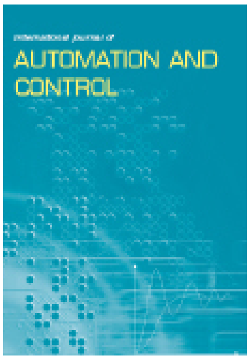


International Journal of Automation and Control

ISSN online: 1740-7524 - ISSN print: 1740-7516
https://www.inderscience.com/ijaac

Control of three-links robot arm based on fuzzy neural Petri nets

Salam Abdul Hady, Abduladhem A. Ali, Waleed I. Breesam, Ameer Lateef Saleh, Yasir I.A. Al-Yasir, Raed A. Abd-Alhameed

DOI: 10.1504/IJAAC.2023.10051797

Article History:

Received:	04 June 2020
Last revised:	30 May 2021
Accepted:	01 October 2021
Published online:	30 November 2022

Control of three-links robot arm based on fuzzy neural Petri nets

Salam Abdul Hady

Basra Oil Training Institute,
61019 Basra, Iraq
Email: salam11eng@gmail.com

Abduladhem A. Ali

Department of Computer Engineering,
University of Basrah, Iraq
Email: adbuladem1@yahoo.com

Waleed I. Breesam*

Basra Oil Training Institute,
61019 Basra, Iraq
Email: waleedbreesam@gmail.com
*Corresponding author

Ameer Lateef Saleh

Department of Electrical Engineering,
University of Misan, Iraq
Email: ameer-lateef@uomisan.edu.iq

Yasir I.A. Al-Yasir and Raed A. Abd-Alhameed

School of Electrical Engineering and Computer Science,
Faculty of Engineering and Informatics,
University of Bradford,
Bradford, UK
Email: Alyasir442@gmail.com
Email: r.a.a.abd@bradford.ac.uk

Abstract: A fuzzy neural Petri nets (FNPNs) controller is utilised for controlling a three-links robot arm which considers a nonlinear dynamic system. The incorporation of the classical FNN with a Petri net (PN) has been suggested to produce a new representing system called FNPN structure to alleviate the computation burden. The motion equation of three links robot arm is derived from Lagrange's equation. This equation has been incorporated with the motion equations of DC servo motors which motivate the robot. For nonlinearity dynamic problems, this paper presents a direct adaptive control technique to control three links robot arm utilising the FNPN controller. The computer simulation depicts that the present FNPN controller accomplished better performance with fast response and minimum error.

Keywords: fuzzy neural Petri net; FNPN; forward adaptive control; robot arm control.

Reference to this paper should be made as follows: Hady, S.A., Ali, A.A., Breesam, W.I., Saleh, A.L., Al-Yasir, Y.I.A. and Abd-Alhameed, R.A. (2023) 'Control of three-links robot arm based on fuzzy neural Petri nets', *Int. J. Automation and Control*, Vol. 17, No. 1, pp.116–131.

Biographical notes: Salam Abdul Hady received his BS and MS degrees from the Basrah University, Iraq, in 2004 and 2011, respectively, worked as an Automation Engineer, Instrumentation Engineer and Computers Engineer. His research interests include fuzzy neural, fuzzy neural petri nets, robotics control and IoT.

Abduladhem A. Ali received his MSc and PhD degrees from Department of Electrical Engineering University of Basrah, Iraq in 1983 and 1996. He worked as an Assistant Lecturer, Lecturer, and Assistant Professor at the same department at 1984, 1987 and 1981, respectively. Then, he is an Assistant Professor and Professor at the Department of Computer Engineering at 1997 and 2004, respectively. He is retired from University of Basrah at 2020 and currently an independent researcher. He worked as a consultant to many industrial firms to design industrial control systems. Have more than 100 published papers, one patent, supervised many MSc and PhD dissertations. He worked as an editorial board chair and member of many national and international journals and as a Chair for IEEE and ACM conferences. He is a senior member of IEEE and ACM.

Waleed I. Breesam has received his MSc in Electrical Engineering, Faculty of Engineering, Basra University in 2013. He is working a Lecturer, and Supervisor of Instrumentation and Control Laboratory in Basra Oil Training Institute 2016–present. In general, he has theoretical and practical experience in the following fields: renewable energy, motor speed control and drive, artificial intelligence and machine learning, process control and instrumentation, programmable logic controller and microcontrollers.

Ameer Lateef Saleh is a Lecturer at the Department of Electrical Engineering, University of Misan, Iraq. He holds a Master's in Electrical Engineering in Power and Machine specialisation from Basra University. His dissertation title was 'Speed control of brushless DC motor based on PID and wavelet neural networks'. He has more than 25 research papers published in international journals that cover fields: energy conversion, renewable energy, power electronics and converters, speed control of electrical motors, intelligent control techniques, automation and robotics and more.

Yasir I.A. Al-Yasir received his BSc in Electrical Engineering from University of Basra, Basra, Iraq in 2012. In 2014, he joined the research team of Antennas and Radio Frequency Engineering research group as a research visitor for four months at School of Electrical Engineering and Computer Science, Bradford University, UK. He has awarded the MSc degree in Electrical Engineering (Communications and Electronics) in 2015, from University of Basra, Basra, Iraq. He was appointed as a Lecturer at Southern Technical University, Basra, Iraq in 2015. Then, he was appointed as a Lecturer at Iraq University College in 2015 and 2016. In 2016 and 2017, he was working as an Electrical Engineer at Basra Oil Training Institute, Ministry of Oil in Iraq. Since January 2018, he has been appointed as Early-Stage Researcher (ESR) at School of Electrical Engineering and Computer Science, Bradford University, UK.

Raed A. Abd-Alhameed is a Professor of Electromagnetic and Radio Frequency Engineering at the University of Bradford, UK. He received his BSc and MSc degrees from Basrah University, in 1982 and 1985 respectively, and PhD degree from the University of Bradford, UK in 1997, all in Electrical Engineering. He has long years research experience over 25 years in the areas of radio frequency, communications systems, signal processing, antennas and electromagnetic computational techniques, and has published over 600 academic journal and conference papers; in addition he is co-authors of four books and several book chapters. He is the senior academic responsible for electromagnetics research in the communications research group, in which a new antenna design configurations and computational techniques have been developed including several patents were considered and filed.

1 Introduction

In recent, the intelligent control techniques such as neural-network, fuzzy logic control (FLC) and fuzzy-neural network (NN) vastly utilised for motion control of robotic applications (Wai and Chen, 2006; Huang and Lee, 2000; Yoo and Ham, 2000; Chang, 2000; Kim and Lewis, 2000). In popular, the robotic manipulators confront different difficulties related to the problems in their dynamics, such as disturbance, uncertainties, pay-load parameter and friction. Therefore, building a suitable mathematical model for these robots with their control system will face some difficulties. Thereby, the common demand on the intelligent control techniques is that they can alleviate the influences of parameter uncertainties and disorganised disturbance by utilising their robust learning capability without comprehensive knowledge of the system being controlled in the design procedures. The FLC has different features such as linguistic information, universal approximation theorem, robustness and rule-based algorithm (Wang, 1997).

Recently, the feed-forward NN has been utilised for many research applications for controlling dynamic systems and their identification (Omidvar and Elliott, 1997; Yang and Meng, 2003; Das et al., 2006). In addition, the research has been expanded utilising recurrent NN (Funahashi and Nakamura, 1993; Jin et al., 1995; Ku and Lee, 1995). For example, Jin et al. (1995) studied "the approximation of continuous-time dynamic systems using the dynamic recurrent (DRNN) and a Hopfield-type DRNN was presented by Funahashi and Nakamura (1993)."

As is vastly known, both NN systems and fuzzy logic systems strive to harness human-like knowledge processing ability. Furthermore, the incorporations between the two structures have vast applications. This approach comprises combining the FLC and NN into an incorporated system to acquire the merits of both of them (Lee and Teng, 2000; Hameed et al., 2019). For case, Hameed et al. (2019) suggested a common NN for a FLC which is tuning to control an unmanned vehicle. In the last decades, Petri net (PN) has been sophisticated into a robust tool for analysis, modelling, controlling, optimisation and implementation of different engineering systems (Ameer et al., 2018; David and Alla, 1994; Shen, 2003). In this paper, fuzzy neural Petri net (FNPN) has been utilised without identification for controlling a three links robot arm. The simulation of the robot arm application has been implemented in MATLAB software to presents the results.

2 A dynamic model of three links robot arm

The mathematical model of the arm robot manipulator consists of finding the mapping between the mutual positions and the forces applied to the structures, accelerations and velocities (Ishii et al., 1997). Two formularisations are mostly utilised to derive the dynamic model: namely the Newton-Euler formularisation and the Lagrange formularisation (Selvam et al., 2019; Shi and Zhang, 2020). A large number of authors and researchers (Schilling, 1998; Spong and Vidyasagar, 2001; Abdul Baqi, 2004), used Lagrange's approach to drive the general form of robot equation of motion. The Lagrange formularisations consequently take the alternative form (Wai et al., 2004):

$$I(\theta)\ddot{\theta} + \dot{I}\left(\theta, \dot{\theta}\right) - \frac{1}{2}\left[\frac{\partial\left(I\dot{\theta}\right)}{\partial\theta}\right]^T \dot{\theta} + \frac{\partial P}{\partial\theta} = \tau_n \tag{1}$$

where $\theta, \dot{\theta}$ and $\ddot{\theta} \in R^n$ indicate the vectors of joint link positions, velocities and acceleration respectively, $I(\theta) \in R^{n\times n}$ denotes the inertia matrix, n denote the number of links, P is the potential energy and τ_n denoted the torque of n link. In Figure 1, the robot manipulator is designed with links which their mass centres C_1, C_2 and C_3 are located at the midpoints of segments O_1O_2, O_2O_3 and O_3P, respectively. Furthermore, the i^{th} link has a mass (m_n) and a centroidal moment of inertia in a direction normal to the plane of motion (I_n); while the joints are moved by motors delivering torques τ_1, τ_2, and τ_3, the lubricant of the joints producing dissipative torques that we will neglect in this model. By the supposition that gravity acts in the direction of the Y-axis. In common, the mathematical model of the armature-controlled DC servo motors based on an n- link robot manipulator can be described as follows (Abdul Baqi, 2004):

$$\tau_e = K_T i_a \tag{2}$$

$$\tau_e = J_m \ddot{q}_m + B_m \dot{q}_m + \tau_m \tag{3}$$

$$v_t = R_a i_a + L_a \frac{di_a}{dt} + K_E \dot{q}_m \tag{4}$$

where $\tau_e \in R^n$ represents 'the vector of electromagnetic torque', $K_T \in R^{n\times n}$ represents 'the diagonal matrix of motor torque constants', $i_a \in R^n$ represents 'the vector of armature currents', $J_m \in R^{n\times n}$ represents 'the diagonal matrix of the moment inertia', $B_m \in R^{n\times n}$ represents 'the diagonal matrix of torsional damping coefficients', $q_m, \dot{q}_m$ and $\ddot{q}_m \in R^n$ indicate 'the vectors of motor shaft positions, velocities and accelerations, respectively', $\tau_m \in R^n$ represents 'the vector of load torque', $v_t \in R^n$ represents 'the vector of armature input voltages', $R_a \in R^{n\times n}$ represents 'the diagonal matrix of armature resistance', $L_a \in R^{n\times n}$ represents 'the diagonal matrix of armature inductance', and $K^E \in R^{n\times n}$ represents 'the diagonal matrix of the back electromotive force (EMF) coefficients'. In order to apply the DC servo motors for driving an n-link robot manipulator, a relationship between the motor-shaft position q_m and the joint position θ can be illustrated as (Abdul Baqi, 2004):

$$g_r = \frac{q_m}{\theta} = \frac{\tau}{\tau_m} \tag{5}$$

The governed equation of an n-link robot manipulator with actuator dynamics can be acquired as (Wai and Chen, 2006):

$$I^*(\theta)\dddot{\theta} + D(\theta, \dot{\theta}, \ddot{\theta}) + d = U \tag{6}$$

where $U \in R^n$ indicate the control effort vector, i.e., the input of armature voltages,

$$\begin{aligned} I(\theta) &= I_n + I_q(\theta) \\ I^* &= L_n[I_n + J_n] \end{aligned} \tag{7}$$

$$\begin{aligned} C(\theta, \dot{\theta}) &= \dot{I}(\theta, \dot{\theta}) - \frac{1}{2}\left[\frac{\partial(I\dot{\theta})}{\partial\theta}\right]^T \\ D(\theta, \dot{\theta}, \ddot{\theta}) &= \left\{L_n\left[\dot{I}(\theta, \dot{\theta}) + (\theta, \dot{\theta}) + B_n\right] + [I(\theta) + J_n]\right\}\ddot{\theta} \\ &+ \left[L_n\dot{C}(\theta, \dot{\theta}, \ddot{\theta})\dot{\theta} + R_nC(\theta, \dot{\theta})\dot{\theta} + R_nB_n\dot{\theta} + K_{En}\dot{\theta} + L_n\dot{G}(\theta, \dot{\theta}) + R_nG(\theta)\right] \end{aligned} \tag{8}$$

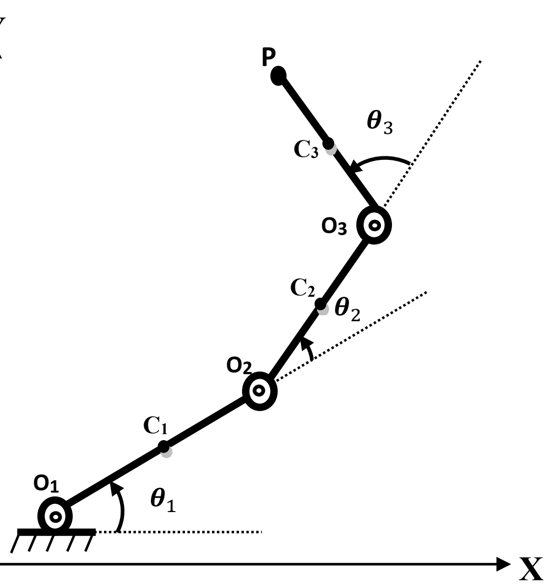
$$d = L_nI_q(\theta)\dddot{\theta} + L_n\dot{N} + R_nN \tag{9}$$

where $G(\theta)$ is gravity vector, N indicates the vector of external disturbance t_l and friction term $f(\dot{\theta})$. Then, we can rewrite equation (6) as:

$$\dddot{\theta} = I^*(\theta)^{-1}\left[U - \left(D(\theta, \dot{\theta}, \ddot{\theta}) + d\right)\right] \tag{10}$$

By using the method of numerical integration such Euler method for equation (10), we can get position, velocity and acceleration for each link.

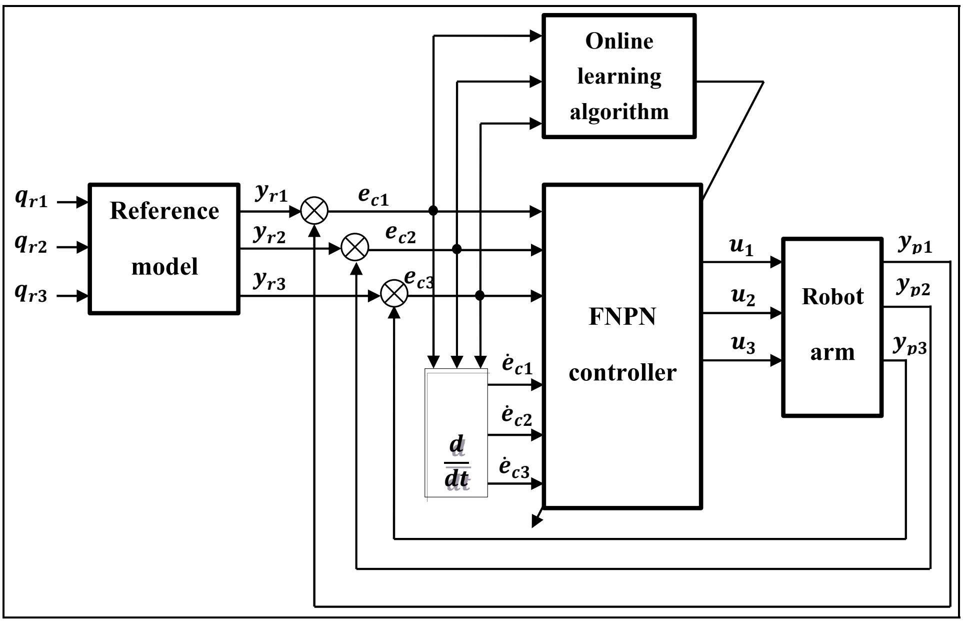
Figure 1 Three links robot arm



3 FNPN structure and control

To achieve the path (trajectory) tracking of a three links robot arm under various moving angles efficiently (Reza et al., 2019), the FNPN used to control the trajectory tracking of the arm robot, as depicts in Figure 2. The proposed controller structure with an online learning algorithm is utilised in this paper.

Figure 2 Block diagram of robot arm based on FNPN controller



3.1 FNPN structure

The essential idea of a PN is combined with a conventional FNN is utilised to established an FNPN system. The FNPN structure is shown in Figure 3. Where the major difference between the classical FNN and the FNPN is the transition layer (Shen, 2003; Wai and Liu, 2009). The basic function and the signal propagation in each layer of the FNPN are described as follows.

3.1.1 Input layer

For every node, j in the input layer transmits the input linguistic variables x_j ($j = 1, 2, \ldots, n$) to the next layer directly.

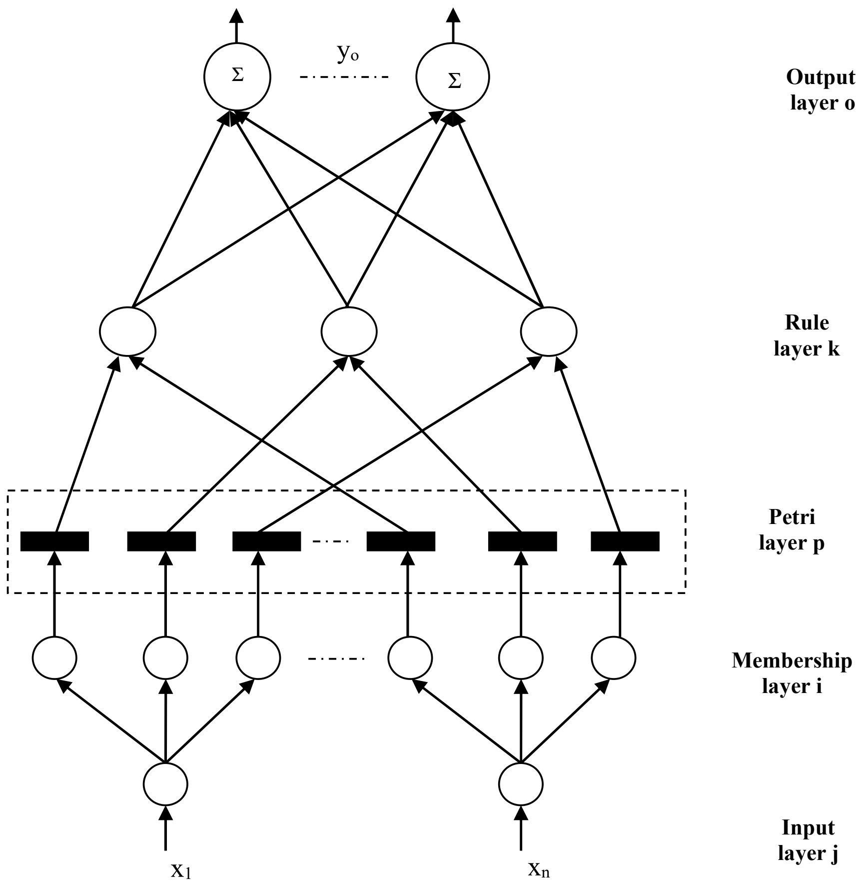
3.1.2 Membership layer

Each node in the membership layer carry out a membership function; the input of this layer can be represented by the output of the previous layer (input layer) which can be defined through Gaussian membership functions as follow (Angels, 2003):

$$net_i(x_j) = -\frac{1}{2}\frac{(x_j - m_{cij})^2}{s_{cij}^2}$$
$$mf_{ij}[net_i(x_j)] = \exp[net_i(x_j)] \tag{11}$$

where exp[.] is the exponential function, and m_{cij} and s_{cij} are, respectively, the mean and the standard deviation of the Gaussian function.

Figure 3 Architecture of FNPN



3.1.3 Petri layer

The Petri layer of the FNPN producing tokens makes utilise of competition laws to choose the convenient fired nodes as follows (Angels, 2003):

$$tr_{ij} = \begin{cases} 1, \text{if} \quad mf_{ij}[net_i(x_j)] \geq d_{th} \\ 0, \text{if} \quad mf_{ij}[net_i(x_j)] < d_{th} \end{cases} \tag{12}$$

where tr_{ij} is the transition and d_{th} is a dynamic threshold value diverse with the corresponding error to be introduced later. Can be seen that the dynamic threshold value in equation (12) is adjusted by the following equation (Angels, 2003):

$$d_{th} = \frac{\alpha \exp(-\beta E)}{1 + \exp(-\beta E)} \tag{13}$$

where α and β are positive constants. This means that the higher the error, the lower the threshold value. In other words, if the error becomes large, the threshold values will be reduced in order to fire more rules for the present situation (Angels, 2003).

3.1.4 Rule layer

Each node i in the rule layer is indicated by Π. which means that all the input signals are multiplied and the result represents the output. The output of the rule layer is presented as following (Angels, 2003):

$$\varnothing_{ci} = \begin{cases} \prod_{j}^{n} mf_{ij}\left[net_i\left(x_j\right)\right], & tr_{ij} = 1 \\ 0, & tr_{ij} = 0 \end{cases} \tag{14}$$

where $\varnothing_i$ represent the i_{th} (output of the rule layer).

3.1.5 Output layer

Each node y_o calculates the total output as the gathering of all input signals as (Angels, 2003):

$$y_o = \sum_{i}^{n_i} w_{coi} \varnothing_{ci} \tag{15}$$

where w_{coi} is weight connection between i_{th} (output of rule layer) and o_{th} (output layer).

3.2 Online learning algorithm

The main goal of the optimisation algorithms is learning the parameters or variables for any controller or network to get the optimal solution. In this paper, the optimisation algorithm is utilised to learning the parameters of FNPN controller and how to get a recurrent gradient vector, where the learning algorithm is working according to a fitness (error) function for a parameter of the network utilising the string rule. Due to calculate the gradient vector in the opposite direction to the flow of the output of each node. Generally, the algorithm is indicated as the back-propagation learning rule. To depict the online tuning algorithm of the FNPN utilising the supervised gradient-descent algorithm the fitness (error) function E_c defines as follow:

$$E_c = \frac{1}{2}\left(e_{c1}^2 + e_{c2}^2 + e_{c3}^2\right) \tag{16}$$

where e_{c1}, e_{c2} and e_{c3} are errors between reference outputs and robot's link1, link2 and link3 output respectively, then the gradient of the error E_c with regard to weights, the mean and standard deviation of the Gaussian function are given:

$$\frac{\partial E_c}{\partial w_{coi}} = \frac{\partial E_c}{\partial_{co}} \frac{\partial e_{co}}{\partial y_{po}} \frac{\partial y_{po}}{\partial u_o} \frac{\partial u_o}{\partial w_{coi}} = -e_{co} \frac{\partial y_{po}}{\partial u_o} \varnothing_{ci} \tag{17}$$

$$\frac{\partial E_c}{\partial m_{cij}} = -\sum\nolimits_{o=1}^{3} e_{co} \frac{\partial y_{po}}{\partial u_o} w_{coi} \varnothing_{ci} \frac{X_{cj} - m_{cij}}{s_{cij}^2} \tag{18}$$

$$\frac{\partial E_c}{\partial s_{cij}} = -\sum\nolimits_{o=1}^{3} e_{co} \frac{\partial y_{po}}{\partial u_o} w_{coi} \varnothing_{ci} \frac{\left(X_{cj} - m_{cij}\right)^2}{s_{cij}^3} \tag{19}$$

where $\frac{\partial y_{po}}{\partial u_o}$ is the sensitivity system, u_o and y_{po} are FNPN control output and robot arm output, respectively. The accurate estimation for the sensitivity of the system is difficult to determine due to the uncertain dynamics of the robot arm system. Identifiers can be performed to estimate the sensitivity of the system but it is required a hard computation effort. To vanquish this problem and raising the online learning speed, the sensitivity of the system can be approximated by their sign functions (Angels, 2003) as follow:

$$\frac{\partial y_{po}}{\partial u_o} \cong sign\left(\frac{y_{po}(k) - y_{po}(k-1)}{u_o(k) - u_o(k-1)} \right) \tag{20}$$

Equations (17) to (19) are substituted in equations (21) to (23)

$$w_{ci}(k+1) = w_{ci}(k) - \eta_{cw} \frac{\partial E_c}{\partial w_{ci}} \tag{21}$$

$$m_{cij}(k+1) = m_{cij}(k) - \eta_{cm} \frac{\partial E_c}{\partial m_{cij}} \tag{22}$$

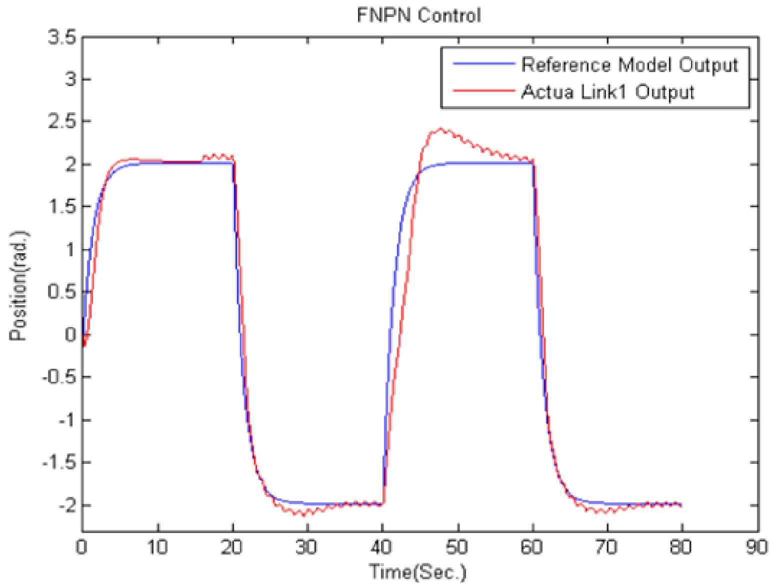
$$s_{cij}(k+1) = s_{cij}(k) - \eta_{cs} \frac{\partial E_c}{\partial s_{cij}} \tag{23}$$

where η_{cw} is learning rate for the weights, η_{cm}, η_{cs} are the learning rate for the mean and standard deviation of the Gaussian function, respectively.

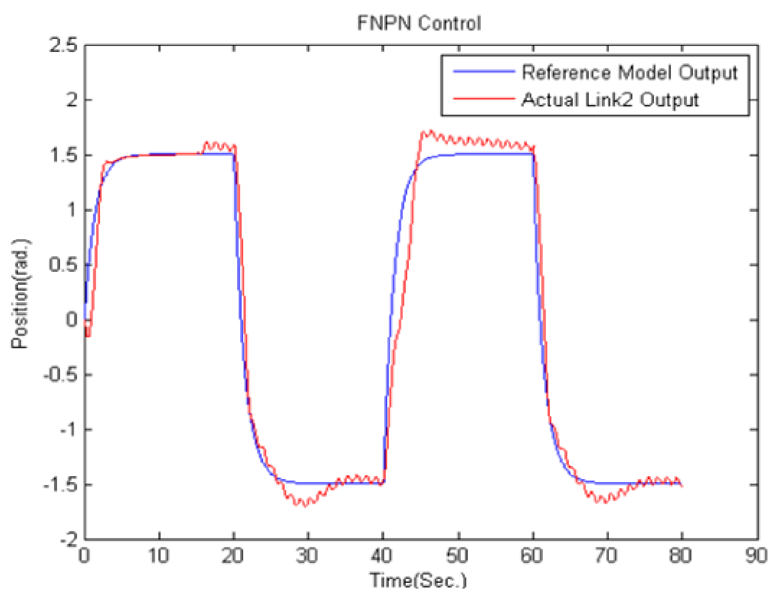
4 Simulation results

The Simulink model of the FNPN controller for three links robot arm has been implemented in MATLAB software program. The number of rules and outputs are 50 and 3, respectively. The initial mean and standard of membership function were computed as equations (24) and (25) (Shen, 2003), beside the 0.001 values for weights. However, the FNPN control without identification is implemented. The plant sensitivity calculated by equation (20), this reduces the time consuming to evaluate the parameters in offline forward or inverse identification also reduces the time required to calculate the plant sensitivity in forward control. The inputs to FNPN controller depended on the error between the reference outputs and robot arm outputs such as shown in Figure 2 and derivative of error, for this reason, six inputs fed to FNPN control. The learning rate of weights, mean and standard are $\eta_{cw} = 0.02$, $\eta_{cm} = 0.00012$ and $\eta_{cs} = 0.001$, respectively.

Figure 4 FNPN control for (a) link1, (b) link2 and (c) link3 for five epochs (see online version for colours)

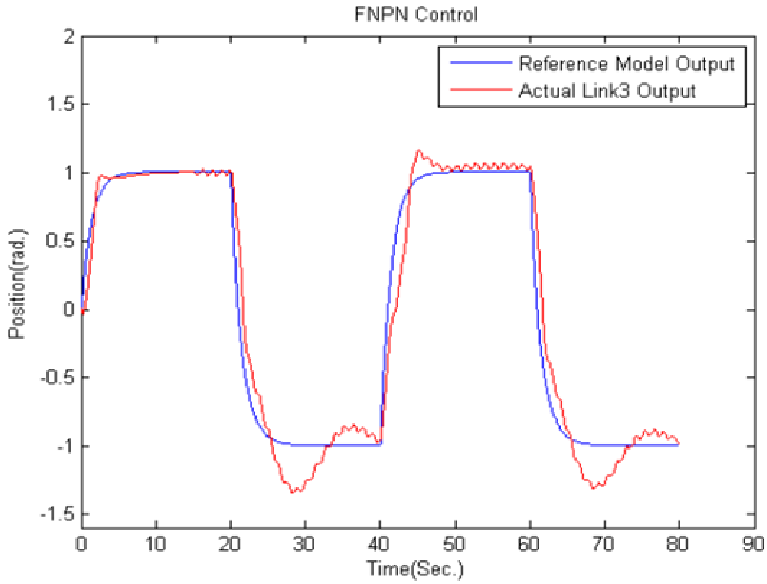


(a)



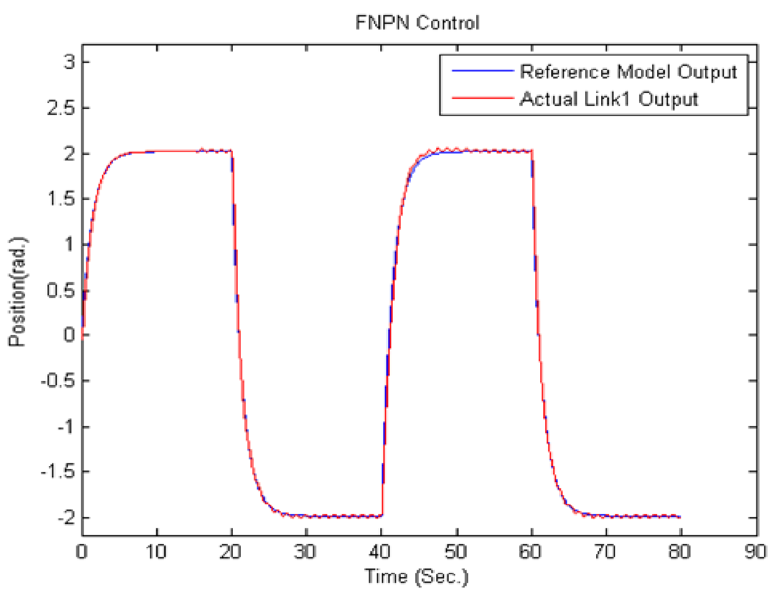
(b)

Figure 4 FNPN control for (a) link1, (b) link2 and (c) link3 for five epochs (continued) (see online version for colours)



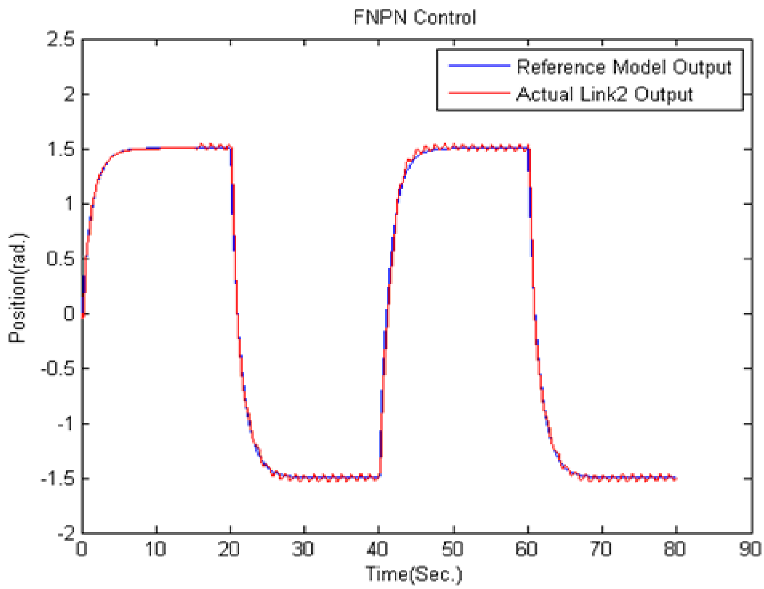
(c)

Figure 5 FNPN control simulation results of position response and MSE for link1, link2 and link3, (a) FNPN control for link1 for 100 epochs (b) FNPN control for link2 for 100 epochs (c) FNPN control for link3 for 100 epochs (d) MSE for link1 after 100 epochs (e) MSE for link2 after 100 epochs (f) MSE for link1 after 100 epochs (see online version for colours)

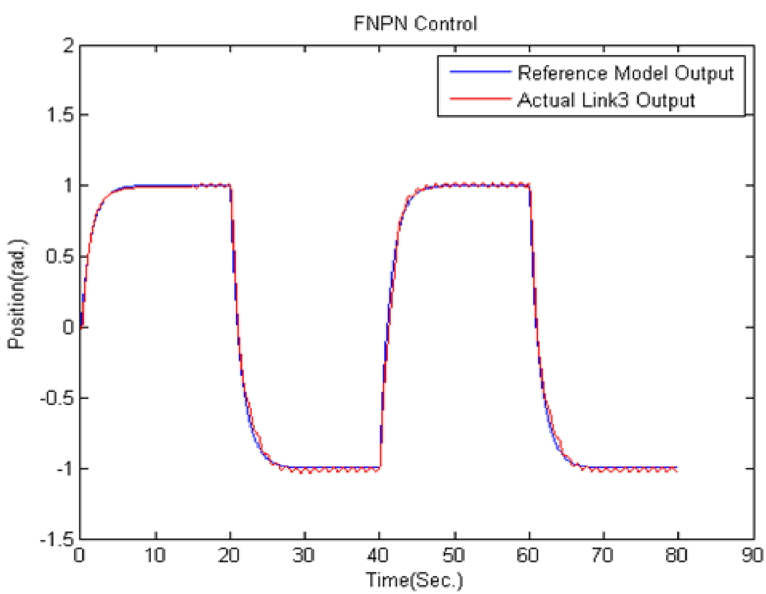


(a)

Figure 5 FNPN control simulation results of position response and MSE for link1, link2 and link3, (a) FNPN control for link1 for 100 epochs (b) FNPN control for link2 for 100 epochs (c) FNPN control for link3 for 100 epochs (d) MSE for link1 after 100 epochs (e) MSE for link2 after 100 epochs (f) MSE for link1 after 100 epochs (continued) (see online version for colours)

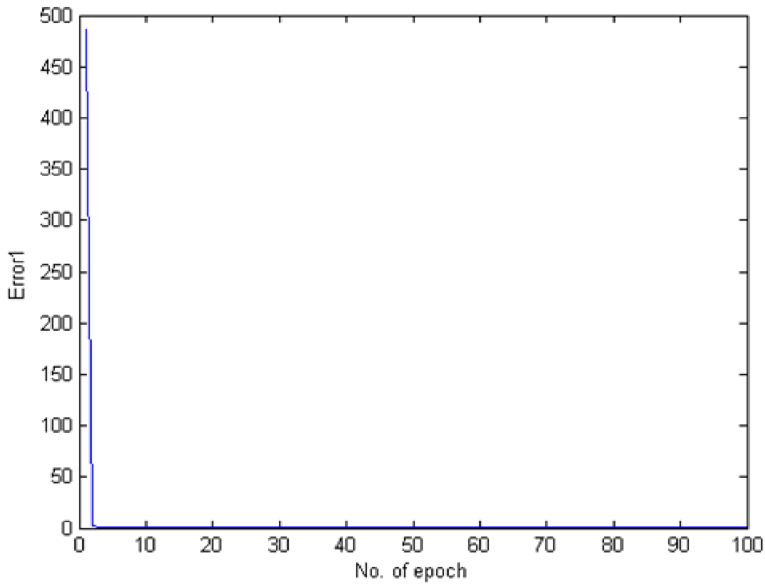


(b)

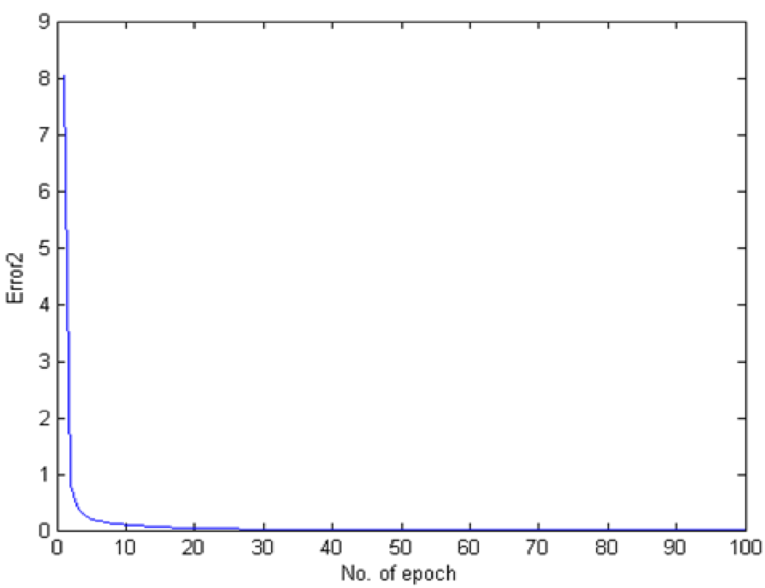


(c)

Figure 5 FNPN control simulation results of position response and MSE for link1, link2 and link3, (a) FNPN control for link1 for 100 epochs (b) FNPN control for link2 for 100 epochs (c) FNPN control for link3 for 100 epochs (d) MSE for link1 after 100 epochs (e) MSE for link2 after 100 epochs (f) MSE for link1 after 100 epochs (continued) (see online version for colours)

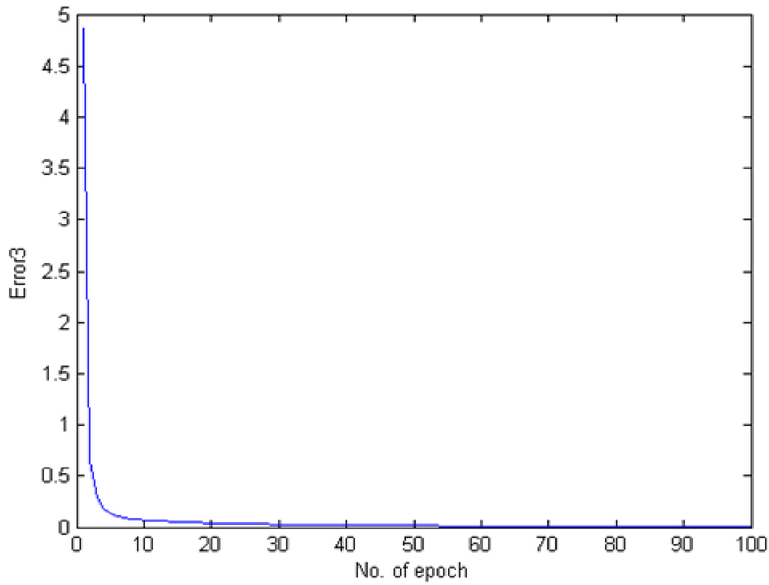


(d)



(e)

Figure 5 FNPN control simulation results of position response and MSE for link1, link2 and link3, (a) FNPN control for link1 for 100 epochs (b) FNPN control for link2 for 100 epochs (c) FNPN control for link3 for 100 epochs (d) MSE for link1 after 100 epochs (e) MSE for link2 after 100 epochs (f) MSE for link1 after 100 epochs (continued) (see online version for colours)



(f)

The robotic system is influenced by some of the important parameters that affect its control performance such as the friction term $f(\dot{\theta})$ the external disturbance $t_l(t)$ and the parameter changing of 3rd link's mass m_3. For the two simulation examples, there are three situations including:

- nominal situation ($m_3 = 1$ kg and $N = 0$) at starting
- parameter changing position takes place at $t = 15$ sec ($m_3 = 2$ kg)
- disturbance, besides, friction forces are also counted in this simulation
- sampling time $Ts = 0.01$ sec.

Hence,

$$t_l(t) = \left[5\sin(5t) \quad 3\sin(5t) \quad \sin(5t)\right]^T$$

$$f(\dot{\theta}) = \left[20\dot{\theta}_1 + .8sign(\dot{\theta}_1) \quad 10\dot{\theta}_2 + .4sign(\dot{\theta}_2) \quad 5\dot{\theta}_3 + .2sign(\dot{\theta}_3)\right]^T$$

$$N = t_l(t) + f(\dot{\theta})$$

$$m_{ij} = X_{n\max} - (i-1)\frac{X_{n\max} - X_{n\min}}{N_i - 1} \tag{24}$$

$$s_{ij} = 2\frac{X_{n\max} - X_{n\min}}{N_i - 1} \tag{25}$$

where $X_{n\max}$, $X_{n\min}$ are the pre-specified maximum and minimum bounds of n^{th} input to FNPN. Figures 4(a) to 4(c) show the earlier FNPN control after five epochs for three links while Figures 5(a) to 5(f) are shown the FNPN the control position response and mean square error (MSE) for link1, link2 and link3, respectively for 100 epochs. Table 1 shows the gradient MSE for the control simulation result.

Figure 4 show the earlier FNPN control after five epochs for three links.

Table 1 Mean square error

Link1 position	*Link2 position*	*Link3 position*
0.0078	0.0075	0.0063

5 Conclusions

In this paper, an FNPN controller utilised for controlling the three-links robot arm without identification. The supervised gradient-descent algorithm is utilised to learning the parameters of the FNPN controller to get an optimal response with minimum error. The online tuning algorithm is used according to the fitness function to get a recurrent gradient vector. This paper proved that the convergence of errors for control of three links using the FNPN controller is very good. For this reason, the state returned to the error is large at starting of control, this makes the threshold equal to 0 (d_{th} = 0) to fire more rules, when the error is reducing the threshold is increase. This reduces the firing rules to decrease the computation this is useful in real-time control and produces a fast response in the nonlinear control systems.

References

Abdul Baqi, J.N. (2004) *Neuro-fuzzy Control of Robot Arm*, MSc thesis, College of Engineering, University of Basrah.

Ameer, L.S., Mohammed, J.M., Ahmed, S.K., Hana'a, M.R. and Hesham, J.M. (2018) 'Design fuzzy neural Petri net controller for trajectory tracking control of mobile robot', *International Journal of Engineering & Technology*, Vol. 7, No. 4, pp.2256–2262.

Angels, J. (2003) *Fundamentals of Robotic Mechanical Systems: Theory, Methods and Algorithms*, Springer-Verlag New York, Inc.

Chang, Y.C. (2000) 'Neural network-based H-infinite tracking control for robotic systems', *Proc. Inst. Electr. Eng. – Control Theory Appl.*, Vol. 147, No. 3, pp.303–311.

Das, T., Kar, I.N. and Chaudhury, S. (2006) 'Simple neuron-based adaptive controller for a nonholonomic mobile robot including actuator dynamics', *Neurocomputing*, Vol. 69, Nos. 16–18, pp.2140–2151.

David, R. and Alla, H. (1994) 'Petri nets for modeling of dynamic systems: a survey', *Automatica*, Vol. 30, No. 2, pp.175–202.

Funahashi, K. and Nakamura, Y. (1993) 'Approximation of dynamical systems by continuous-time recurrent neural network', *Neural Networks*, Vol. 6, No. 6, pp.801–806.

Hameed, W.I., Saleh, A.L., Sawadi, B.A., Al-Yasir, Y.I.A. and Abd-Alhameed, R.A. (2019) 'Maximum power point tracking for photovoltaic system by using fuzzy neural network', *Inventions*, Vol. 4, No. 33, pp.1–12.

Huang, S.J. and Lee, J.S. (2000) 'A stable self-organizing fuzzy controller for robotic motion control', *IEEE Trans. Ind. Electron.*, Vol. 47, No. 2, pp.421–428.

Ishii, C., Shen, T. and Tamura, K. (1997) 'Robust model following control for a robot manipulator', *Proc. Inst. Electr. Eng. – Control Theory Appl.*, Vol. 144, No. 1, pp.53–60.

Jin, L., Nikiforuk, P.N. and Gupta, M. (1995) 'Approximation of discrete-time state-space trajectories using dynamic recurrent neural networks', *IEEE Trans. Automat. Contr.*, Vol. 40, No. 7, pp.1266–1270.

Kim, Y.H. and Lewis, F.L. (2000) 'Optimal design of CMAC neural-network controller for robot manipulators', *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, Vol. 30, No. 1, pp.22–31.

Ku, C.C. and Lee, K.Y. (1995) 'Diagonal recurrent neural networks for dynamic systems control', *IEEE Trans. Neural Networks*, Vol. 6, No. 1, pp.144–156.

Lee, C-H. and Teng, C-C. (2000) 'Identification and control of dynamic systems using recurrent fuzzy neural networks', *IEEE Trans. Fuzzy System*, Vol. 8, No. 4, pp.349–366.

Omidvar, O. and Elliott, D.L. (1997) *Neural Systems for Control*, Academic, New York.

Reza, R.A., Hoda, N.K. and Atikeh, R. (2019) 'Design and simulation of self-tuning fractional order fuzzy PID controller for robotic manipulator', *International Journal of Automation and Control (IJAAC)*, Vol. 13, No. 5, pp.595–618.

Schilling, R.J. (1998) *Fundamentals of Robotics: Analysis and Control*, Prentice-Hall, Hoboken, NJ.

Selvam, B., Natarajan, U., Balasubramonian, M. and Lakshmanaprabu, S.K. (2019) 'Trajectory tracking control of two-link industrial robot manipulator based on C++', *Int. J. Rapid Manufacturing*, Vol. 8, Nos. 1/2, pp.3–15.

Shen, V.R.L. (2003) 'Reinforcement learning for high-level fuzzy Petri nets', *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, Vol. 33, No. 2, pp.351–362.

Shi, Q. and Zhang, J. (2020) 'Research on manipulator motion control based on neural network algorithms', *Int. J. Biometrics*, Vol. 12, No. 1, pp.81–90.

Spong, M.W. and Vidyasagar, S.H.M. (2001) *Robot Modeling and Control*, John Wiley and Sons, Inc., New York.

Wai, R-j. and Chen, P-C. (2006) 'Robust neural-fuzzy-network control for robot manipulator including actuator dynamics', *IEEE Trans. Indst. Elect.*, Vol. 53, No. 4, pp.1328–1349.

Wai, R-J. and Liu, C-M. (2009) 'Design of dynamic Petri recurrent fuzzy neural network and its application to path-tracking control of nonholonomic mobile robot', *IEEE transactions on Ind. Elec.*, Vol. 56, No. 7, pp.2667–2683.

Wai, R-J., Chen, P.C. and Tu, C-Y. (2004) 'Robust neural-fuzzy-network control for rigid-link electrically driven robot manipulator', *IEEE Trans. Ind. Electron., 30th Annual Conference*, pp.1328–1349.

Wang, L.X. (1997) *A Course in Fuzzy Systems and Control*, Prentice-Hall, Englewood Cliffs, NJ.

Yang, S.X. and Meng, M.Q.H. (2003) 'Real-time collision-free motion planning of a mobile robot using a neural dynamics-based approach', *IEEE Trans. Neural Netw.*, Vol. 14, No. 6, pp.1541–1552.

Yoo, B.K. and Ham, W.C. (2000) 'Adaptive control of robot manipulator using fuzzy compensator', *IEEE Trans. Fuzzy Syst.*, Vol. 8, No. 2, pp.186–199.