 and 2, respectively; s_i denotes $\sin(\theta_i)$, c_i denotes $\cos(\theta_i)$, c_{ij} denotes $\cos(\theta_i + \theta_j)$ for $i = 1,2$ and $j = 1,2$, and g is the acceleration of gravity. The parameters of the robot used for simulation are $l_1 = l_2 = 1m, m_1 = 10kg, m_2 = 8kg$ and $g = 9.8m/s^2$. The parameters of motors are given in table 1. Note that the inductances of motors are taken into account to consider a more complicated model in simulations.

The desired joint trajectory for the joints is smooth, expressed as $\theta_d = 1 - \cos(\pi t / 20)$, shown in figure 3.

The maximum voltage of each motor is set to $u_{\max} = 40v$. We set the adaptation law with $\hat{\Theta}(0) = 0$ and $\gamma_1 = 200$ for both motors.

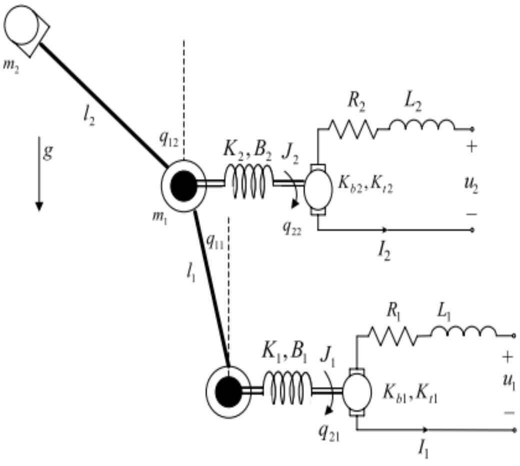


Figure 2. Flexible-joint two-link robot actuated by brushed DC motors [2].

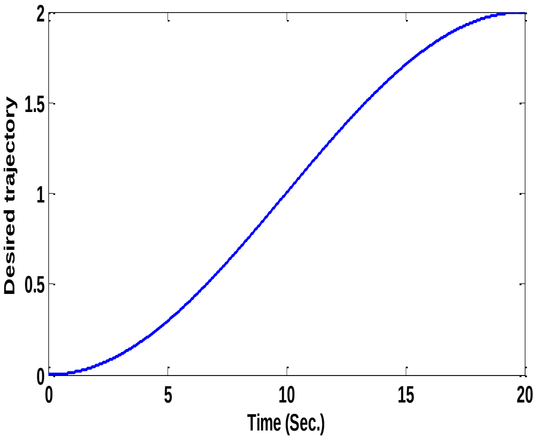


Figure 3. Desired trajectory for joints 1 and 2.

The PSO algorithm searches by 20 particles, and the maximum iteration $iter_{\max}$ is set to 100. The maximum inertia weight w_1 and the minimum inertia weight w_2 are given by 0.9 and 0.4, respectively. The coefficients c_{1i} and c_{2i} are set to 2.5 and 0.5, respectively. In addition, the coefficients c_{1f} and c_{2f} are set to 0.5 and 2.5, respectively. For the purpose of comparison, simulation studies in three cases are carried out.

Case 1: In this simulation, no uncertainty is considered. The PID gains are set to $k_p = 20, k_i = 10$ and $k_d = 2$.The performance of the control system is shown in figure 4. It is evident that the control system is performed well. The maximum value of tracking error is 0.135. The motor voltages behave well under the maximum permitted value of 40 V, as shown in figure 5. It is interesting to note that the control input is free of chattering. The adaptation of parameters in the adaptive law (33) is shown in figure 6. The simulation results confirm the effectiveness of the proposed method.

Case 2: In this simulation, to have a better comparison, and to enhance the control performance, the GA and PSO algorithms are used to optimize the PID control law gains in the outer loop. The search spaces of PID gains are defined as

$0 < k_p < 600, 0 < k_d < 600, 0 < k_i < 20.$

Before proceeding by the optimization operations, a performance criterion must be defined.

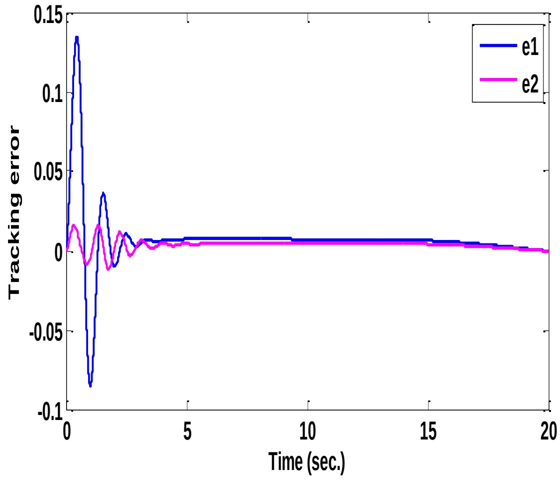


Figure 4. Tracking performance of proposed approach.

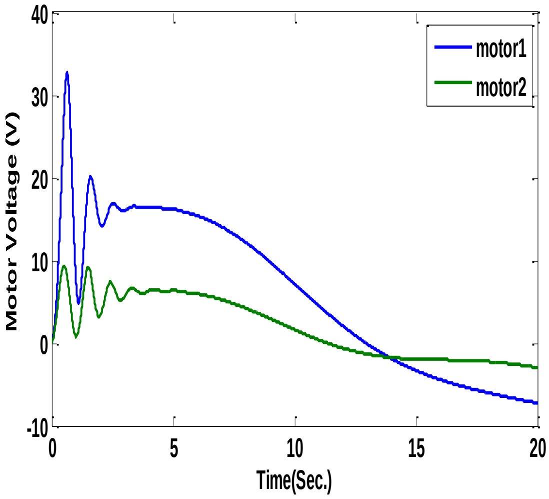


Figure 5. Control effort of proposed approach (motor voltages).

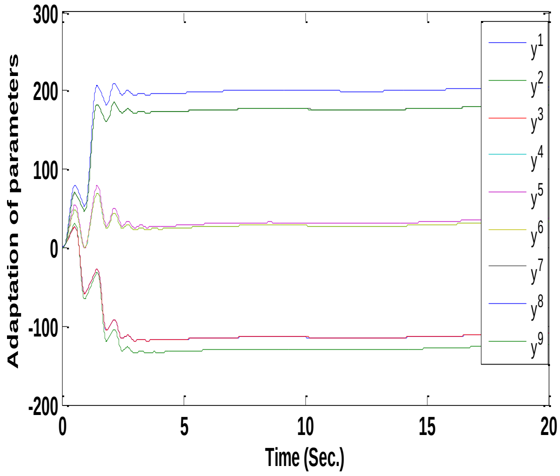


Figure 6. Adaptation of parameters.

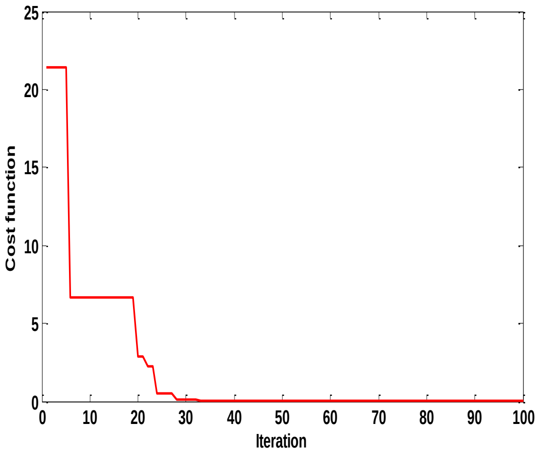


Figure 7. Desired Convergence of cost function.

In this paper, the cost function (CF) is defined as:

$$CF = \frac{1}{20}\int_0^{20} (e_1^2 + e_2^2)dt \qquad (44)$$

Owing to the randomness of the mentioned algorithms, their performance cannot be judged by the result of a single run. Many trials with different initializations should be made to acquire a useful conclusion about the performance of algorithms. An algorithm is robust if it gives consistent results during all the trials.

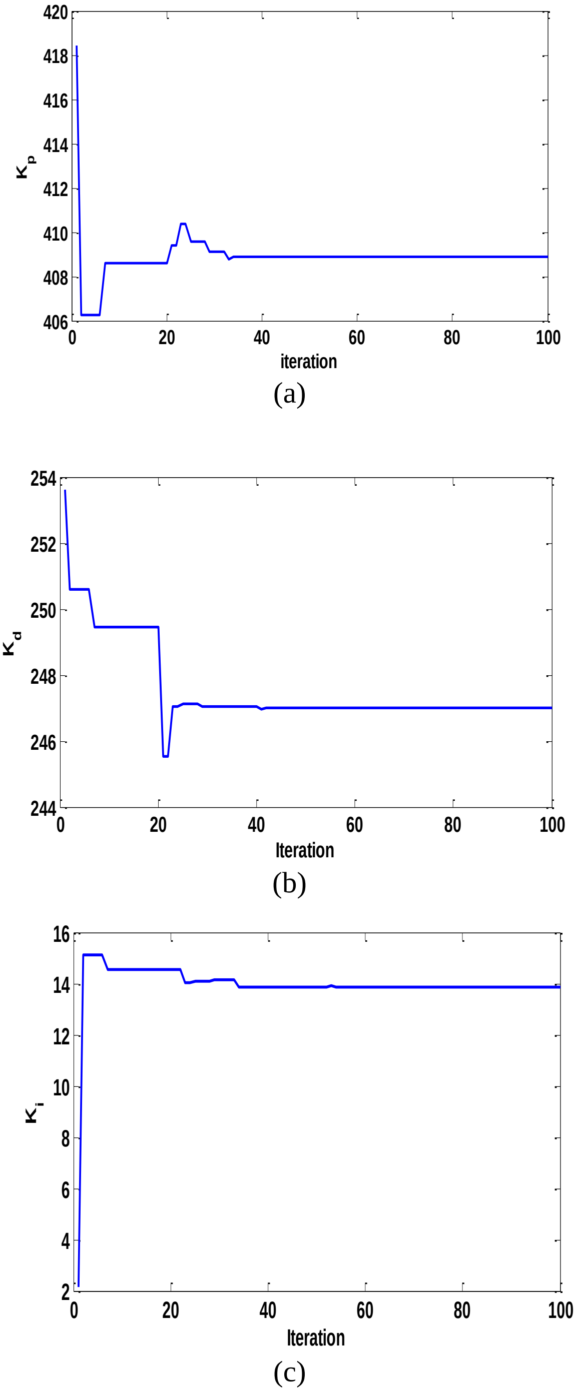
In order to run the PSO and GA algorithms, a population with a size of 20 for 100 iterations is used. Regarding (44), comparison of the results for 20 independent trials is shown in table 2.

Table 2. Comparison of optimized PID parameters.

Method	Best	Mean	Worst
GA	0.4316	0.6376	0.9376
PSO	0.1594	0.1604	0.1622

This comparison shows that PSO is superior to GA because the best and the mean values obtained by PSO are very close to the worst value. Hence, to save space afterward, just the PSO results are presented. In addition, the optimization process for PSO is depicted by figure 7 by calculating the cost function in the global best value at each run.

It is seen that the cost function is well converged. The trajectory of the PID gains is shown in figure 8. It confirms the success of the optimization process by using the PSO algorithm. As seen, the final values for the PID gains are found as $k_p = 408.89, k_d = 246.98, k_i = 13.87$. Figure 9 illustrates the tracking errors. The maximum value for the tracking error for joint 1 is about 0.083, i.e. about 61% of its value in case 1. It can be concluded that the PSO algorithm has worked well to enhance the control performance.



(a)

(b)

(c)

Figure 8. Finding PID gains: (a) trajectory of k_p, (b) trajectory of k_d, (c) trajectory of k_i.

Case 3: In this simulation, performance of the proposed approach in the presence of the uncertainties associated with both robot and motor is investigated. In this regard, for the robustness evaluation of the controllers, external disturbances are added to the robot system. The disturbance is inserted into the input of each motor as a periodic pulse function with a period of 2 S, amplitude 4 V, time delay of 0.7 S, and pulse width 30% of period. This form of disturbance is an example of any form that can be applied but it includes jumps to cover the complex cases.

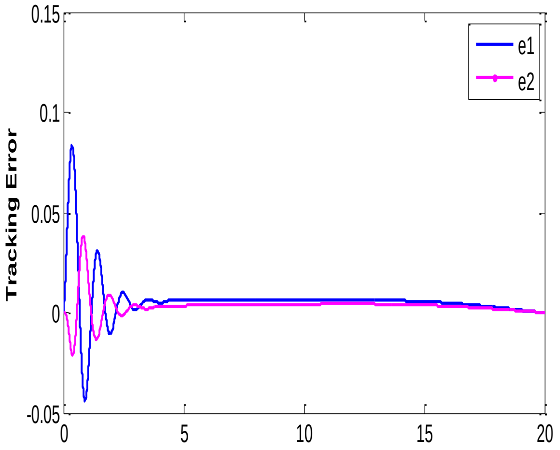


Figure 9. Tracking performance of proposed approach in optimal case.

We also choose the exogenous disturbances as: $\tau_d = [-3\cos(5t) \quad 3\sin(5t)]^T$. Moreover, the motors and robots' link parameters are considered to be 80% of their real values defined as before. The optimal values of PID gains that have been achieved using the PSO algorithm in Case 1 are used. The performance of the control system is shown in figures 10 and 11.

It is evident that the effects of disturbances are represented as small jumps on the curves of tracking errors. In addition, the tracking error in the presence of time-variant disturbance is, to some extent, more than the previous case. These figures show that not only there is no sign of chattering in the control inputs in the presence of the time-variant disturbance, but also they are smooth and in the permitted interval.

The simulation results thus demonstrate that the proposed approach can effectively control the flexible-joint robotic system with model uncertainties and disturbances.

Case 4: To have a better comparison, the proposed method is compared with a non-linear, approach proposed in [14]. The tracking performance of the mentioned approach in [14] is illustrated in figure 12. The maximum value for tracking error is 0.235, i.e. significantly larger than its value in case 2. Generally, it is evident that both methods have performed well. However, to some extent, the proposed approach has perform better.

6. Conclusion

This paper presents a direct adaptive fuzzy controller for flexible-joint electrically-driven robots, considering uncertainties in both the actuator and manipulator dynamics. The design includes two interior loops: the inner loop controls the motor position using the proposed approach, while the outer loop controls the joint angle of the robot using a PID control law. The proposed approach is based upon the voltage-control strategy, which is superior to the well-known torque-control strategy.

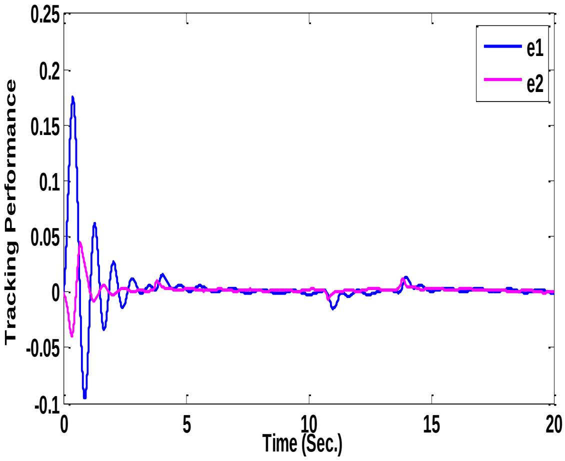


Figure 10. Tracking performance under disturbance.

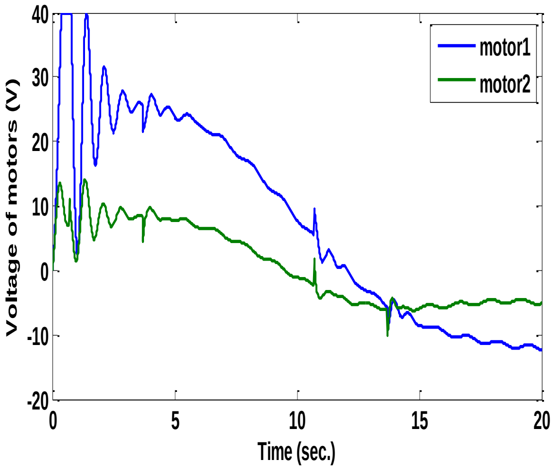


Figure 11. Motor voltages under disturbance.

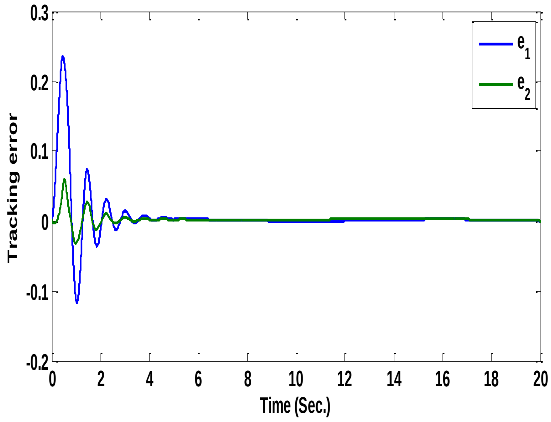


Figure 12. Tracking performance of proposed approach presented in [14].

The main advantage of our proposed methodology is that it uses available feedbacks as an important advantage from a practical viewpoint, the actuator dynamics is considered, and the control performance is also enhanced using the PSO algorithm. The stability analysis has verified the control method, and the simulation results have confirmed its effectiveness.

It is noted that the extension of the proposed method to the controller design for AC motors deserve further investigations. Practical implication of the proposed theoretical results is also part of our future works. It is worthy of note that from the simulation results, it can be concluded that the theoretic results obtained have potentials in applications.

References

[1] Brogliato, B., Ortega, R., & Lozano, R. (1995). Global tracking controllers for flexible-joint manipulators: a comparative study. Automatica, vol. 31, no. 7, pp. 941-956.

[2] Yen, H.-M., Li, T.-H. S., & Chang, Y.-C. (2012). Adaptive neural network based tracking control for electrically driven flexible-joint robots without velocity measurements. Computers & Mathematics with Applications, vol. 64, no. 5, pp. 1022-1032.

[3] Lewis, F. L., Abdallah, C. T., & Dawson, D. M. (1993). Control of robot manipulators, Macmillan New York.

[4] Spong, M. W., & Vidyasagar, M. (2008). Robot dynamics and control, John Wiley & Sons.

[5] Huang, A.-C., & Chen, Y.-C. (2004). Adaptive sliding control for single-link flexible-joint robot with mismatched uncertainties. Control Systems Technology, IEEE Transactions on, vol. 12, no. 5, pp. 770-775.

[6] De Luca, A., Siciliano, B., & Zollo, L. (2005). PD control with on-line gravity compensation for robots with elastic joints: Theory and experiments. Automatica, vol. 41, no. 10, pp. 1809-1819.

[7] Ozgoli, S., & Taghirad, H. (2006). A survey on the control of flexible joint robots. Asian Journal of Control, vol. 8, no. 4, pp. 332-344.

[8] Tarn, T., et al. (1991). Effect of motor dynamics on nonlinear feedback robot arm control. Robotics and Automation, IEEE Transactions on, vol. 7, no. 1, pp. 114-122.

[9] Li, Y., Tong, S., & Li, T. (2013). Adaptive fuzzy output feedback control for a single-link flexible robot manipulator driven DC motor via backstepping. Nonlinear Analysis: Real World Applications, vol. 14, no. 1, pp. 483-494.

[10] Nicosia, S., & Tomei, P. (1990). Robot control by using only joint position measurements. Automatic

Control, IEEE Transactions on, vol. 35, no. 9, pp. 1058-1061.

[11] Peng, J., Liu, Y., & Wang, J. (2014). Fuzzy adaptive output feedback control for robotic systems based on fuzzy adaptive observer. Nonlinear Dynamics, vol. 78, no. 2, pp. 789-801.

[12] Fateh, M. M. (2008). On the voltage-based control of robot manipulators. International Journal of Control, Automation, and Systems, vol. 6, no. 5, pp. 702-712.

[13] Fateh, M. M. (2012). Robust control of flexible-joint robots using voltage control strategy. Nonlinear Dynamics, vol. 67, no. 2, pp. 1525-1537.

[14] Fateh, M. M. (2012). Nonlinear control of electrical flexible-joint robots. Nonlinear Dynamics, vol. 67, no. 4, pp. 2549-2559.

[15] Izadbakhsh, A., & Fateh, M. (2014). Robust Lyapunov-Based Control of Flexible-Joint Robots Using Voltage Control Strategy. Arabian Journal for Science and Engineering, vol. 39, no. 4, pp. 3111-3121.

[16] Huang, A.-C., & Chien, M.-C. Design of a regressor-free adaptive impedance controller for flexible-joint electrically-driven robots, Proc. Industrial Electronics and Applications, 2009. ICIEA 2009. 4th IEEE Conference on, IEEE.

[17] Wang, L.-X. (1999). A course in fuzzy systems, Prentice-Hall press, USA.

[18] Vahedi, M., Hadad Zarif, M., & Akbarzadeh Kalat, A. (2016). An indirect adaptive neuro-fuzzy speed control of induction motors. Journal of AI and Data Mining, vol. 4, no. 2, pp. 243-251.

[19] Abdelhameed, M. M. (2005). Enhancement of sliding mode controller by fuzzy logic with application to robotic manipulators. Mechatronics, vol. 15, no. 4, pp. 439-458.

[20] Kim, E. (2004). Output feedback tracking control of robot manipulators with model uncertainty via adaptive fuzzy logic. Fuzzy Systems, IEEE Transactions on, vol. 12, no. 3, pp. 368-378.

[21] Spong, M. W. (1987). Modeling and control of elastic joint robots. Journal of dynamic systems, measurement, and control, vol. 109, no. 4, pp. 310-318.

[22] Engelbrecht, A. P. (2007). Computational intelligence: an introduction, John Wiley & Sons.

[23] Nickabadi, A., Ebadzadeh, M. M., & Safabakhsh, R. (2011). A novel particle swarm optimization algorithm with adaptive inertia weight. Applied Soft Computing, vol. 11, no. 4, pp. 3658-3670.

[24] Tungadio, D., et al. (2015). Particle swarm optimization for power system state estimation. Neurocomputing, vol. 148, pp. 175-180.

[25] Md Rozali, S., Fua'ad Rahmat, M., & Husain, A. R. (2014). Performance Comparison of Particle Swarm Optimization and Gravitational Search Algorithm to the Designed of Controller for Nonlinear System. Journal of Applied Mathematics, vol. 2014.

[26] Poli, R., Kennedy, J., &Blackwell, T. (2007). Particle swarm optimization. Swarm intelligence, vol. 1, no. 1, pp. 33-57.

[27] Shi, Y., & Eberhart, R. A modified particle swarm optimizer, Proc. Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on, IEEE.

[28] Chatterjee, A., & Siarry, P. (2006). Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. Computers & Operations Research, vol. 33, no. 3, pp. 859-871.

[29] Arumugam, M. S., Rao, M., & Chandramohan, A. (2008). A new and improved version of particle swarm optimization algorithm with global–local best parameters. Knowledge and Information systems, vol. 16, no. 3, pp. 331-357.

[30] Ratnaweera, A., Halgamuge, S. K., & Watson, H. C. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. Evolutionary Computation, IEEE Transactions on, vol. 8, no. 3, pp. 240-255.

[31] Slotine, J.-J. E., & Li, W. (1991). Applied nonlinear control, Prentice-Hall Englewood Cliffs, NJ.

[32] Ho, H., Wong, Y.-K., & Rad, A. B. (2007). Robust fuzzy tracking control for robotic manipulators. Simulation Modelling Practice and Theory, vol. 15, no. 7, pp. 801-816.

نشریه هوش مصنوعی و داده کاوی

کنترل فازی تطبیقی مستقیم ربات با مفاصل انعطاف‌پذیر با در نظر گرفتن دینامیک محرکه با استفاده از الگوریتم پرندگان

مجید مرادی زیرکوهی* و صمصام ایزد پناه

گروه برق، دانشگاه صنعتی خاتم النبیاء بهبهان، بهبهان، ایران.

ارسال ۲۰۱۵/۰۵/۰۵؛ پذیرش ۲۰۱۶/۰۸/۱۸

چکیده:

در این مقاله، یک سیستم فازی تطبیقی مستقیم برای کنترل ربات‌های با مفاصل انعطاف پذیر پیشنهاد می‌شود. روش طراحی شامل دو حلقه تو در تو است: در حلقه داخلی موقعیت موتور با استفاده از روش پیشنهادی کنترل می‌شود و در حلقه خارجی زوایای مفصل ربات با استفاده از کنترل کننده PID کنترل می‌شود. یکی از نوآوری‌های این مقاله استفاده از الگوریتم بهینه‌سازی پرندگان بمنظور بهینه کردن پارامترهای سیستم کنترل برای رسیدن به عمکرد مطلوب است. لازم به ذکر است که در تشکیل قانون کنترل با درنظر گرفتن ملاحظات عملی فقط از سیگنال‌های در دسترس فیدبک گرفته شده است. این مسئله در کاربردهای صنعتی سودمند است. روش پیشنهادی دارای پاسخ سریع با عمکرد ردگیری مناسب و سیگنال کنترلی همواری است. پایداری سیستم کنترل در حضور عدم قطعیت‌های پارامتری و غیر پارامتری اثبات می‌شود. در نتیجه تمام سیگنال‌های سیستم کراندار می‌باشند. شبیه‌سازی سیستم کنترل برای یک ربات دو لینکی با مفاصل انعطاف‌پذیر نشان از کارا بودن روش پیشنهادی دارد.

کلمات کلیدی: سیستم فازی، الگوریتم پرندگان، ربات با مفاصل انعطاف‌پذیر، دینامیک محرکه.