Development of a real-time-position prediction algorithm for under-actuated robot manipulator by using of artificial neural network

H M A A Al-Assadi[1]*, **A A Mat Isa**[1], **Ali T Hasan**[2], **Z A Rahman**[1], and **B Heimann**[3]

[1]Faculty of Mechanical Engineering, Universiti Teknologi MARA (UiTM), Selangor, Malaysia

[2]Ministry of Higher Education and Scientific Research, Baghdad, Iraq

[3]Institute of Robotics, Hannover Center of Mechatronics, Hannover University, Hannover, Germany

The manuscript was received on 29 May 2010 and was accepted after revision for publication on 24 January 2011.

DOI: 10.1177/0954406211400345

Abstract: An adaptive learning algorithm using an artificial neural network (ANN) has been proposed to predict the passive joint position of under-actuated robot manipulator. In this approach, a specific ANN model has been designed and trained to learn a desired set of joint angular positions for the passive joint from a given set of input torque and angular position for the active joint over a certain period of time. Trying to overcome the disadvantages of many used techniques in the literature, the ANNs have a significant advantage of being a model-free method. The learning algorithm can directly determine the position of its passive joint, and can, therefore, completely eliminate the need for any system modelling. Even though it is very difficult in practice, data used in this study were recorded experimentally from sensors fixed on robot's joints to overcome the effect of kinematics uncertainties present in the real world such as ill-defined linkage parameters and backlashes in gear trains. An ANN was trained using the experimentally obtained data and then used to predict the path of the passive joint that is positioned by the dynamic coupling of the active joint. The generality and efficiency of the proposed algorithm are demonstrated through simulations of an under-actuated robot manipulator; finally, the obtained results were successfully verified experimentally.

Keywords: under-actuated robot, artificial neural network, prediction algorithm

1 INTRODUCTION

Robot manipulators, in general, are required to have the same number of actuators as the number of joints to obtain full control. In the case of under-actuated robots, this condition is not satisfied, which makes the behaviour of that class of robots very difficult to be predicted. Under-actuated robots can be a better design choice for robots in space and other industrial applications, and their advantages over fully actuated robots led to many studies to predict their behaviour [**1–15**]. As a first advantage, a light-weight and low power consumption manipulator can be made. This feature is required in low-cost automation and space robots. Second, they can easily overcome actuator failure due to unexpected accident. The under-actuated manipulator could be the model of the direct drive manipulator that has some failed joints; such fault-tolerant behaviour is highly desirable for robots in remote or hazardous environments [**1**]. Other interesting applications include the Acrobot [**2**, **3**], the gymnast robots [**4**], the brachiating robots [**5**], and surgical robots [**6**].

The mathematical complexity and wide variety of applications have kept under-actuated robots an area

**Corresponding author: Faculty of Mechanical Engineering, Universiti Teknologi MARA (UiTM), 40450 Shah Alam, Selangor, Malaysia.*

emails: ali.hayder@gmail; hayder@salam.uitm.edu.my

of open research. Luca *et al.* [**7**, **8**] have investigated the behaviour of a 2 R manipulator moving in a horizontal plane with a single actuator at the first joint using a mathematical model, neglecting joint friction which is not easy to achieve in real world as it involves high manufacturing cost. As it is well-known, using mathematical model, any parameter can be neglected, in this case the friction in the joints, which is a major drawback using the mathematical model approach. Trying to overcome that problem, some researchers have implemented additional equipments such as breaks at the passive joint [**9–12**]. In this case, the brake can generate torque; that means, after all, that kind of system is considered some kind of actuator. So, it will be difficult to consider that robot as an under-actuated manipulator.

Motivated by this problem, Yu *et al.* [**1**] have investigated the dynamic characteristics of a two-link manipulator in view of global motion including joint friction by proposing a mathematical model; they have found that the manipulator can be positioned if the friction acts on the passive joint. In this case, any additional equipment such as brakes is not needed in positioning all the joints to desired position. (A mathematical model showing the ability of friction acting to fix passive joints is discussed in detail in section 2 in the manuscript). Their results were verified using numerical simulation. The main purpose of this article is the use of an ANN to learn the system characteristics rather than to explicit a system model. The addition of the mathematical model was just to show the complexity of the model rather than to use it in the solution provided in this article.

Later on, Mahindrakar *et al.* [**13**] have presented a mathematical model for a two-link under-actuated manipulator wherein the motion of the system was confined to a horizontal plane; their proposed dynamic model takes into account the frictional forces acting on the joints. Results obtained were also verified through numerical simulation.

Many attempts to solve the problem have been found in the literature. Yet, solutions proposed still lack generality and systematization. To overcome this problem, artificial intelligence (AI) was introduced for predicting and making robot systems able to attribute more intelligence and high degree of autonomy.

Applying fuzzy logic to under-actuated robots (as an AI method), there were few studies in recent past [**14**, **15**]. Although the results presented were promising, several drawbacks are also found. First these results cannot be generalized to other systems, because they only came from practical considerations. Second, despite the fact that unlike most learning control algorithms, multiple trials are not necessary for the robot to learn the desired trajectory. Additional major drawback was that fuzzy logic-based approaches only remember the most recent data points introduced [**16**]. Gleaning the learning abilities of genetic algorithms GA (as another method of AI) to solve the problem was an alternative. Blending of GA with fuzzy rules in order to capture the hidden non-linearities of the system will be useful in developing any learning techniques. Lee and Zak [**17**] have presented the design criterion of a GA-based neural fuzzy controller for an anti-break system.

As it has been seen, each of the previously mentioned techniques has its own drawbacks. To overcome this problem, researchers have recommended neural networks so that they would remember the trajectories as they traverse them [**16**].

Artificial neural networks (ANNs) have been widely used for their extreme flexibility due to their learning ability and the capability to make non-linear function approximation. Their ability to learn by example makes them very flexible and powerful. ANNs, while implemented on computers, are not programmed to perform specific tasks. Instead, they are trained with respect to data sets until they learn the patterns presented to them. Once they are trained, new patterns may be presented to them for prediction or classification [**18**, **19**]. Therefore, ANNs have been intensively used for solving regression and classification problems in many fields. A number of realistic approaches have been proposed and justified for applications to robotic systems [**20–25**].

In real world application, no physical property such as the friction coefficient can be exactly derived. Besides, there are always kinematics uncertainties present in the real world such as ill-defined linkage parameters and backlashes in gear trains [**26**, **27**]. In this article, and to overcome whichever uncertainty presented in the real world, data were recorded experimentally from sensors fixed on each joint for a horizontal two-link under-actuated robot.

The implementation of ANN, as the prediction algorithm, is established on learning the target parameters based on weight adaptation by minimizing the tracking error after each iteration process. This scheme does not require any prior knowledge of the dynamic model of the system. The basic idea of this concept is the use of the ANNs to learn the characteristics of the robot system rather than to specify an explicit robot system model, so that every uncertainty in the system will be counted for. Experimental trajectory tracking has shown the ability of the proposed approach to overcome the disadvantages of using some schemes, like the fuzzy learning for example, that only remembers the most recent data sets introduced, as the literature has shown.

2 EQUATIONS OF MOTION WITH FRICTION EFFECT

As Fig. 1 show, the space coordinate of the manipulator is parameterized by q. The coordinate q_i, $i = 1, 2$ are the joint angles. The Euler–Lagrange equation of motion is [**13**]

$$M(q)\ddot{q} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \tau, \tag{1}$$

where $\dot{q}$ and $\ddot{q}$ are the generalized velocities and accelerations, respectively. $M(q)$ is the inertia matrix, which is symmetric and positive definite. The centripetal and Coriolis terms are collected in the vector $\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}})$. The vector $\boldsymbol{h}$ contains terms purely quadratic in the velocities; gravity terms are absent since it assumed that the manipulator moves in a horizontal plane.

Define the following constants

$$c_1 = m_1 r_1^2 + m_2 l_1^2 + I_1, \quad c_2 = m_2 r_2^2 + I_2, \quad c_3 = m_2 l_1 r_2$$

The equations of motion accounting for the Coulomb plus viscous friction at the joints become

$$m_{11}\ddot{q}_1 + m_{12}\ddot{q}_2 + h_1 = \tau - SGN(\dot{q}_1)F_1 - b_1\dot{q}_1 \tag{2}$$

$$m_{21}\ddot{q}_1 + m_{22}\ddot{q}_2 + h_2 = -SGN(\dot{q}_2)F_2 - b_2\dot{q}_2 \tag{3}$$

where

$$m_{11} = c_1 + c_2 + 2c_3 \cos q_2, \quad m_{12} = c_2 + c_3 \cos q_2$$
$$m_{21} = m_{12}, \quad m_{22} = c_2$$
$$h_1 = -c_3(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2)\sin q_2, \quad h_2 = c_3\dot{q}_1^2 \sin q_2$$

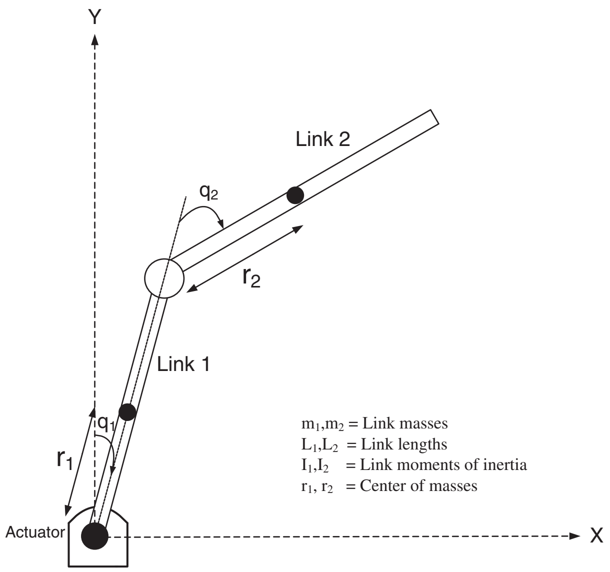

Fig. 1 Schematic diagram of the robot used

The F_i, $b_i\dot{q}_i$, $i = 1, 2$ represent the Coulomb and viscous friction forces, respectively; more details on joint friction could be found in Mahindrakar [**13**]. The set-valued signum function $sign(\cdot)$ is defined as:

$$SGN(x) \begin{cases} \{1\} & \text{if } x > 0, \\ \{-1\} & \text{if } x < 0, \\ [-1, 1] & \text{if } x = 0. \end{cases} \tag{4}$$

The above shown functions (1 till 4) suffer from the fact that the solution does not give a clear indication on how to select an appropriate solution from the several possible solutions for a particular arm configuration. However, the dynamic model with friction was not used or involved in this study attributed to the collection of real data from fabricated model. The real model is definite and naturally operates with the effect of friction and it is incorporated in the experimentally data collected, which are provided to the ANN to learn.

3 EXPERIMENT PROCEDURE

In this section, the real-time implementation of the experimentally collecting data procedure is discussed. A 2 R under-actuated horizontal robot was fabricated, as shown in Fig. 2. The robot arms were made of an aluminium square section beam to ensure a resisting to bending lightweight arm. Lengths of arms are $l_1 = 40$ cm and $l_2 = 35$ cm, respectively. The robot consists of base, actuated joint, two links, passive joint, and two encoders. The base is carrying the robot arm. The used actuator is a DC motor connected to the first link through a gearbox with a reduction ratio of 100 : 1, while the second joint is passive. Each of the joints has an encoder attached to it, in order to measure the rotation angle. The motor torque was calculated from the

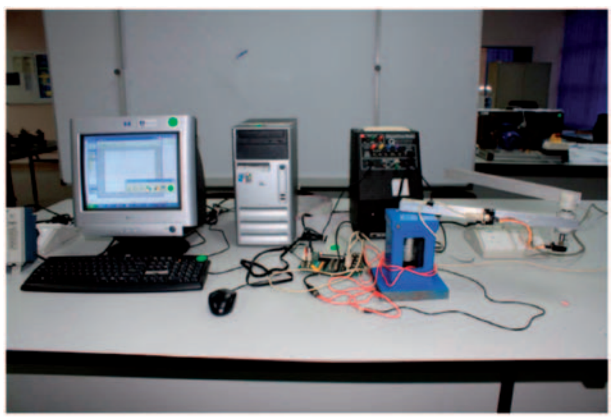

Fig. 2 The robot system used showing the computer, the data acquisition card, and the robot arms

voltage–torque relationship graph provided with the motor. Practically, it is a linear proportional relationship that allowed the torque to match the generated signals.

Different methods for collecting data have been found in the literature. Using a prespecified model, using a trajectory planning method, and using a simulation program for this purpose are examples for some of these methods. However, there are always kinematics uncertainties present in the real world such as ill-defined linkage parameters, link flexibility, and backlashes in gear train. In this approach, data were recorded directly from sensors fixed on each joint, so that every uncertainty in the dynamics of the system will be counted for.

The process of operating the manipulator and collecting of data were computerized, as shown in Fig. 2. A personal computer (PC) operating a MATLAB/SIMULINK software package attached with QUANSER data acquisition card (DAS) is used. The DAS is extracting the signals generated by the SIMULINK model via a digital to analogue convertor (DAC). Then, the signals are manipulated to the desired value using an amplifier device. At the same time, the DAS would read the encoder signals digitally, which could be read and stored in the SIMULINK model.

A square wave excitation signal was applied to the actuator causing different torque to the joints and the dynamic coupling effect was moving the passive joint correspondingly. As a standard signal generated by the MATLAB/SIMULINK, the square wave excitation signal, seen in Fig. 3, was chosen in order to cause a robot motion that covers the whole working cell rather than being a specified signal to perform a predefined trajectory.

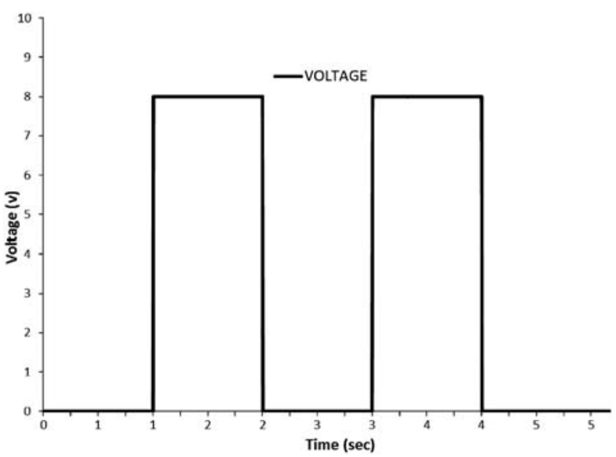

Fig. 3 The square wave excitation signal applied

4 THE ADAPTIVE LEARNING ALGORITHM

The fundamental idea underlying the design of the network is that the information entering the input layer is mapped as an internal representation in the units of the hidden layer and the outputs are generated by this internal representation rather than by the input vector. Given that there are enough hidden neurons, input vectors can always be encoded in a form so that the appropriate output vector can be generated from any input vector.

Figure 4 shows the developed ANN model. The outputs of the units in input layer are multiplied by appropriate weights W_{ij} and these are fed as inputs to the hidden layer. Hence, if O_i are the output of units in input layer, then the total input to the hidden layer is

$$sum_B = \sum_i O_i W_{ij} \tag{5}$$

Also, the output O_j of a unit in hidden layer is

$$O_j = f(sum_B) \tag{6}$$

where f is a non-linear activation function, it is a common practice to choose the sigmoid function given by

$$f(O_j) = \frac{1}{1 + e^{-O_j}} \tag{7}$$

as a non-linear activation function.

However, any input–output function that possesses a bounded derivative can be used in place of the sigmoid function.

If there is a fixed, finite set of input–output pairs, the total error in the performance of the network with a particular set of weights can be computed by comparing the actual and the desired output vectors for each presentation of an input vector. The error at any output unit e_K in the output layer can be calculated by

$$e_K = d_K - O_K \tag{8}$$

where d_K is the desired output for that unit in output layer and O_K the actual output produced by the network. The total error (E) at the output can be calculated by

$$E = \frac{1}{2} \sum_K (d_K - O_K)^2 \tag{9}$$

Learning comprises changing weights so as to minimize the error function (E) by the gradient descent method. It is necessary to compute the partial

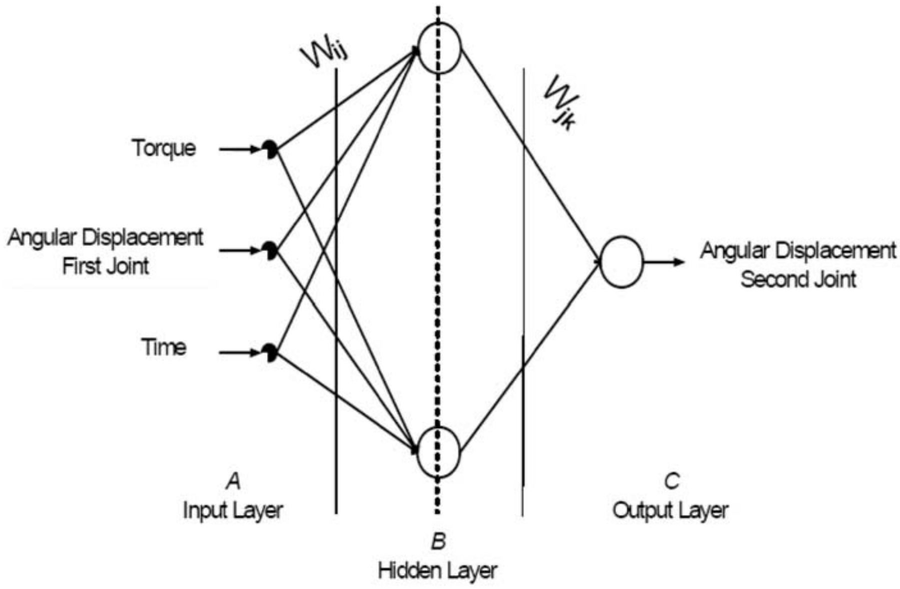

Fig. 4 The topology of the ANN model

derivative of (E) with respect to each weight in the network. Equations (5) and (6) describe the forward pass through the network where units in each layer have their states determined by the inputs they received from units of lower layer.

The backward pass through the network that involves 'back propagation' of weight error derivatives from the output layer back to the input layer is more complicated. For the sigmoid activation function given in equation (7), the so-called delta rule for iterative convergence towards a solution, stated in general is given as

$$\Delta W_{JK} = \eta \delta_K O_J \tag{10}$$

where η is the learning rate parameter, and the error δ_K at an output layer unit K is given by

$$\delta_K = O_K(1 - O_K)(d_K - O_K) \tag{11}$$

Also, the error δ_J at a hidden layer unit is given by

$$\delta_J = O_J(1 - O_J) \sum_K \delta_K W_{JK} \tag{12}$$

Using the generalized delta rule to adjust weights leading to the hidden units is back propagating the error adjustment, which allows for adjustment of weights leading to the hidden layer neurons in addition to the usual adjustments to the weights leading to the output layer neurons.

A back propagation network trains with a two-step procedure, the activity from the input pattern flows forward through the network, and the error signal flows backwards to adjust the weights using the following equations

$$W_{IJ} = W_{IJ} + \eta \delta_J O_I \tag{13}$$

$$W_{JK} = W_{JK} + \eta \delta_K O_J \tag{14}$$

until for each input vector the output vector produced by the network is the same as (or sufficiently close to) the desired output vector [**18–19**]. The number of hidden neurons and the learning factor are determined by trial and error [**28**]. Figure 5 graphically shows the design steps of the network.

5 RESULTS AND DISCUSSION

The 2 R manipulator equipped with one active joint, which is the main source of motion. Consequently, when the excitation signal is given, it would cause angular motion of the active joint and the corresponding response of the passive joint. The attached encoders collected both angular positions, as shown in Fig. 6. These collected data will be used as the target vector to be captured by the ANN model during the training process.

A supervised feed-forward ANN was designed using C programming language to learn the system behaviour over its workspace. The network consists of input, output, and one hidden layer. The input vector for the network consists of the angular displacement, the torque applied at the active joint (first joint), and the time interval, while the output vector was the angular position of the passive joint (second joint). As seen in Fig. 4, every neuron in the

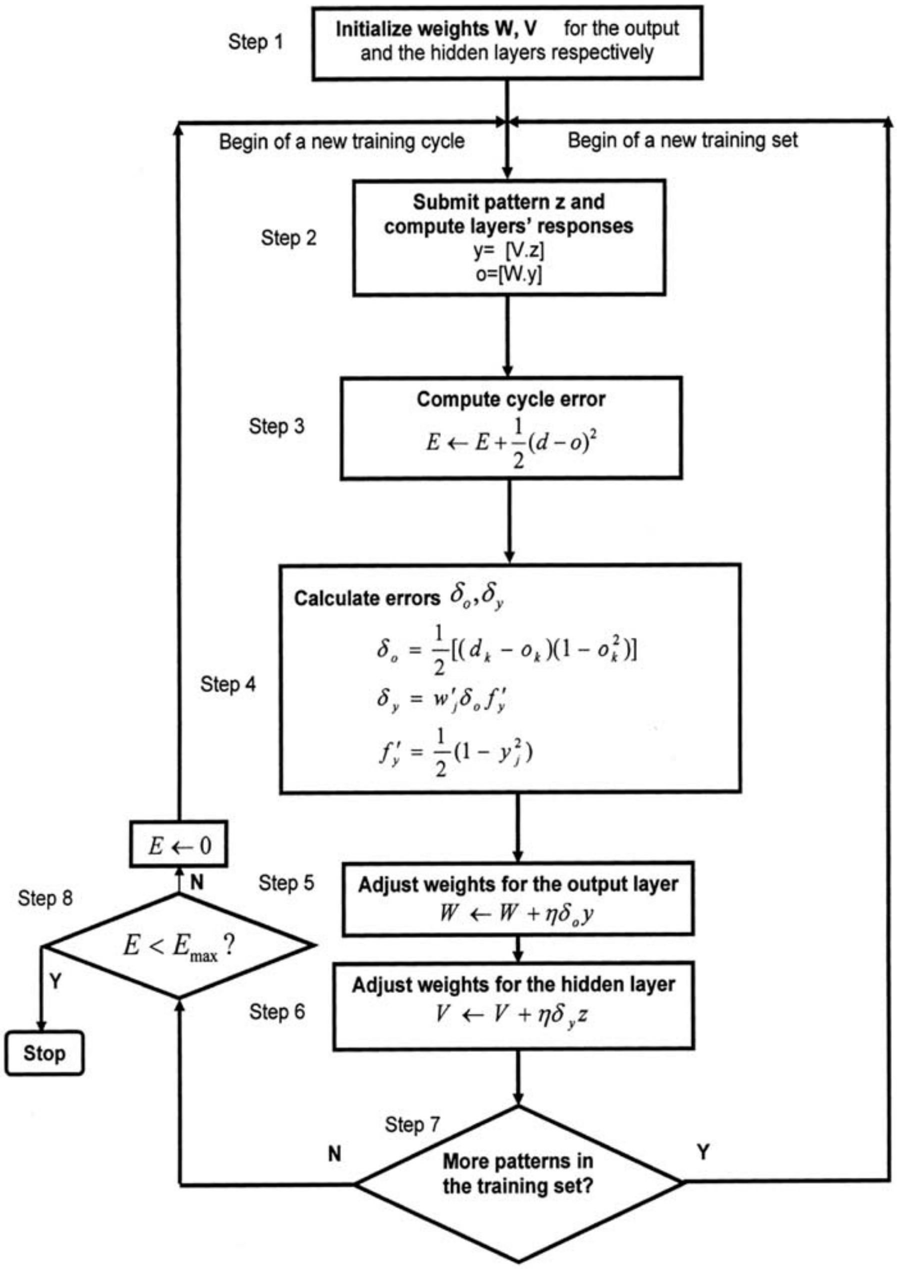

Fig. 5 The design steps of the network [**28**]

network is fully connected with each other, sigmoid transfer function was used to be the activation function, and generalized backpropagation delta learning rule (GDR) algorithm was used in the training process. All control data sets values had been scaled individually so that the overall difference in the dataset was maximized.

Training data were divided into 75 input–output sets that cover the entire work cell of the manipulator. To learn the target parameters, a training process was carried out using the experimentally obtained data. The network was trained by presenting several target points that the network had to learn. The number of neurons in the hidden layer was set to 25 with a constant learning factor of 0.9 by trial and error. Figure 7 shows the building knowledge process for the system.

To verify the success of the algorithm, the predicted values of the passive joint were compared to the experimentally collected data. The average absolute error was 4.9 per cent after 100 000 Iterations. Figure 8 graphically shows the trajectory tracking of the passive joint, and the results obtained show that the designed network is capable of learning

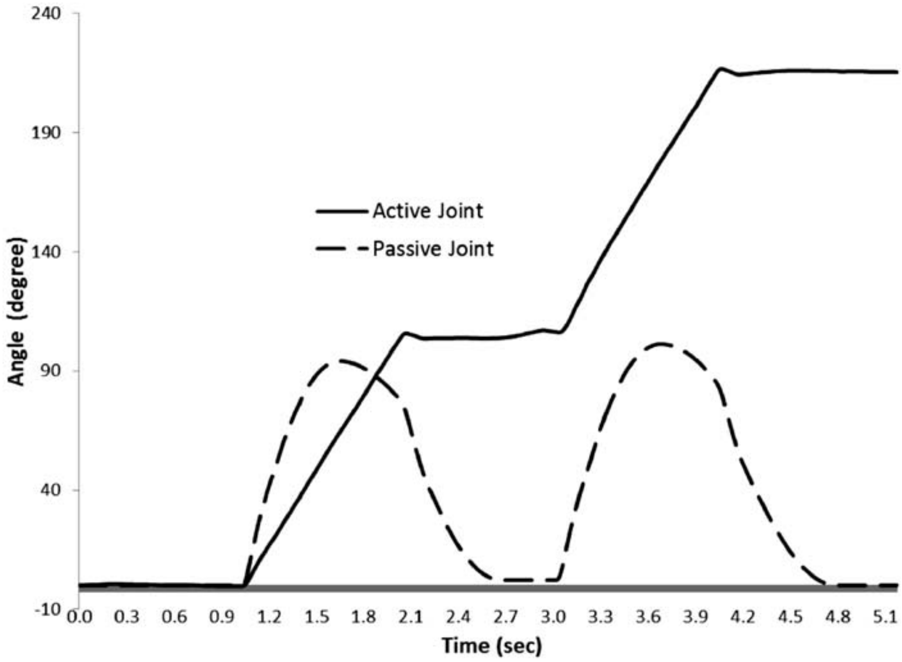

Fig. 6 Corresponding trajectory of the active and passive joints

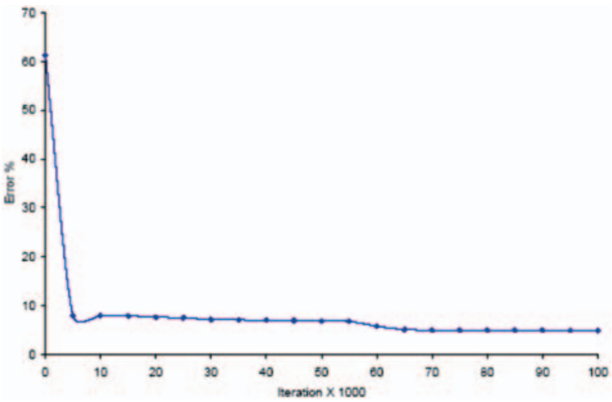

Fig. 7 Building knowledge of the system

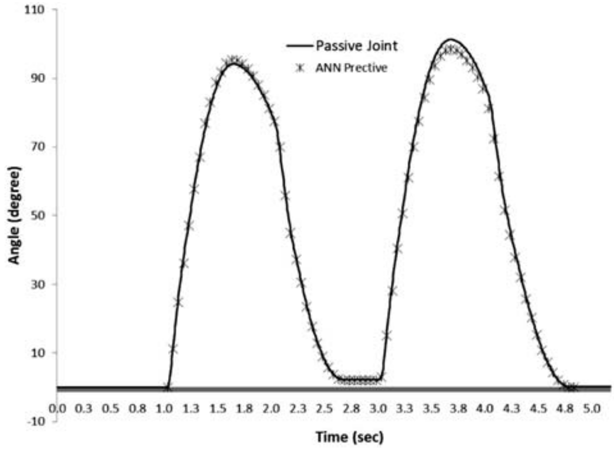

Fig. 8 Predicted trajectory tracking of the passive joint

and predicting the position of the passive joint successfully.

6 CONCLUSIONS

ANN technique was applied to the problem of positioning an under-actuated robot manipulator. The positioning of the passive joint of 2 R under-actuated manipulator was achieved by learning through training an ANN based only on observation of the input–output relationship. The proposed technique does not require any prior knowledge of the target system. The basic idea of this concept is the use of the ANN to learn and predict the behaviour of the robot system rather than to specify explicit robot system model, which is a significant advantage of using neural network approach.

A generated signal is used to operate the actuator. Results obtained have shown the ability of the network to predict the trajectory of the passive joint, which is positioned by the dynamic coupling of the active joint, overcoming the disadvantages of using some schemes, like the fuzzy learning for example, that only remember the most recent data sets introduced.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support of the Ministry of Higher Education, Malaysia under the 600-RMI/FRGS-09/2008 research grant scheme and also the Lab of Dynamic, Vibration and Control at the Faculty of Mechanical Engineering – Univeristi Teknolgi MARA for providing the facilities used in this project.

REFERENCES

1 **Yu, K.-H., Shito, Y.,** and **Inooka, H.** Position control of an underactuated manipulator using joint friction. *Int. J. Non-Linear Mech.*, 1998, **33**(4), 607–614.
2 **Berkemeier, M. D.** and **Fearing, R. S.** Tracking fast inverted trajectories of the underactuated acrobot. *IEEE Trans. Rob. Autom.*, 1999, **15**(4), 740–750.
3 **Spong, M. W.** The swing up control problem for the acrobat. *IEEE Control Syst. Mag.*, 1995, **15**(1), 49–55.
4 **Ono, K., Yamamoto, K.,** and **Imadu, A.** Control of giant swing motion of a two-link horizontal bar gymnastic robot. *Adv. Rob.*, 2001, **15**(4), 449–465.
5 **Nakanishi, J., Fukuda, T.,** and **Koditschek, D.** A brachiating robot controller. *IEEE Trans. Rob. Autom.*, 2000, **16**(2), 109–123.
6 **Funda, J., Taylor, R., Eldridge, B., Gomory, S.,** and **Gruben, K.** Constrained Cartesian motion control for teleoperated surgical robots. *IEEE Trans. Rob. Autom.*, 1996, **12**(3), 453–465.
7 **Luca, A. D., Mattone, R.,** and **Oriolo, G.** Stabilization of an underactuated planar 2R manipulator. *Int. J. Robust Nonlinear Control*, 2000, **24**, 181–198.
8 **Luca, A. D.** and **Oriolo, G.** Trajectory planning and control for planar robots with passive last joint. *Int. J. Rob. Res.*, 2002, **21**, 575–590.
9 **Arai, H.** and **Tachi, S.** Position control of a manipulator with passive joints using dynamic coupling. *IEEE Trans. Rob. Autom.*, 1991, **7**(4), 528–534.
10 **Mukherjee, R.** and **Chen, D.** Control of free-flying underactuated space manipulators to equilibrium manifolds. *IEEE Trans. Rob. Autom.*, 1993, **9**(5), 561–570.
11 **Yu, K.-H., Takahashi, T.,** and **Inooka, H.** Dynamics and motion control of a two-link robot manipulator with a passive joint. *Proc. 1995 IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, 1995, **2**, 311–316.
12 **Bergerman, M., Lee, C.,** and **Xu, Y.** Experimental study of an underactuated manipulator. *Proc. 1995 IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, 1995, **2**, 317–322.
13 **Mahindrakar, A. D., Rao, S.,** and **Banavar, R. N.** Point-to point control of a 2R planar horizontal underactuated manipulator. *J. Mech. Mach. Theory*, 2006, **41**, 838–844.
14 **Muscato, G.** Fuzzy control of an underactuated robot with a fuzzy microcontroller. *Microprocess. Microsyst.*, 1999, **23**, 385–391.
15 **Begovich, O., Sanchez, E. N.,** and **Maldonado, M.** Takagi–Sugeno fuzzy scheme for real-time trajectory tracking of an under-actuated robot. *IEEE Trans. Control Syst. Technol.*, 2002, **10**(1), 14–20.
16 **Graca, R. A.** and **Gu, Y.** A fuzzy learning algorithm for kinematics control of a robotic system. In Proceeding of the 32nd Conference on *Decision and Control*, San Antonio, Texas, 15–17 December 1993, pp. 1274–1279.
17 **Lee, Y.** and **Zak, S. H.** Designing a genetic neural antilock-break system controller. *IEEE Trans. Evol. Comput.*, 2002, **6**(2), 198–211.
18 **Soteris, A. K.** Artificial neural networks in renewable energy systems applications: a review. *Renewable Sustainable Energy Rev.*, 2001, **5**, 373–401.
19 **Hasan, A. T., Hamouda, A. M. S., Ismail, N.,** and **Al-Assadi, H. M. A. A.** An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 D.O.F serial robot manipulator. *J. Adv. Eng. Software*, 2006, **37**(7), 432–438.
20 **Balakrishnan, S., Popplewell, N.,** and **Thomlinson, M.** Intelligent robotic assembly. *Int. J. Comput. Ind. Eng.*, 2000, **38**, 467–478.
21 **Kim, I. S., Son, J. S., Park, C. E., Lee, C. W.,** and **Prasad, Y. K. D. V.** A study on prediction of bead height in robotic arc welding using a neural network. *J. Mater. Process. Technol.*, 2002, **130–131**, 229–234.
22 **K—ker, R.** Reliability-based approach to the inverse kinematics solution of robots using Elman's networks. *Eng. Appl. Artif. Intell.*, 2005, **18**, 685–693.
23 **Hasan, A. T., Hamouda, A. M. S., Ismail, N.,** and **Al-Assadi, H. M. A. A.** A new adaptive learning algorithm for robot manipulator control. *Proc. IMechE, Part I: J. Systems and Control Engineering*, 2007, **221**(4), 663–672.
24 **Al-Assadi, H. M. A. A., Hamouda, A. M. S., Ismail, N.,** and **Aris, I.** An adaptive learning algorithm for controlling a two-degree-of-freedom serial ball-and-socket actuator. *Proc. IMechE Part I: J. Systems and Control Engineering*, 2007, **221**(7), 1001–1006.
25 **Siqueira, A. A. G.** and **Terra, M. H.** Neural network-based H_∞ control for fully actuated and underactuated cooperative manipulator. *Control Eng. Pract.*, 2009, **17**, 418–425.
26 **Hasan, A. T., Hamouda, A. M. S., Ismail, N., Aris, I.,** and **Marhaban, M. H.** Trajectory tracking for a serial robot manipulator passing through singular configurations based on the adaptive kinematics Jacobian method. *Proc. IMechE Part I: J. Systems and Control Eng.*, 2009, **223**(3), 393–415.
27 **Hasan, A. T., Ismail, N., Hamouda, A. M. S., Aris, I., Marhaban, M. H.,** and **Al-Assadi, H. M. A. A.** Artificial neural network-based kinematics Jacobian solution for serial manipulator passing through singular configurations. *Adv. Eng. Software*, 2010, **41**, 359–367.
28 **Zurada, M. J.** *Introduction to artificial neural system network*, 1992 (West Publishing Companies, Singapore).