International Journal of Power Electronics and Drive Systems (IJPEDS)
Vol. 14, No. 4, December 2023, pp. 2546~2556
ISSN: 2088-8694, DOI: 10.11591/ijpeds.v14.i4.pp2546-2556

Adaptive control based neural network sliding mode approach for two links robot

Siham Massou[1], Ismail Boumhidi[2]
[1]LABTIC Laboratory, Department of Information Systems and Communication, National School of Applied Science, University Abdel Malek Essadi, Campus Ziaten, Tangier, Morocco
[2]LISAC Laboratory, Department of Physics, Faculty of Sciences Dhar El Mehrez, University Sidi Mohamed Ben Abdellah, Fes-Atlas, Morocco

Article Info

Article history:

Received Nov 8, 2022
Revised Mar 15, 2023
Accepted Mar 30, 2023

Keywords:

Adaptive neural network
Back propagation
Lyapunov stability
Nonlinear system
Sliding mode control
Two links robots

ABSTRACT

In order to improve the control accuracy of the robot manipulator, the sliding mode control combined with the adaptive neural network (ANNSMC) is proposed. Sliding mode control (SMC) is a nonlinear control recognized for its efficiency, easy tuning and implementation, accuracy and robustness. However, higher amplitude of chattering is produced due to the higher switching gain to handle the large uncertainties. For the purpose of reducing this gain, the uncertain parts of the system are estimated using neural network (NN) with on-line training using back propagation (BP) technique. The results of the online interconnection weights between the input and the hidden layers and between the hidden and the output layers are injected offline in order to improve the network performance in term of the convergence speed. In order to reduce the response time caused by the online training, the obtained output and input weights are updated using the adaptive laws derived from the Lyapunov stability approach the proposed control ANNSMC has improved the convergence speed with 41.13% for the first link and 40.15% for the second link comparing to NNSMC. The simulation result illustrates the performance of the proposed approach by using MATLAB and the control action suggested did not manifest any chattering behavior.

Corresponding Author:

Massou Siham
Department of information systems and communication, National School of Applied Science
University Abdel Malek Essadi
Campus Ziaten, Tangier 90000, Morocco
Email: s.massou@uae.ac.ma

1. INTRODUCTION

The motion control design for two links robot is an intense area of research that has fascinate considerable focus, it's a big challenge because of the nonlinearities of the system and the uncertainties of their parameters. Furthermore, the right dynamic models are nearly hard to obtain because the system is described by a nominal model with major uncertainties, to name a few: external disturbance, internal friction, and payload parameter. Several strategies have been proposed to cope with parameter uncertainties including neural network (NN) based controls [1]-[22], neural adaptive proportional-integral-derivative (PID) control [11], the sliding mode control (SMC) [23]–[31], SMC and PID controller tuned by whale optimizer algorithm (WOA) [29], nonlinear model predictive control tuned by NN [32]–[34] , the adaptive fuzzy control [35]–[37], the particle swarm optimization (PSO) combined with NN and SMC [38], the combined PID and SMC [30] were, the self-tuning fuzzy PID-nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators is studied. Sliding mode control (SMC) is an important robust control, it provides a systematic approach to the problem of uncertainties, bounded external disturbance and nonlinearities [5], [30], [32]–[37].

Unfortunately, there is an undesirable chattering in the control effort and it involves extremely high control activity. It is worth mentioning that eliminating this chattering problem several solution are proposed to perform properly as the boundary layer solution [23]–[25]. However, this method in only efficient for small uncertainties. In case of large uncertainties, the unknown parts of the two-links robot are estimated by using a neural network architecture, this approach leads to fewer uncertainties and a lower switching gain as result.

The back propagation technique [4], [38], [39] is used to train the weights of the neural network on-line. The most important task is to ensure the system's asymptotical stability and the tracking error's convergence to zero. In this study, the online input and output weights are injected and offline changed using the adaptive rules obtained from the Lyapunov stability theorem [40], therefore the asymptotical stability of the system is ensured, and also the speed of convergence is widely improved and illustrated in simulation results. In addition, we used integral squared error ISE [41] to validate numerically the theoretical statements, and that gives outstanding results.

This paper is organized as follows. The dynamic model of two links robot systems is introduced in section 2. Section 3 covers the neural network design technique; outlines sliding mode control and illustrates the adaptive laws for modifying the input and output weights obtained from the Lyapunov synthesis approach and the stability proof is presented. Section 4 displays the simulation results obtained by using a two-link robot manipulator. Finally, the conclusion is addressed in section 5.

2. SYSTEM MODEL OF THE TWO LINKS ROBOT

Consider the dynamic model of the two links robot written as (1) [7].

$$M(q)\ddot{q} + V_m(q,\dot{q})\dot{q} + F\dot{q} + f_c(\dot{q}) + G(q) + \tau_d = \tau \quad (1)$$

With $q, \dot{q}, \ddot{q} \in \Re^2$represent respectively the joints position, velocity and acceleration. $M(q) \in \Re^{2\times2}$ is a symmetric, positive definite inertia matrix, $V_m(q,\dot{q}) \in \Re^{2\times2}$ is the centripetal and coriolis matrix, $F \in \Re^2$ denotes the viscous friction coefficients, $f_c(\dot{q}) \in \Re^2$ is the coefficients of coulomb friction, $G(q) \in \Re^2$ is the vector of gravitation, $\tau_d \in \Re^2$ represents unmodeled dynamics and bounded unknown disturbances, $\tau \in \Re^2$ is the input torque vector applied to the servo motor.

$$V_m(q,\dot{q}) = \begin{bmatrix} V_{m11} & V_{m12} \\ V_{m21} & V_{m22} \end{bmatrix}; M(q) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

$$M_{11} = \frac{m_2 l^2}{3} + \frac{4m_2 l^2}{3} + m_2 l^2 \cos(q_2)\ ; M_{12} = M_{21} = \frac{m_2 l^2}{3} + \frac{m_2 l^2}{2}\cos(q_2); M_{22} = \frac{m_2 l^2}{3}$$

With $q = [q_1 \quad q_2]^T$ the positions; $l = l_1 = l_2$: the lengths; m_1, m_2: the masses.

$$V_{m11} = -\frac{m_2 l^2}{2}\dot{q}_2 \sin(q_2), V_{m12} = -\frac{m_2 l^2}{2}(\dot{q}_2 + \dot{q}_1)\sin(q_2)$$

$$V_{m21} = \frac{m_2 l^2}{2}\dot{q}_1 \sin(q_2)\ , V_{m22} = 0\ ; G(q) = \begin{bmatrix} m_2 g l \cos(q_1) + \frac{m_1 g l}{2}\cos(q_1) + \frac{m_2 g l}{2}\cos(q_1 + q_2) \\ \frac{m_2 g l}{2}\cos(q_1 + q_2) \end{bmatrix}$$

With g is the gravitation term.

$$H = F\dot{q} + f_c(\dot{q}) = \begin{bmatrix} f_{d1}\dot{q}_1 + k_1 \, sgn(\dot{q}_1) \\ f_{d2}\dot{q}_2 + k_2 \, sgn(\dot{q}_2) \end{bmatrix}$$

System model described by its state space is as (2).

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_1(\underline{x}) + g_{11}(x_1,x_3)u_1 + g_{12}(x_1,x_3)u_2 + \zeta_1(\underline{x},t) \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_2(\underline{x}) + g_{21}(x_1,x_3)u_1 + g_{22}(x_1,x_3)u_2 + \zeta_2(\underline{x},t) \end{cases} \quad (2)$$

Where: $\zeta_1(\underline{x},t)$ and $\zeta_2(\underline{x},t)$ the unknown parts, $\tau = \underline{U} = [u_1 \quad u_2]^T = [\tau_1 \quad \tau_2]^T q = [x_1 \quad x_3]^T, \dot{q} = [x_2 \quad x_4]^T, \ddot{q} = [\dot{x}_2 \quad \dot{x}_4]^T$, $\underline{x} = [x_1 x_2 x_3 x_4]^T$ and $M^{-1} = \underline{g}(x_1, x_3) = \begin{pmatrix} g_{11}(x_1,x_3) & g_{12}(x_1,x_3) \\ g_{21}(x_1,x_3) & g_{22}(x_1,x_3) \end{pmatrix}$ with $g_{11}(x_1,x_3) > 0 and g_{22}(x_1,x_3) > 0$ and $\underline{f}(\underline{x}) = [f_1(\underline{x}) \quad f_2(\underline{x})]^T = -M^{-1}\{-V_m[x_2 \quad x_4]^T -$ H-G}.

3. ADAPTIVE NEURAL NETWORK SLIDING MODE CONTROL DESIGN

3.1. Controller design

This study examines the trajectory tracking problem of the robot manipulator mentioned previously. The challenge of trajectory tracking control is to construct a robot controller based on any of $x(0)\varepsilon R^n$, $x(t) - x_d(t) \to 0$as $t \to \infty$. Where $x(t)$ and $x_d(t)$ are respectively the trajectory and the desired trajectory of a robot. In the following, some variables are defined as (3).

$$\xi(\underline{x},t) \rightleftarrows\rightleftarrows= [\xi_1(\underline{x},t) \rightleftarrows\rightleftarrows\rightleftarrows \xi_2(\underline{x},t)] \tag{3}$$

The system's output tracking error can be described as (4).

$$e = q - q_d = [x_1 - x_{1d} \quad x_3 - x_{3d}]^T$$

$$\begin{aligned} \dot{e} &= \dot{q} - \dot{q}_d = [\dot{x}_1 - \dot{x}_{1d} \quad \dot{x}_3 - \dot{x}_{3d}]^T \\ \ddot{e} &= \ddot{q} - \ddot{q}_d = [\ddot{x}_1 - \ddot{x}_{1d} \quad \ddot{x}_3 - \ddot{x}_{3d}]^T \end{aligned} \tag{4}$$

With x_{1d} and x_{4d} are the desired output. $r = 2$ is the relative degree of the system (2), and the sliding surface is characterized as (5).

$$S = \dot{e} + \beta e \tag{5}$$

β is designed diagonal matrix as $\beta = \begin{pmatrix} \beta_{11} & 0 \\ 0 & \beta_{22} \end{pmatrix}$. The selection of β must satisfy the following Hurwitz polynomial as (6).

$$\begin{aligned} s^{(2)} + \beta_{11} s^{(1)} = 0 \\ s^{(2)} + \beta_{22} s^{(1)} = 0 \end{aligned} \tag{6}$$

With $s^{(i)} = \frac{d^i(s)}{dt^i}$. The sliding variable derivative is (7).

$$\dot{S} = f_n + \xi(\underline{x},t) + g_n u - \begin{pmatrix} \dot{x}_{2d} \\ \dot{x}_{4d} \end{pmatrix} + \beta\dot{e} \tag{7}$$

The following is the robust adaptive controller that was used (8).

$$\underline{u} = \underline{u}_n + \underline{u}_s \tag{8}$$

with $\underline{u}_s = -k sign(S)$. The control law that respect (8) is (9),

$$\underline{u} = g_n^{-1}(\underline{x}) \left(-f_n(\underline{x}) + \begin{pmatrix} \dot{x}_{2d} \\ \dot{x}_{4d} \end{pmatrix} - \beta\dot{e} \right) - k sign(S) \tag{9}$$

where $sign(\,)$is the sign function given by:

$$sign(S) = \begin{cases} 1 \text{if} S > 0 \\ 0 \text{if} S = 0 \\ -1 \text{if} S < 0. \end{cases}$$

In order to compensate the uncertainties, the positive switching gain k is designed as $B < k$. With B the upper bound of uncertainties given by $\|\xi(\underline{x},t)\| < B$. The boundary layer method may be utilized to remove the chattering effect induced by the discontinuous control law. The control is now as (10),

$$\underline{u} = g_n^{-1}(\underline{x})\left(-f_n(\underline{x}) + \begin{pmatrix}\dot{x}_{2d}\\ \dot{x}_{4d}\end{pmatrix} - \beta\dot{e}\right) - ksat(S) \tag{10}$$

with the saturation function is described by (11),

$$sat(S) = \begin{cases} S/\delta \text{if} \|S\| < \delta \\ sgn(S) \text{otherwise} \end{cases} \tag{11}$$

with δ is the boundary layer thickness.

This approach is suitable for systems with minor uncertainty. We suggest using NN to estimate the uncertain terms of the system presented in (1) for large uncertain systems, in order to keep system uncertainties manageable. Let denote the prediction of unknown nonlinear functions parts as: $\hat{\xi}(\underline{x},t) = \left[\hat{\xi}_1(\underline{x},t) \quad \hat{\xi}_2(\underline{x},t)\right]^T$, with as (12):

$$\varepsilon(\underline{x},t) = \xi(\underline{x},t) - \hat{\xi}(\underline{x},t) \tag{12}$$

and: $\|\xi(\underline{x},t)\| < \varepsilon^*$ where ε^* is the network prediction errors' upper bound.

- Theorem 1: Consider the robot manipulator stated in (1) in the context of significant uncertainty. If the system control is programmed as (13):

$$\underline{u} = g_n^{-1}(\underline{x})\left(-\left(f_n(\underline{x}) + \hat{\xi}(\underline{x},t)\right) + \begin{pmatrix}\dot{x}_{2d}\\ \dot{x}_{4d}\end{pmatrix} - \beta\dot{e} - ksat(S)\right) \tag{13}$$

  with $\varepsilon^* < k$ and $\hat{\xi}(\underline{x},t)$ is predicted by the proposed off-line NN structure, then the trajectory tracking errors converge to zero in limited time.
- Proof. Consider the potential Lyapunov function: $V = \frac{1}{2}S^TS$ then $\dot{V} = S^T\dot{S}$ replacing the expression of $\dot{S}$ given in (7) we have:

$$\dot{V} = S^T(f_n + \xi(\underline{x},t) + g_n\underline{u} + \begin{pmatrix}\dot{x}_{2d}\\ \dot{x}_{4d}\end{pmatrix} + \gamma\ddot{e} + \beta\dot{e})$$

  by substituting the expression of $\underline{u}$ provided in the theorem we get:

$$\begin{aligned}\dot{V} = S^T(\xi(\underline{x},t) - \hat{\xi}(\underline{x},t) - ksat(S)) = S^T\varepsilon(\underline{x},t) - kS^Tsat(S) \le \|S^T\|\|\varepsilon(\underline{x},t)\| - kS^Tsat(S)\\ \le \|S^T\|\varepsilon^* - kS^Tsat(S)\end{aligned}$$

  by selecting $\varepsilon^* < k$, with k s a minor gain that is just responsible for compensating network faults forecast, we have: if $\|S\| \ge \delta$ then we take $sat(S) = sign(S)$ for any $\delta > 0$ and the function $\dot{V} = (\varepsilon^* - k)\|S\| < 0$. However, we take $sat(S) = \frac{S}{\delta}$ as continuous function in a boundary layer (a small δ -vicinity of the origin), as a result the system trajectories are restricted to a sliding mode manifold boundary layer. $S = 0$. Basing on (10), (8) and (4) we can write as (14).

$$\ddot{e} = A\dot{e} + B\tilde{\zeta}(\underline{x},t) - Bg_n(\underline{x})\underline{u}_s \tag{14}$$

  The neural controller aim is to drive the system output to match the defined intended reference trajectory as closely as feasible. The suggested neural network's design technique will be described in the following paragraph.

3.2. Neural network representation

We investigate a NN with two layers of tunable weights in this study [8]. The architecture used is illustrated in Figure 1 with one hidden layer, where $\underline{x}$ is the state input variables: the joints position, and the output variables are the unknown parts given by (2): $y_1 = \hat{\xi}_1(\underline{x},t)$ and $y_2 = \hat{\xi}_2(\underline{x},t)$.

$y_k = W_k^T\sigma(W_j^T\underline{x})$ $k = 1,2$ Where $\sigma()$is the activation function for hidden-layer considered as a sigmoid $\sigma(s) = \frac{1}{1+e^{-s}}$ function given by $W_k = [W_{k1}W_{k2}W_{kN}]^T$ and $W_j = \left[W_{j1}W_{j2}W_{jN}\right]^T$ are the connectivity weights between the hidden and output layers, as well as between the input and hidden layers. The actual output $y_{dk}(\underline{x})$ which is the difference between the actual and nominal functions is (15).

$$y_{dk}(\underline{x}) = y_k(\underline{x}) + \varepsilon(\underline{x}) \tag{15}$$

Where the approximation error of NN is $\varepsilon(\underline{x})$. The weights of the network are modified during the offline implementation. The technique used is based on the BP algorithm, it's a gradient-descent method and widely used. The back propagation technique is used to determine the necessary modifications after randomly selecting the network weights. The algorithm may be broken down into five steps:
i) Forward propagation of operating signal
ii) Calculate loss function
iii) Back propagation of error signal
iv) Calculate gradients
v) Weight updates
When the value of the error function becomes sufficiently minimal, the algorithm is terminated. It worth mentioning that BP algorithm converges slowly. In order to improve the speed of convergence, the adaptive principles obtained from the Lyapunov stability theorem is used to update online the obtained inputs and outputs weights. The detailed steps of the proposed algorithm are illustrated in [40].

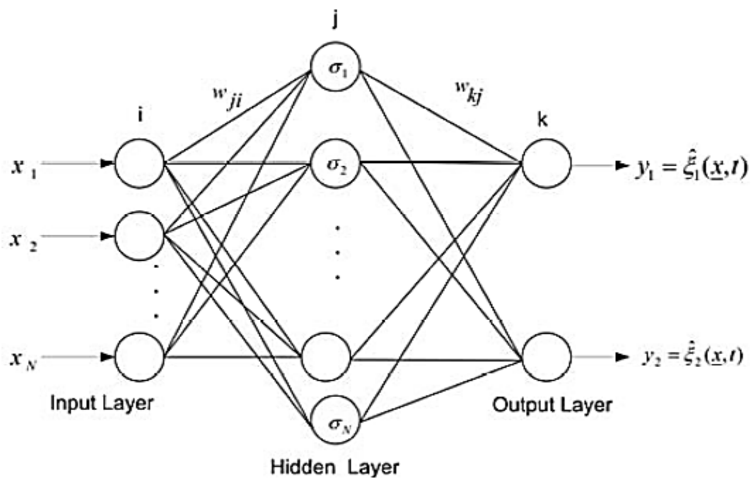

Figure 1. A multilayer neural network's design for predicting unknown components

3.3. Implementation of adaptation laws

The network weights are adjusted using the hybrid BP algorithm which takes important time to have a result, to deal with this time response weights are adjusted offline. In this case the output of ANN with 5 hidden nodes can be presented by (16).

$$\tilde{\zeta}(\underline{x}, t) = W_k * \sigma_k(x, W_j) \tag{16}$$

The parameters W_k and W_j need to be adjusted further for the purpose to minimize approximation errors. The adaptive rules for them were developed as (17) [23],

$$\begin{cases} \dot{W}_k = -\eta_1 \sigma_k^T B^T P \dot{e} \\ \dot{W}_j = -\eta_2 x^T B^T P \dot{e} \end{cases} \tag{17}$$

where η_1 and η_2 are constants that are always positive. P is the positive and symmetric definite matrix that corresponds to:

$$J = -(A^T P + PA) \tag{18}$$

where the designer selected J as a asymmetric definite matrix.

- Theorem 2: Suppose the nonlinear system described by (1). If the adaptive neural control rule mentioned in (13) is used with the parameter adaptation laws (17), as a result, the tracking errors converge to zero as $t \to \infty$ and all signals in the closed-loop system are limited.
- Proof: Take into consideration the possible Lyapunov function, which is:

$$\dot{V} = \frac{1}{2}\dot{e}^T P\dot{e} + \frac{1}{2\eta_1} W_k^T W_k$$

The Lyapunov function's derivative is stated as:

$$\dot{V} = \frac{1}{2}(\ddot{e}^T P\dot{e} + \dot{e}^T P\ddot{e}) + \frac{1}{\eta_1}\dot{W}_k^T W_k$$

using in (14) we have:

$$\dot{V} = \frac{1}{2}((A\dot{e} + B\tilde{\zeta}(\underline{x},t) - Bg_n(\underline{x})\underline{u}_s)^T P\dot{e} + \dot{e}^T P(A\dot{e} + B\tilde{\zeta}(\underline{x},t) - Bg_n(\underline{x})\underline{u}_s)) + \frac{1}{\eta_1}\dot{W}_k^T W_k$$

applying (16) and (17) we get:

$$\dot{V} = \frac{1}{2}((A\dot{e} + B\tilde{\zeta}(\underline{x},t) - Bg_n(\underline{x})\underline{u}_s)^T P\dot{e} + \dot{e}^T P(A\dot{e} + B\tilde{\zeta}(\underline{x},t) - Bg_n(\underline{x})\underline{u}_s)) + \frac{1}{\eta_1}(-\eta_1\sigma_k^T B^T P\dot{e})^T W_k$$

$$\dot{V} = \frac{1}{2}\dot{e}^T(A^T P + PA)\dot{e} + \frac{1}{2}(\tilde{\zeta}^T(\underline{x},t)B^T P\dot{e} + \dot{e}^T PB\tilde{\zeta}(\underline{x},t)) - \frac{1}{2}(\underline{u}_s^T B^T g_n^T(\underline{x})P\dot{e} + \dot{e}^T PBg_n(\underline{x})\underline{u}_s) - \dot{e}^T PB\,\sigma_k W_k$$

P is symmetric, we get:

$$\dot{V} = \frac{1}{2}\dot{e}^T(A^T P + PA)\dot{e} - \frac{1}{2}\underline{u}_s^T B^T g_n^T(\underline{x})P\dot{e}$$

$$\dot{V} \le \frac{1}{2}\dot{e}^T J\dot{e} - \frac{1}{2}\|\underline{u}_s^T\|\|B^T g_n^T(\underline{x})P\|\|\dot{e}\| \le 0$$

Hence $\dot{V}$is negative semi definite, the signals $\dot{e}$and W_k are all bounded. The parameters W_k and W_j described in (17) are adjusted using the projection algorithm as (19):

$$\dot{W}_k = \begin{cases} -\eta_1\sigma^T B^T Pe \; if \; \|W_k\| < M_B or \\ \quad if \begin{cases} \|W_k\| = M_B \quad and \\ e^T PB\sigma^T W_k \ge 0 \end{cases} \\ -\eta_1\sigma^T B^T Pe - \frac{e^T PB\sigma^T W_k}{\|W_k\|^2} W_k \quad if \begin{cases} \|W_k\| = M_B \; and \\ e^T PB\sigma^T W_k < 0 \end{cases} \end{cases}$$
$$\dot{W}_j = \begin{cases} -\eta_2 x^T B^T Peif \; \|W_j\| < M_{B2} or \\ \quad if \begin{cases} \|W_j\| = M_{B2} \quad and \\ e^T PBx^T W_j \ge 0 \end{cases} \\ -\eta_2 x^T B^T Pe - \frac{e^T PBx^T W_j}{\|W_j\|^2} W_j \quad if \begin{cases} \|W_j\| = M_{B2} \; and \\ e^T PBx^T W_j < 0 \end{cases} \end{cases} \qquad (19)$$

The utilization of projection algorithm has a good performance on the tracking trajectory and also in the control law illustrated in the next section.

3.4. Integral squared error (ISE)

In advance of talking about how to configure a controller, we must first define what makes a satisfactory response. It is a great challenge to take up. In reality, there are several metrics that may be used to compare the quality of regulated replies. The control measure described in this section is integral squared error (ISE) which is used to illustrate the success of the designed control ANNSMC. According to [41], the mathematical expression for ISE is (20),

$$\text{ISE} = \int e^2 dt \qquad (20)$$

where t is the simulation time, and e is the system's output tracking error.

4. SIMULATION RESULT

In this part, we put the suggested control strategy to the test on a two-link robot represented by the model (1). The aim of the control is to keep the system tracking the desired angle trajectory:$x_{1d} = \frac{\pi}{2} - \frac{\pi}{3}cos(0.5t)$ and $x_{3d} = (\pi/3)\,sin(0.5t)$. The criteria are thought to be :$\mathrm{m}_1 = \mathrm{m}_2 = 1; \mathrm{l}_1 = \mathrm{l}_2 = 1$ the initial conditions are: $q(0) = [x_1(0) \quad x_3(0)]^T = 0$,$\dot{q}(0) = \left[x_2(0) \quad x_4(0)\right]^T = 0$. The uncertainties under consideration are vector random noise with a value of unity $|\tau_d| \leq 1$. The following are the parameters connected with the controller design: $J = \begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix} M_B = 1.35, M_{B2} = 2, \eta_1 = 6, \eta_2 = 1.5$, A=[1 0;0 1], B=[1 0;0 1], P=[-50 0;0 -50], the number of hidden nodes is 5, the gain is $k_1 = k_2 = 1.5$.

The simulation results show position tracking for links 1 and 2, which are depicted respectively in Figure 2 and Figure 3, where the reference signal is represented by the dashed line (red), the results using adaptive law for weigh (ANNSMC) is represented by solid line (green), the results without adaptive law for weigh (NNSMC) is represented by dashed line (blue) and the results using conventional SM is represented by dot line (black). According to Figure 2, the gap between the ANNSMC outcomes position of link 1 and the reference value is high at first, but the system remains stable, and the system outputs faster converge to the intended trajectory than NNSMC, however the SMC result position don't converge as the ANNSMC result position. Similarly, the ANNSMC position of link 2 closely matches the reference signals and quickly than NNSMC. The corresponding control torque signals given in Figure 4 and Figure 5 are smooth even in the presence of significant uncertainty, there is no oscillatory behavior.

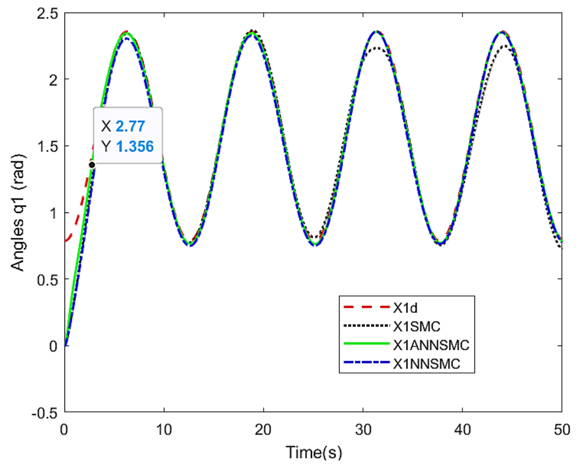

Figure 2. Angle response $x_{1ANNSMC}$ using adaptive law, x_{1NNSMC} angle response without adaptive law x_{1SMC} angle response with conventional SMC and desired trajectory x_{1d}

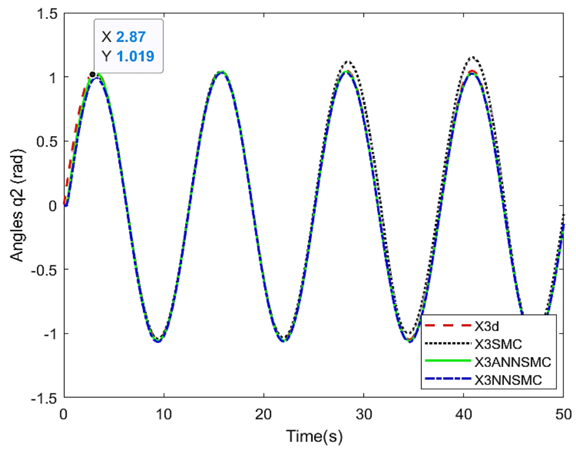

Figure 3. Angle response $x_{3ANNSMC}$using adaptive law, x_{3NNSMC} angle response without adaptive law x_{3SM} angle response using conventional SMC and desired trajectory x_{3d}

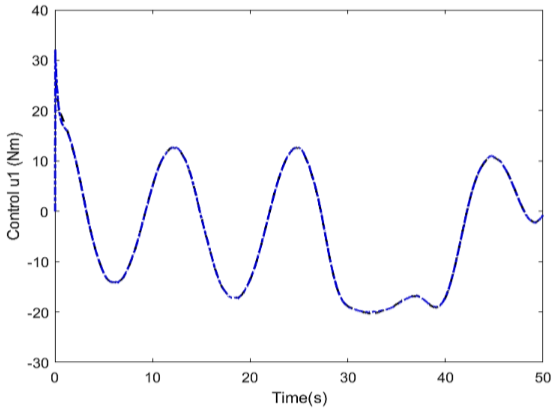

Figure 4. Control u_1 (input torque of join actuator 1)

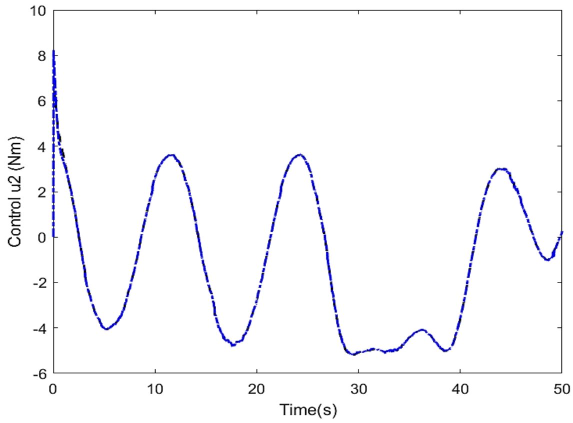

Figure 5. Control u_2 (input torque of join actuator 2)

In addition, the suggested adaptive neural control method's performance evaluation in term of the speed of convergence is investigated in Figure 6 and Figure 7, that represent the position tracking for link 1 and link 2 where the reference signals is represented by dash red line and the results without adaptive law for weight is represented by solid line in blue, by comparing the time response made by the results using this approach, Figure 2 and Figure 3, and the results without it, Figure 6 and Figure 7, it's obvious that the time response is widely improved. Furthermore, the error response is also reduced by using the proposed control and illustrated in Figure 8 and Figure 9.

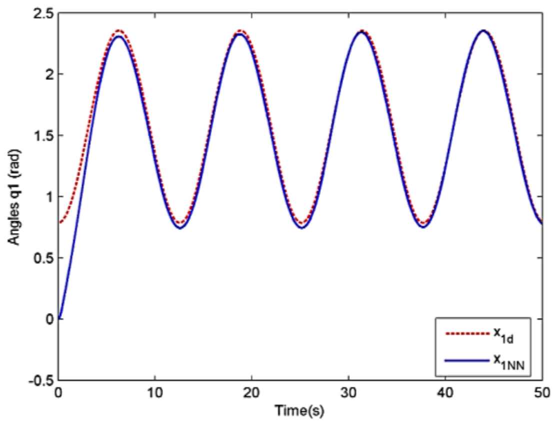

Figure 6. Angle response x_{1NNSMC} without adaptive law, desired trajectory x_{1d}

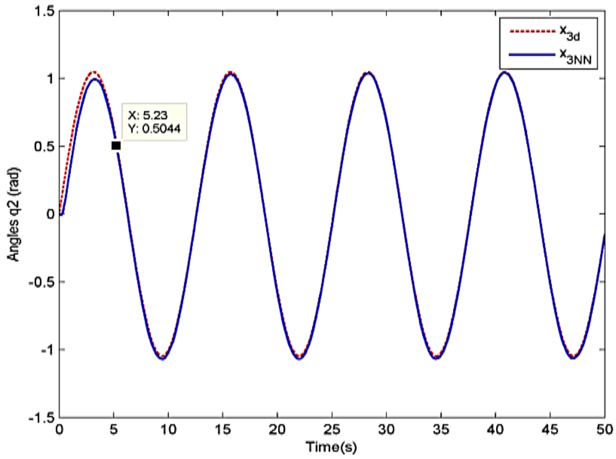

Figure 7. Angle response x_{3NNSMC} without adaptive law, desired trajectory x_{3d}

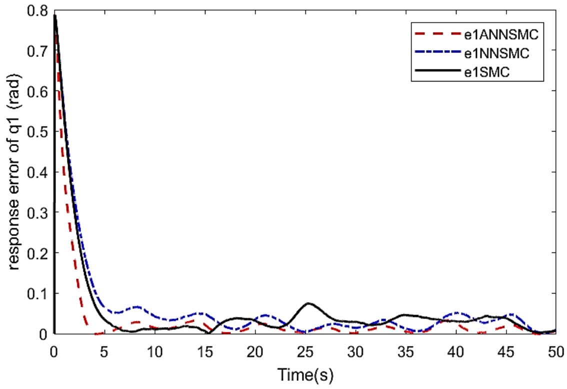

Figure 8. Error response e_{1NNSMC} , $e_{1ANNSMC}$ and e_{1SMC} of angle q_1, respectively without and with adaptive law and SMC

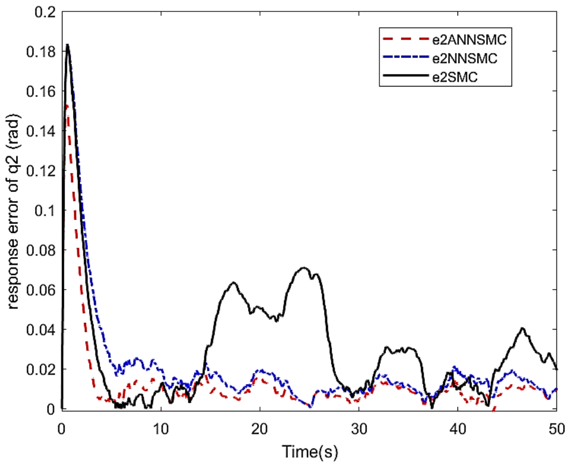

Figure 9. Error response e_{2NNSMC}, $e_{2ANNSMC}$ and e_{2SMC} of angle q_2, respectively without and with adaptive law and SMC

The performance comparison of the ANNSMC control in Table 1 shows that it outperforms the neural network sliding mode control (without the adaptive law derived based on Lyapunov theory). The control system designed to minimize ISE tend to rapidly reduce huge mistakes, this leads to fast responses This is seen in Figures 2, 3, 6, and 7.

Table 1. Comparative ISE for ANNSMC and SMC control

	NNSMC		ANNSMC		SMC	
	e_1	e_2	e_{1a}	e_{2a}	e_{1s}	e_{2s}
ISE	6.1398	80.2373	2.8940	47.241	9.6930	108.6011

5. CONCLUSION

The robust reference tracking problem for two-link robot manipulators was solved in this study. The developed method is a hybrid of the sliding mode control SMC technology and the adaptive neural network. ANN used online to extract weights adjusted firstly with BP algorithm. However, these weights are injected to be adjusted secondly by the adaptation rule obtained from the Lyapunov stability theorem. The ability of ANN to approximate fatly the uncertainties and external interference were assessed through comparison with an traditional SMC methods. Simulations conducted on computers of a two-link robot manipulator confirm outstanding results and show that the proposed algorithm can quickly approach the desired trajectory and effectively suppresses the chattering. Furthermore, the measure of ISE is used in addition to verify the theoretical statements achieved.

REFERENCES

[1] H. D. Patiño, R. Carelli, and B. R. Kuchen, “Neural networks for advanced control of robot manipulators,” *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 343–354, 2002, doi: 10.1109/72.991420.

[2] M. A. Hussain and P. Y. Ho, “Adaptive sliding mode control with neural network based hybrid models,” *Journal of Process Control*, vol. 14, no. 2, pp. 157–176, 2004, doi: 10.1016/S0959-1524(03)00031-3.

[3] P. X. Liu, M. J. Zuo, and M. Q. H. Meng, “Using neural network function approximation for optimal design of continuous-state parallel-series systems,” *Computers and Operations Research*, vol. 30, no. 3, pp. 339–352, 2003, doi: 10.1016/S0305-0548(01)00100-9.

[4] S. Sefriti, J. Boumhidi, R. Naoual, and Y. Boumhidi, “Adaptive neural network sliding mode control for electrically-driven robot manipulators,” *Control Engineering and Applied Informatics*, vol. 14, no. 4, pp. 27–32, 2012.

[5] Y. J. Liu, J. Li, S. Tong, and C. L. P. Chen, “Neural network control-based adaptive learning design for nonlinear systems with full-state constraints,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 7, pp. 1562–1571, 2016, doi: 10.1109/TNNLS.2015.2508926.

[6] Z. Liu, C. Chen, Y. Zhang, and C. L. P. Chen, “Adaptive neural control for dual-arm coordination of humanoid robot with unknown nonlinearities in output mechanism,” *IEEE Transactions on Cybernetics*, vol. 45, no. 3, pp. 507–518, 2015, doi: 10.1109/TCYB.2014.2329931.

[7] S.Ge, T. H Lee, and C. J. Harris, *Adaptive neural network control of robotic manipulators*. Singapore: World Scientific, 1998.

[8] F. W Lewis, S. Jagannathan, and A. Yesildirak, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. London: Taylor and Francis, 1999.

[9] L. Fu, *Neural Networks in Computer Intelligence*. New York: McGraw-Hill, 1995.

[10] H. N. Rahimi, I. Howard, and L. Cui, “Neural impedance adaption for assistive human–robot interaction,” *Neurocomputing*, vol. 290, pp. 50–59, 2018, doi: 10.1016/j.neucom.2018.02.025.

[11] H. Rahimi Nohooji, “Constrained neural adaptive PID control for robot manipulators,” *Journal of the Franklin Institute*, vol. 357, no. 7, pp. 3907–3923, May 2020, doi: 10.1016/j.jfranklin.2019.12.042.

[12] H. N. Rahimi, I. Howard, and L. Cui, “Neural adaptive tracking control for an uncertain robot manipulator with time-varying joint space constraints,” *Mechanical Systems and Signal Processing*, vol. 112, pp. 44–60, 2018, doi: 10.1016/j.ymssp.2018.03.042.

[13] Y. Song, B. Zhang, and K. Zhao, “Indirect neuroadaptive control of unknown MIMO systems tracking uncertain target under sensor failures,” *Automatica*, vol. 77, pp. 103–111, 2017, doi: 10.1016/j.automatica.2016.11.034.

[14] K. Yong, M. Chen, and Q. Wu, “Constrained adaptive neural control for a class of nonstrict-feedback nonlinear systems with disturbances,” *Neurocomputing*, vol. 272, pp. 405–415, 2018, doi: 10.1016/j.neucom.2017.07.015.

[15] Z. Chen, Z. Li, and C. L. P. Chen, “Adaptive Neural Control of Uncertain MIMO Nonlinear Systems With State and Input Constraints,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 6, pp. 1318–1330, 2017, doi: 10.1109/TNNLS.2016.2538779.

[16] R. Yang, C. Yang, M. Chen, and A. S. Annamalai, “Discrete-time optimal adaptive RBFNN control for robot manipulators with uncertain dynamics,” *Neurocomputing*, vol. 234, pp. 107–115, 2017, doi: 10.1016/j.neucom.2016.12.048.

[17] D. Zhang and B. Wei, “A review on model reference adaptive control of robotic manipulators,” *Annual Reviews in Control*, vol. 43, pp. 188–198, 2017, doi: 10.1016/j.arcontrol.2017.02.002.

[18] S. S. Ge and C. Wang, “Adaptive Neural Control of Uncertain MIMO Nonlinear Systems,” *IEEE Transactions on Neural Networks*, vol. 15, no. 3, pp. 674–692, May 2004, doi: 10.1109/TNN.2004.826130.

[19] Z. L. Tang, S. S. Ge, K. P. Tee, and W. He, “Adaptive neural control for an uncertain robotic manipulator with joint space constraints,” *International Journal of Control*, vol. 89, no. 7, pp. 1428–1446, 2016, doi: 10.1080/00207179.2015.1135351.

[20] Z. Yang, J. Peng, and Y. Liu, “Adaptive neural network force tracking impedance control for uncertain robotic manipulator based on nonlinear velocity observer,” *Neurocomputing*, vol. 331, pp. 263–280, 2019, doi: 10.1016/j.neucom.2018.11.068.

[21] C. Cao, F. Wang, Q. Cao, H. Sun, W. Xu, and M. Cui, “Neural network–based terminal sliding mode applied to position/force adaptive control for constrained robotic manipulators,” *Advances in Mechanical Engineering*, vol. 10, no. 6, 2018, doi: 10.1177/1687814018781288.

[22] S. Zhang, Y. Dong, Y. Ouyang, Z. Yin, and K. Peng, “Adaptive Neural Control for Robotic Manipulators with Output Constraints and Uncertainties,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5554–5564, 2018, doi: 10.1109/TNNLS.2018.2803827.

[23] J.J.E.~Slotine, “Sliding controller design for non-linear systems,” *International Journal of control*, vol. 40, no. 2, pp. 421–434, 1984.

[24] V. I. Utkin, *Sliding Modes in Control and Optimization*, 1st ed. Berlin: Springer Berlin Heidelberg, 1992. doi: 10.1007/978-3-642-84379-2.

[25] J. J. Slotine and S. S. Sastry, “Tracking Control of Non-Linear Systems Using Sliding Surfaces With Application To Robot Manipulators.,” *Proceedings of the American Control Conference*, vol. 1, pp. 132–135, 1983, doi: 10.23919/acc.1983.4788090.

[26] H. Li, P. Shi, D. Yao, and L. Wu, "Observer-based adaptive sliding mode control for nonlinear Markovian jump systems," *Automatica*, vol. 64, pp. 133–142, 2016, doi: 10.1016/j.automatica.2015.11.007.

[27] X. Yin, L. Pan, and S. Cai, "Robust adaptive fuzzy sliding mode trajectory tracking control for serial robotic manipulators," *Robotics and Computer-Integrated Manufacturing*, vol. 72, 2021, doi: 10.1016/j.rcim.2019.101884.

[28] Z. Zhou, H. Ji, and Z. Zhu, "Online sequential fuzzy dropout extreme learning machine compensate for sliding-mode control system errors of uncertain robot manipulator," *International Journal of Machine Learning and Cybernetics*, vol. 13, no. 8, pp. 2171–2187, 2022, doi: 10.1007/s13042-022-01513-x.

[29] F. Loucif, S. Kechida, and A. Sebbagh, "Whale optimizer algorithm to tune PID controller for the trajectory tracking control of robot manipulator," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 42, no. 1, 2020, doi: 10.1007/s40430-019-2074-3.

[30] M. Van, X. P. Do, and M. Mavrovouniotis, "Self-tuning fuzzy PID-nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators," *ISA Transactions*, vol. 96, pp. 60–68, 2020, doi: 10.1016/j.isatra.2019.06.017.

[31] H. Shi, Y. Liang, and Z. Liu, "An approach to the dynamic modeling and sliding mode control of the constrained robot," *Advances in Mechanical Engineering*, vol. 9, no. 2, 2017, doi: 10.1177/1687814017690470.

[32] A. A. Adeniran and S. El Ferik, "Modeling and Identification of Nonlinear Systems: A Review of the Multimodel Approach—Part 1," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1149–1159, Jul. 2017, doi: 10.1109/TSMC.2016.2560147.

[33] M. A. Ahmad, M. Z. M. Tumari and A. N. Kasruddin Nasir, "Composite Fuzzy Logic Control Approach to a Flexible Joint Manipulator," *International Journal of Advanced Robotic Systems*, vol. 10, 2012, doi: 10.5772/52562.

[34] L. Zhang, H. Liu, D. Tang, Y. Hou, and Y. Wang, "Adaptive Fixed-Time Fault-Tolerant Tracking Control and Its Application for Robot Manipulators," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 3, pp. 2956–2966, 2022, doi: 10.1109/TIE.2021.3070494.

[35] J. Zhou and Q. Zhang, "Adaptive Fuzzy Control of Uncertain Robotic Manipulator," *Mathematical Problems in Engineering*, vol. 2018, 2018, doi: 10.1155/2018/4703492.

[36] F. Qu, S. Tong, and Y. Li, "Observer-based adaptive fuzzy output constrained control for uncertain nonlinear multi-agent systems," *Information Sciences*, vol. 467, pp. 446–463, 2018, doi: 10.1016/j.ins.2018.08.025.

[37] A. Karamali Ravandi, E. Khanmirza, and K. Daneshjou, "Hybrid force/position control of robotic arms manipulating in uncertain environments based on adaptive fuzzy sliding mode control," *Applied Soft Computing Journal*, vol. 70, pp. 864–874, 2018, doi: 10.1016/j.asoc.2018.05.048.

[38] S. Massou and I. Boumhidi, "Optimal neural network-based sliding mode adaptive control for two-link robot," *International Journal of Systems, Control and Communications*, vol. 8, no. 3, p. 204, 2017, doi: 10.1504/ijscc.2017.10006534.

[39] D. E. Rumelhart and J. L. McClelland, "Learning Internal Representations by Error Propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, 1987, pp. 318–362.

[40] H. J. Rong, J. T. Wei, J. M. Bai, G. S. Zhao, and Y. Q. Liang, "Adaptive neural control for a class of MIMO nonlinear systems with extreme learning machine," *Neurocomputing*, vol. 149, no. Part A, pp. 405–414, 2015, doi: 10.1016/j.neucom.2014.01.066.

[41] B. Dhanasekaran, S. Siddhan, and J. Kaliannan, "Ant colony optimization technique tuned controller for frequency regulation of single area nuclear power generating system," *Microprocessors and Microsystems*, vol. 73, 2020, doi: 10.1016/j.micpro.2019.102953.

BIOGRAPHIES OF AUTHORS

Siham Massou is assistant professor of signals and information technology in department of information system and communication in National School of Applied Sciences, University Abdel Malek Essaadi. She received engineering degree from faculty of sciences and technology, University Abdel Malek Essaadi, Tangier, and Ph.D. degree from faculty Dhar El Mahraz, University Sidi Mohamed Ben Abdellah, Fes. Her research interests focus in the areas of fuzzy logic, neural network, optimization, nonlinear control systems and robust control. She can be contacted at email: s.massou@uae.ac.ma.

Ismail Boumhidi is a full Professor of Electronics and automation in department of physics in the Faculty of Sciences Dhar El Mahrez, University Sidi Mohamed ben Abdellah, Fez Morocco. He received advanced graduate studies degree and his Ph.D. from Faculty of Sciences Dhar El Mahrez. His research areas include control system synthesis, fuzzy control, fuzzy systems, nonlinear control systems, wind turbines, continuous time systems, reduced order systems. He can be contacted at email: iboumhidi@hotmail.com.