11

Real-Time-Position Prediction Algorithm for Under-actuated Robot Manipulator Using of Artificial Neural Network

Ahmad Azlan Mat Isa, Hayder M.A.A. Al-Assadi and Ali T. Hasan
Faculty of Mechanical Engineering, University Technology MARA (UiTM)
Shah Alam, 40450
Malaysia

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 make the behavior 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, their advantages over fully actuated robots led to many studies to predict their behavior (Yu et al., 1998; Berkemeier & Fearing, 1999; Spong, 1995; Ono et al., 2001; Nakanishi et al., 2000; Funda et al., 1996; Luca et al., 2000; Luca & Oriolo, 2002; Arai & Tachi, 1991; Mukherjee & Chen, 1993;Yu et al., 1993;Bergerman et al., 1995; Mahindrakar et al., 2006; Muscato, 2006; Begovich et al., 2002). 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 behavior is highly desirable for robots in remote or hazardous environments (Yu et al., 1998). Other interesting applications include the Acrobot (Berkemeier & Fearing, 1999; Spong, 1995), the gymnast robots (Ono et al., 2001), the brachiating robots (Nakanishi et al., 2000), and surgical robots (Funda et al., 1996).

The mathematical complexity and wide variety of applications have kept under-actuated robots an area of open research. (Luca et al., 2000; Luca & Oriolo, 2002) have investigated the behavior of a 2R manipulator moving in a horizontal plane with a single actuator at the first joint, neglecting joint friction which is not easy to achieve in real world as it involves high manufacturing cost. Trying to overcome that problem, some researchers have implemented additional equipments such as breaks at the passive joint (Arai & Tachi, 1991; Mukherjee & Chen, 1993; Yu et al., 1993; Bergerman et al., 1995). In this case, the brake can generate torque that means after all that kind of systems 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., 1998) 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. Their results were verified using numerical simulation. Later on, (Mahindrakar et al., 2006) 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 are still lack of generality and systematization. To overcome this problem, artificial intelligence was introduced for prediction and making robot systems able to attribute more intelligence and high degree of autonomy.

Appling fuzzy logic to under-actuated robots (as an artificial intelligence method), there were few studies in recent past (Muscato, 2006; Begovich et al., 2002).

Although the results presented were promising, these results cannot be generalized to other systems, because they only came from practical considerations. Besides, despite the fact that unlike most learning control algorithms, multiple trials are not necessary for the robot to learn the desired trajectory. A major drawback was that Fuzzy Logic based approaches only remembers the most recent data points introduced (Graca & Gu, 1993). Gleaning the learning abilities of genetic algorithms GA (as another method of artificial intelligence) to solve the problem was an alternative. Blending of GA with fuzzy rules, in order to capture the hidden nonlinearities of the system will be useful in developing any learning techniques. (Lee & Zak, 2002) 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 their own drawbacks. To overcome this problem researchers have recommended neural networks so that it would remember the trajectories as it traversed them (Graca & Gu, 1993).

Artificial neural networks (ANNs) have been widely used for their extreme flexibility due to its learning ability and the capability of 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 (Kalogirou, 2001; Hasan et al., 2006). 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 (Balakrishnan et al., 2000; Kim et al., 2002; Köker, 2005; Hasan et al., 2007; Al-Assadi et al., 2007; Siqueira & Terra, 2009).

In real world application, no physical property such as the friction coefficient can be exactly derived. Besides, there are always kinematics uncertainties presence in the real world such as ill-defined linkage parameters and backlashes in gear trains (Hasan et al., 2009; Hasan et al., 2010). In this paper, 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 developed learning algorithm is based on weight adaptation of the network, by minimizing the tracking error after each iteration process. This scheme does not require any prior knowledge of the dynamic model of the system being controlled. 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, 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 Figure 1 show, the space coordinate of the manipulator is parameterized by q.

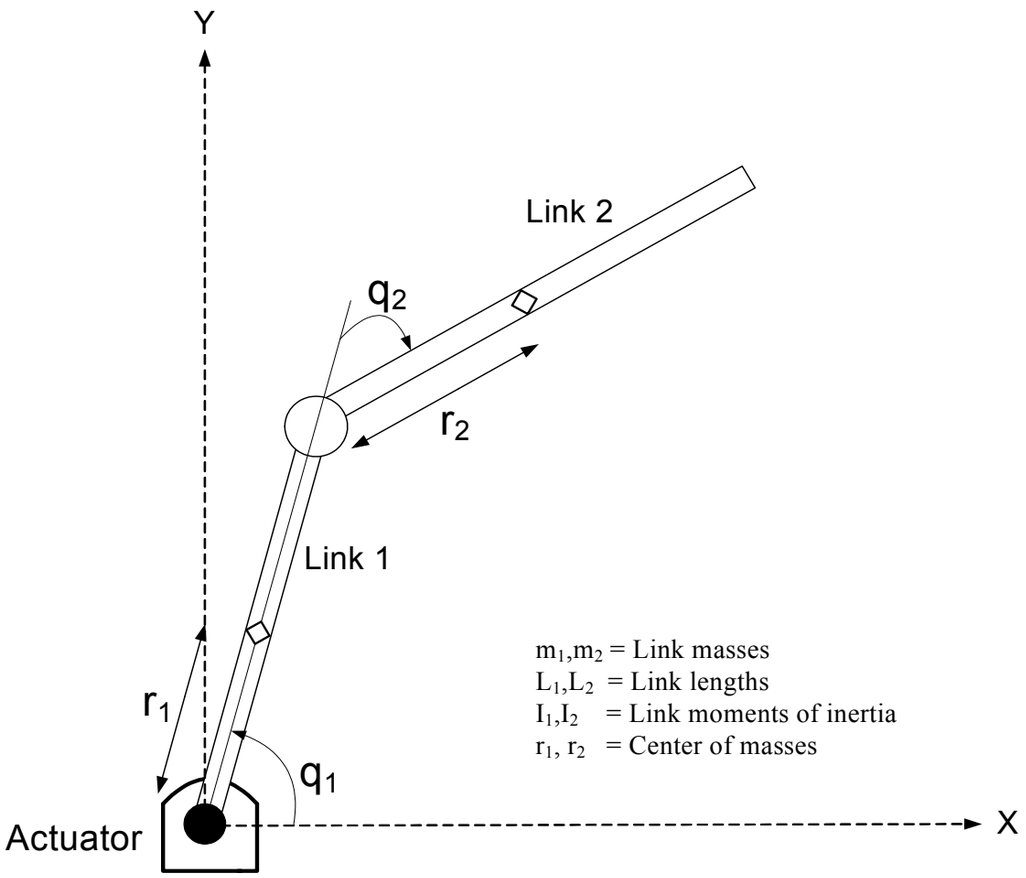


Fig. 1. Schematic diagram of the robot used

The coordinate q_i, $i=1,2$ are the joint angles. The Euler–Lagrange equation of motion is (Mahindrakar et al., 2006):

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

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 $h(q,\dot{q})$. The vector 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,

$m_{11} = c_1 + c_2 + 2c_3 Cosq_2, \quad m_{12} = c_2 + c_3 Cosq_2,$

$m_{21} = m_{12}, \quad m_{22} = c_2,$

$h_1 = -c_3(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2)Sinq_2, \quad h_2 = c_3\dot{q}_1^2 Sinq_2.$

The $F_i, b_i\dot{q}_i, i = 1,2$ represent the Coulomb and viscous friction forces respectively. The set-valued signum function is defined as:

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

The above shown function suffers 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.

3. Experiment procedure

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

The manipulator used is shown in Figure 2, which is actuated only at the first joint. The actuator used 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 have an encoder attached to it, in order to measure the rotation angle and there are torque sensors between the motor output shaft and the robot joint to measure the torque being supplied by the motor. Joints encoders are connected to a computer equipped with MATLAB software through a data acquisition card. The robot arms were made of an aluminum square section beam to ensure a resisting to bending lightweight arm. Length of arms are l_1 = 40 cm and l_2 = 30 cm respectively. The control circuit is made up of computer

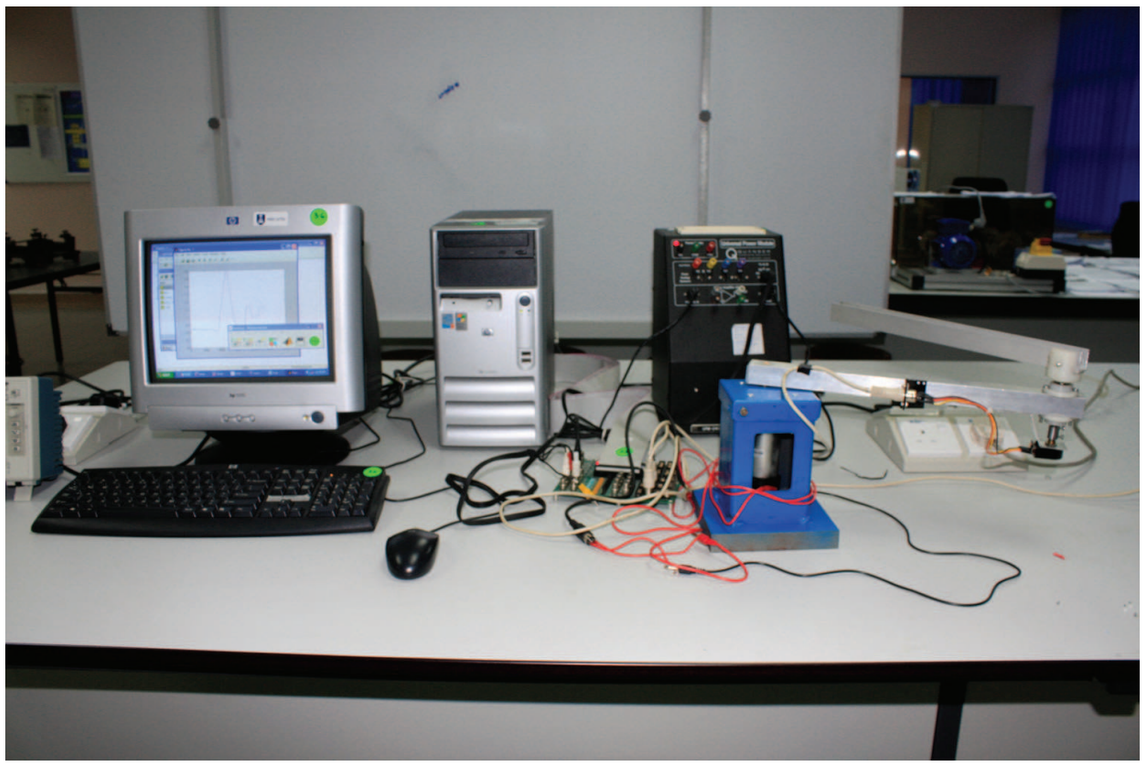


Fig. 2. The robot system used showing the computer, the data acquisition card and the robot arms

with the MATLAB software connected to the robot through a data acquisition card that acquires the motion data of the two links. Input signal is generated by the MATLAB software and transferred to the motor using the electrical board, and the robot response is recorded using the MATLAB software.

A Sinusoidal excitation signal was applied to the actuator causing different torque to the joints and the dynamic coupling effect was moving the passive joint correspondently. As a standard signal generated by the MATLAB, Sinusoidal excitation signal, was chosen in order to cause a robot motion that covers the whole working cell rather than being a specified signal to perform a pre-defined trajectory.

When the excitation signal is given, the motion of the active joint and the corresponding response of the passive joint that can be seen in Figures 3 and 4 respectively were recorded in order to be used in the training process of the ANN.

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 5 shows the network used. The output of the units in layer A 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 layer A, then the total input to the hidden layer, i.e., layer B is:

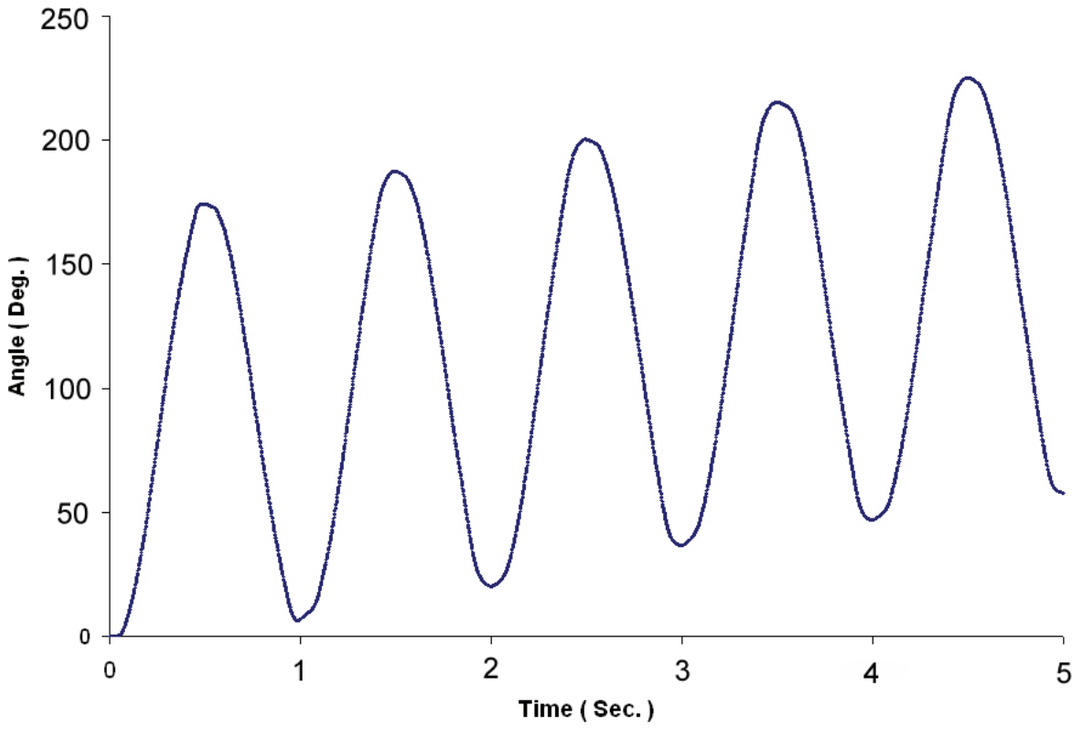


Fig. 3. Trajectory of the active joint when the excitation signal was applied

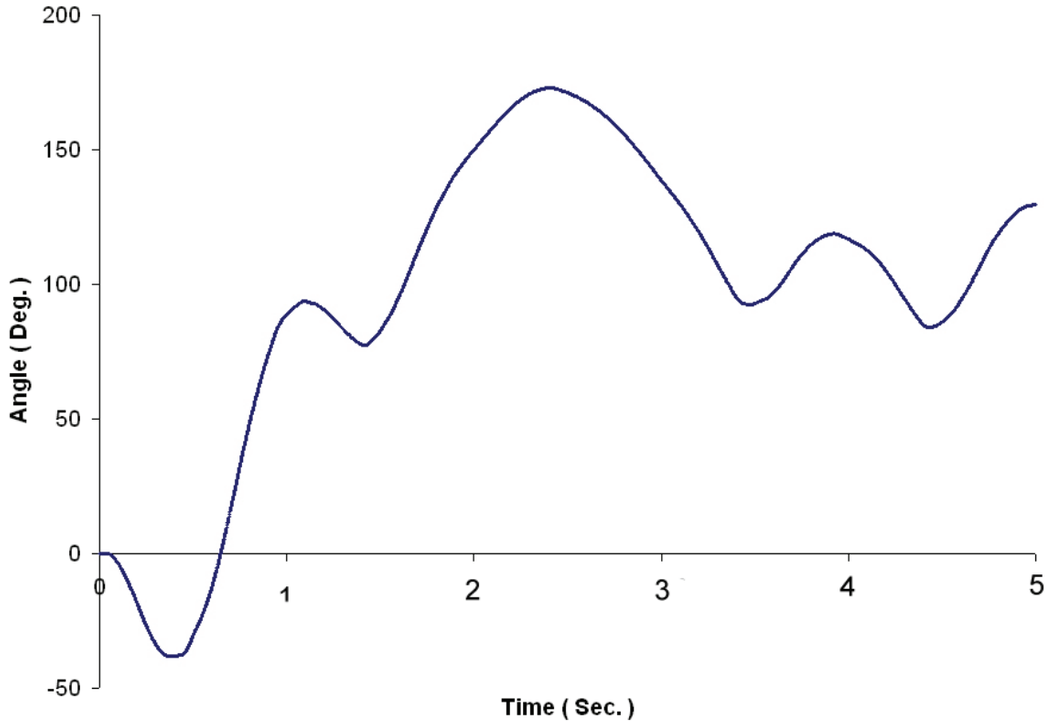


Fig. 4. Corresponding trajectory of the passive joint

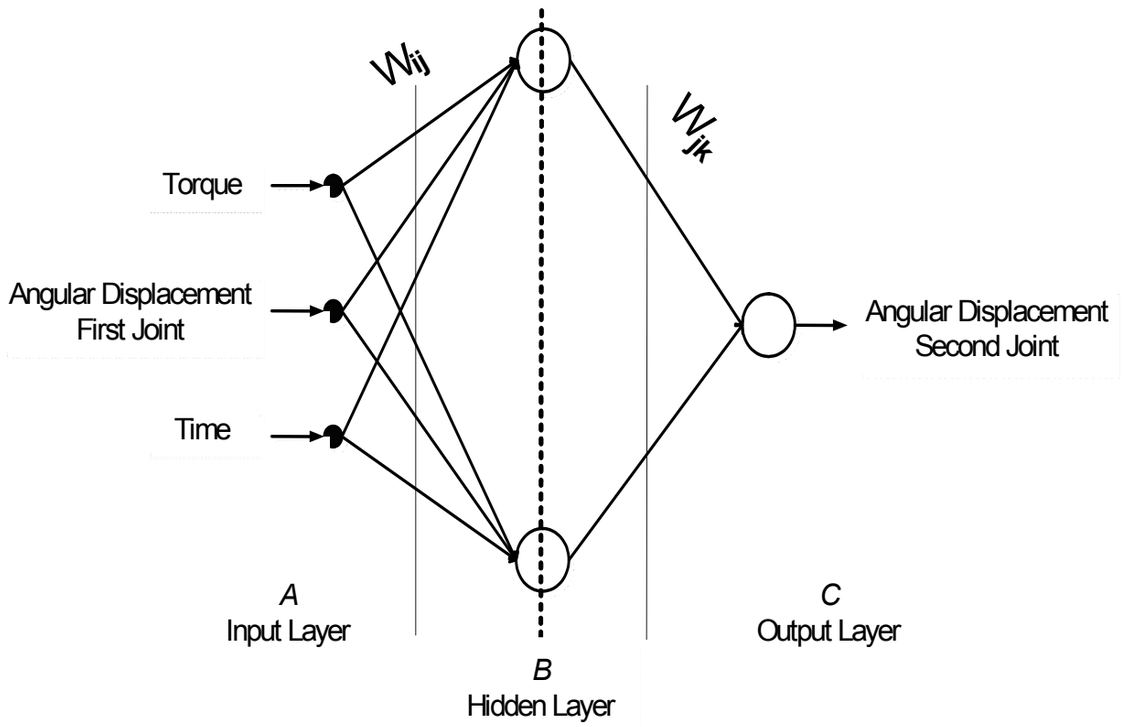


Fig. 5. The topology of the ANN used

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

And the output O_j of a unit in layer B 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 nonlinear 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.
Error at any output unit e_K in the layer C can be calculated by: -

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

Where d_K is the desired output for that unit in layer C and O_K is 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.
To minimize E by the gradient descent method. It is necessary to compute the partial 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 there 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 maybe stated in general 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: -

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

And 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 generalize 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 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 (Kalogirou, 2001; Hasan et al., 2006). Number of hidden neurons and the learning factor are determined by trial and error.

5. Results

A supervised feed forward ANN was designed using C programming language to learn the system behavior 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 can be seen in Figure 5, every neuron in the 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 datasets values had been scaled individually so that the overall difference in the dataset was maximized.

Training data were divided into 50 input–output sets, which covered the entire work cell of the manipulator. To build the control knowledge, 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, number of neurons in the hidden layer was set to 25 with a constant learning factor of 0.9 by trial and error. Figure 6 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% after 100,000 Iterations. Figure 7 graphically shows the trajectory tracking of the passive joint, Results obtained show that the design network is capable of learning and predicting the position of the passive joint successfully.

6. Conclusions and recommendations for further research

In this paper, the Artificial Neural Network technique was applied to the problem of positioning an under-actuated robot manipulator. The position of the passive joint of under-actuated 2R manipulator is now learned through training a network based only on observation of the input–output relationship.

The proposed technique does not require any prior knowledge of the system model, the basic idea of this concept is the use of the ANN to learn the characteristics of the robot system rather than to specify explicit robot system model. Any modification in the physical set-up of the robot such as the addition of a new tool would only require training for a new trajectory without the need for any major system software modification, which is a significant advantage of using neural network approach.

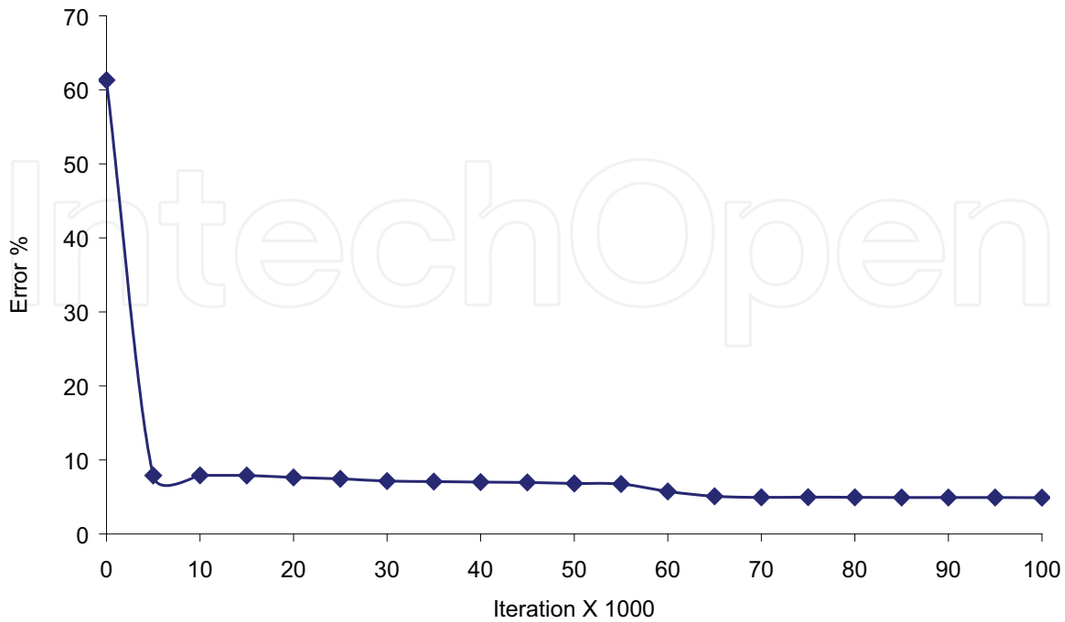


Fig. 6. Building knowledge curve of the system

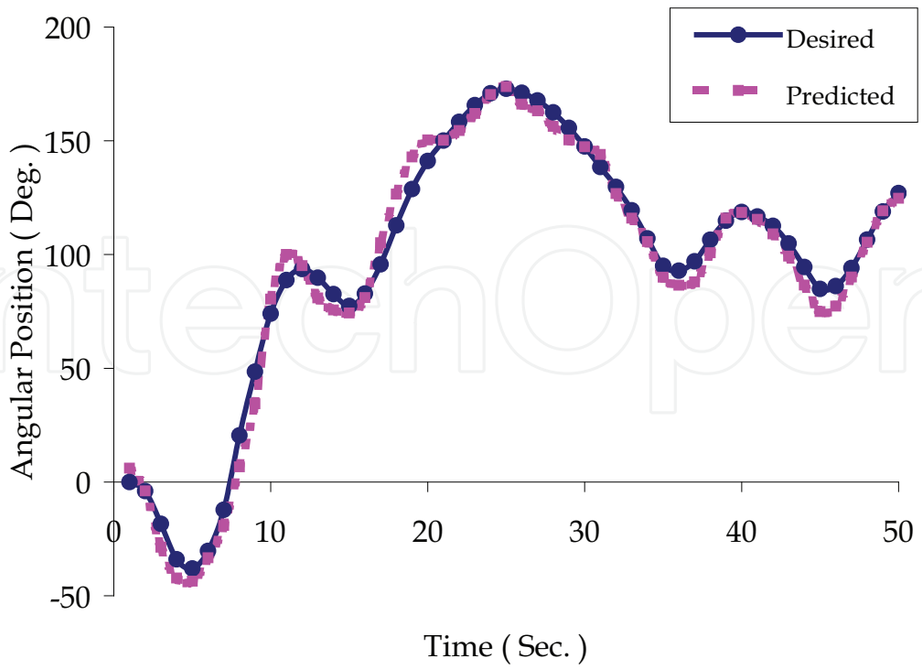


Fig. 7. Predicted trajectory tracking of the passive joint

Results obtained have shown the ability of the network to predict the trajectory of the passive joint, that 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 remembers the most recent data sets introduced.

Backpropagation algorithm was used as a learning algorithm with sigmoid transfer function as an activation function in all neurons, For further research, we recommend that a different learning algorithm, different activation function and/or different number of hidden layers to be used in order to achieve, if possible, a better response in terms of precision and iteration.

7. References

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. Proceedings of the IMechE Part I: Journal of Systems and Control Engineering, 2007,221(7): 1001-1006.

Arai, H. and Tachi, S., Position Control of a Manipulator With Passive Joints Using Dynamic Coupling. IEEE Trans. On Robotics and Automation, 1991,7(4): 528-534.

Balakrishnan, S., Popplewell, N. and Thomlinson, M., Intelligent Robotic Assembly. International Journal of Computers and Industrial Engineering, 2000, 38: 467-478.

Begovich, O., Sanchez, E.N., and Maldonado, M., Takagi–Sugeno fuzzy scheme for real-time trajectory tracking of an under-actuated robot. IEEE Transactions on Control Systems Technology, January 2002, 10(1): 14–20.

Bergerman, M., Lee, C. and Xu, Y., Experimental Study of an Underactuated Manipulator. Proc. 1995 IEEE/RSJ Int. Conf. On Intelligent Robotics and Systems, 1995,2:317-322.

Berkemeier, M. D. and Fearing, R. S., Tracking fast inverted trajectories of the underactuated

acrobot, IEEE Transactions on Robotics and Automation, Aug. 1999, 15(4): 740 – 750.

Funda, J., Taylor, R., Eldridge, B., Gomory, S. and Gruben, K., Constrained Cartesian motion control for teleoperated surgical robots. IEEE Transactions on Robotics and Automation, June 1996, 12(3): 453 – 465.

Graca, R.A. and Gu, Y., A Fuzzy Learning Algorithm for Kinematics Control of a Robotic System. Proceeding of the 32nd Conference on Decision and Control. San Antonio, Texas. December, 1993: 1274-1279.

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. Proceedings of the IMechE Part I: Journal of Systems and Control Engineering, 2009, 223(3): 393-415.

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. International Journal of Advanced in Engineering Software, 2010, 41:359-367.

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. Journal of Advances in Engineering Software, 2006, 37(7): 432-438.

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. Proceeding of the IMechE, Part I: Journal of System and Control Engineering, 2007, 221(4): 663-672.

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. International Journal of Materials Processing Technology, 2002,130–131: 229–234.

Köker, R., Reliability-based approach to the inverse kinematics solution of robots using Elman's networks. International Journal of Engineering Applications of Artificial Intelligence, 2005, 18: 685-693.

Lee, Y. and Zak, S. H., Designing a genetic neural antilock-break system controller. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 198 – 211.

Luca, A.D. and Oriolo, G., Trajectory Planning and Control for Planar Robots With Passive Last Joint. International Journal of Robotics Research, 2002, 21:575-590.

Luca, A.D., Mattone, R. and Oriolo, G., Stabilization of an Underactuated Planar 2R Manipulator. International Journal of Robust and Nonlinear Control, 2000,24:181-198.

Mahindrakar, A.D., Rao, S. and Banavar, R.N., Point-to Point Control of a 2R Planar Horizontal Underactuated Manipulator. International Journal of Mechanism and Machine Theory, 2006, 41:838-844.

Mukherjee, R. and Chen, D., Control of Free-Flying Underactuated Space Manipulators To Equilibrium Manifolds. IEEE Trans. On Robotics and Automation, 1993, 9(5): 561-570.

Muscato, G., Fuzzy Control of an Underactuated Robot With a Fuzzy Microcontroller. International Journal of Microprocessors and Microsystems, 1999,23:385-391.

Nakanishi, J. , Fukuda, T. and Koditschek, D., A brachiating robot controller. IEEE Transactions on Robotics and Automation, April 2000,16(2): 109 – 123.

Ono, K., Yamamoto, K. and Imadu, A. Control of giant swing motion of a two-link horizontal bar gymnastic robot. Advanced Robotics, 2001, 15(4): 449 – 465.

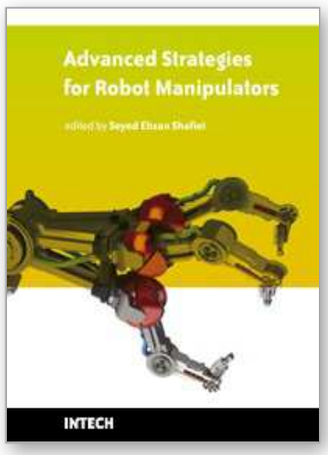
Siqueira, A. A. G. and Terra, M. H., Neural Network-Based H_∞ Control for Fully Actuated and Underactuated Cooperative Manipulator. International Journal of Control Engineering Practice, 2009, 17:418-425.

Kalogirou, S. A., Artificial Neural Networks In Renewable Energy Systems Applications: a review. International Journal of Renewable and Sustainable Energy Reviews. 2001, 5: 373-401.

Spong, M. W., The swing up control problem for the acrobat. IEEE Control Systems Magazine, Feb. 1995, 15(1): 49 – 55.

Yu, K-H, Shito, Y. and Inooka, H., Position Control of an Underactuated Manipulator Using Joint Friction. International Journal of Non-Linear Mechanics, 1998, 33(4): 607- 614.

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. On Intelligent Robots and Systems, 1995, 2:311-316.



Advanced Strategies for Robot Manipulators

Edited by S. Ehsan Shafiei

ISBN 978-953-307-099-5
Hard cover, 428 pages
Publisher Sciyo
Published online 12, August, 2010
Published in print edition August, 2010

Amongst the robotic systems, robot manipulators have proven themselves to be of increasing importance and are widely adopted to substitute for human in repetitive and/or hazardous tasks. Modern manipulators are designed complicatedly and need to do more precise, crucial and critical tasks. So, the simple traditional control methods cannot be efficient, and advanced control strategies with considering special constraints are needed to establish. In spite of the fact that groundbreaking researches have been carried out in this realm until now, there are still many novel aspects which have to be explored.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen