

Research Article

Under-Actuated Robot Manipulator Positioning Control Using Artificial Neural Network Inversion Technique

Ali T. Hasan

*Department of Scholarships and Cultural Relations, Ministry of Higher Education and Scientific Research,
Al-Nidhal Street, Baghdad, Iraq*

Correspondence should be addressed to Ali T. Hasan, ali.taqi.hasan@gmail.com

Received 18 May 2012; Accepted 8 September 2012

Academic Editor: Joanna Józefowska

Copyright © 2012 Ali T. Hasan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper is devoted to solve the positioning control problem of underactuated robot manipulator. Artificial Neural Networks Inversion technique was used where a network represents the forward dynamics of the system trained to learn the position of the passive joint over the working space of a 2R underactuated robot. The obtained weights from the learning process were fixed, and the network was inverted to represent the inverse dynamics of the system and then used in the estimation phase to estimate the position of the passive joint for a new set of data the network was not previously trained for. Data used in this research are recorded experimentally from sensors fixed on the robot joints in order to overcome whichever uncertainties presence in the real world such as ill-defined linkage parameters, links flexibility, and backlashes in gear trains. Results were verified experimentally to show the success of the proposed control strategy.

1. Introduction

Underactuated robot manipulator possesses fewer actuators than degrees of freedom (DOF). Complex internal dynamics, nonholonomic behavior, and lack of feedback linearizability are often exhibited in such systems, making that class of robots a challenging one for synthesis of control schemes. Due to their advantages over fully actuated robots, this type of manipulators has gained the interest of several researchers [1–16]. Saving in weight and cost is an advantage, where low-cost automation and space robots require this feature. Another advantage is that underactuated robots can easily overcome actuator failure due to unexpected accident. Such fault-tolerant control is highly desirable for robots in remote or hazardous environments [1, 2].

The difficulty of the control problem for underactuated mechanisms is obviously due to the reduced dimension of the input space. In particular it has been shown that this system is highly nonlinear and it is impossible to stabilize asymptotically with a smooth feedback [3]. Sordalen et al. [4] have designed an n joint robot controlled by just two motors using nonholonomic gears. Other researchers have

tried controlling an underactuated robot in a gravity field, such as the Acrobot [5–7]. The control of a high-bar robot was investigated by Takashima [8] while Saito et al. [9] have investigated the control of a brachiation robot. Neglecting joint friction which is not easy to achieve in real world as it involves high manufacturing cost, Luca et al. [10, 11] have studied the control of two-link manipulator moving in a horizontal plane with a single actuator at the first joint. In practice, joint friction cannot be neglected especially at the passive joint as friction at the active joint can be directly compensated; the same is not true for the passive joint. Some researchers have tried to overcome this problem by implementing additional equipments such as breaks at the passive joint [12–15]. The dynamic characteristics of a two-link manipulator including joint friction by proposing a mathematical model have been studied by Yu et al. [1] as a result of this research; they have found that the manipulator can be controlled if the friction acts on the passive joint. Later on, Mahindrakar et al. [16] have presented a mathematical model for a two-link underactuated manipulator wherein the motion of the system was confined to a horizontal plane; their proposed dynamic model takes into account

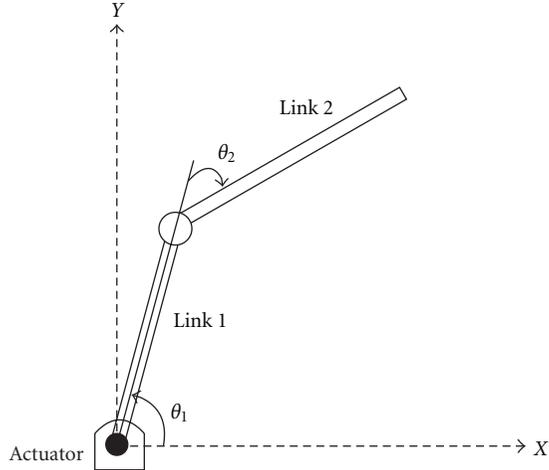


FIGURE 1: Schematic diagram of the manipulator.

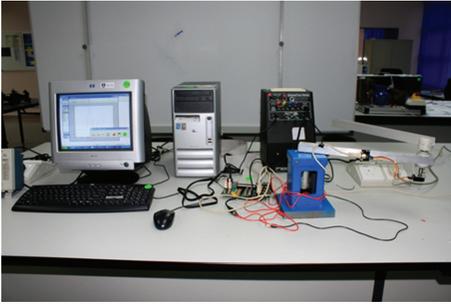


FIGURE 2: The robot system used.

the frictional forces acting on the joints. In this case, any additional equipment such as brakes is not needed in positioning all the joints to desired position.

While some interesting techniques and results have been presented in the previously mentioned publications, the control of such systems still remains an open problem. Most of the control schemes mentioned above either failed to provide a thorough analysis of the overall system stability or assumed that friction forces do not act on the passive joints. Furthermore, the precise knowledge of the dynamic model is generally required. 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, links flexibility, and backlashes in gear train [17, 18].

In the last few years much of the research interest shown in the control of nonlinear systems has focused on methods where classical methodologies valid for linear systems do not give satisfactory performance [19, 20]. Among of which, artificial neural networks (ANN) technique has gained a great deal of interest for their extreme flexibility due to its learning ability and the capability of non-linear function approximation. They can learn from examples, are fault tolerant in the sense that they are able to handle noisy and incomplete data, are able to deal with nonlinear

problems, and, once trained, can perform prediction and generalization at high speed. They are particularly useful in system modeling such as in implementing complex mappings and system identification [21, 22]. During the past decade, many researchers have justified using ANNs for various applications of robotics system control [23–25].

Recently, several neural network approaches have been studied, and the network inversion approach was one of justified approaches for inverse problems solution [26–28]. In this paper, the network inversion technique was implemented; the input was estimated from the given output using the learned network inversely. Data used in this paper was collected experimentally to overcome any uncertainty presents in the real world. The efficiency of the proposed method is shown experimentally using 2R underactuated robot manipulator.

2. The Robot System

The dynamic equations of the manipulator of which the schematic diagram is shown in Figure 1 can easily be derived [29]:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} l_2^2 m_2 + 2l_1 l_2 m_2 C_2 + l_1^2 (m_1 + m_2) & l_2^2 m_2 + l_1 l_2 m_2 C_2 \\ l_2^2 m_2 + l_1 l_2 m_2 C_2 & l_2^2 m_2 \end{bmatrix} \times \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} -m_2 l_1 l_2 S_2 \ddot{\theta}_2^{\bullet\bullet} - 2m_2 l_1 l_2 S_2 \dot{\theta}_1 \dot{\theta}_2 \\ m_2 l_1 l_2 S_2 \dot{\theta}_1^{\bullet} \end{bmatrix} + \begin{bmatrix} f_1 \dot{\theta}_1 \\ f_2 \dot{\theta}_2 \end{bmatrix}, \quad (1)$$

where τ_i is the torque applied to each joint, l_i is the length of each link, m_i is the mass of each link, C_i is $\cos \theta_i$, S_i is $\sin \theta_i$, and f_i is the friction factor of each joint.

In order to analyze the system's behavior, a 2R underactuated manipulator was manufactured. As can be seen in Figure 2, the manipulator 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 has 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. 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 $l_2 = 20$ cm, respectively.

The control circuit is made up of computer with the MATLAB software and an electrical board 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. An excitation signal was applied to the actuator and the dynamic coupling effect was moving the passive joint. It can be observed that after a clockwise torque

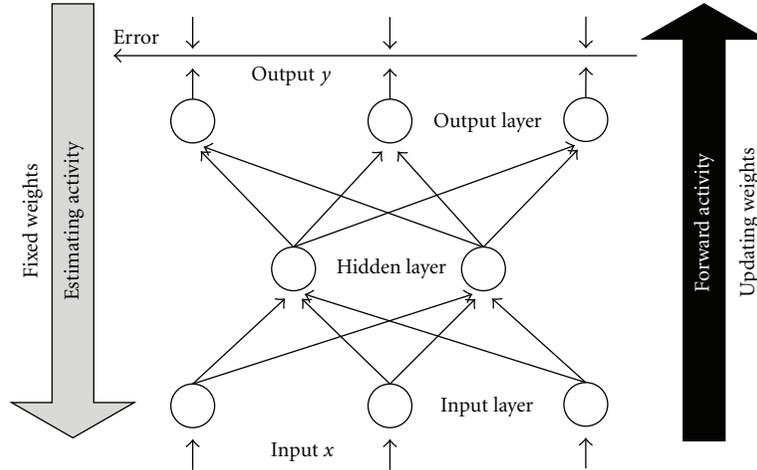


FIGURE 3: Inverse estimation by network inversion.

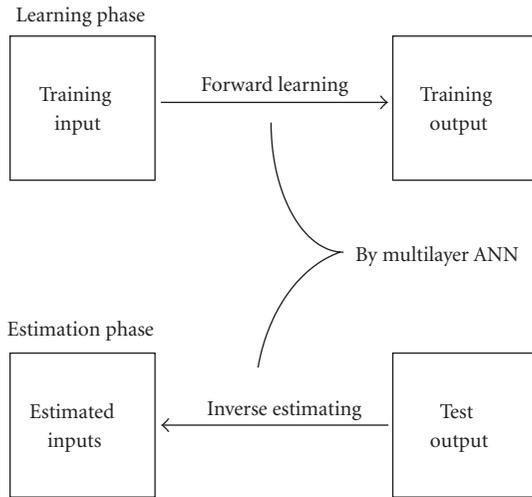


FIGURE 4: Two-step procedure to solve the inverse problem by network inversion.

applied, this joint will rotate clockwise while the passive joint will rotate counterclockwise.

Different methods for collecting data have been found in the literature. Using a prespecified 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 measured directly from sensors fixed on each joint, so every uncertainty in the dynamics of the system will be counted for.

3. Network Inversion Technique

The method of the network inversion was first proposed by Linden and Kindermann [30], where a conventional multilayer neural network is used to solve the forward problem, fixing the weights obtained during the training

process after that in order to be used to estimate the inverse problem parameters.

In the usual multilayer network whose learning has completed, the input/output relation is given by:

$$y = f(w, x), \tag{2}$$

where x , y , and f are the input vector, the output vector, and the function defined by the interlayer weights w of the network, respectively. Given the input vector x , the network calculates the output vector y .

In this method, after finding forward relation f by learning, the output data can be estimated using the fixed weights obtained from the learning process. Then, the input x can be updated according to the calculated input correction signal, based on the duality of the weights and input in (2). Actually, the input is estimated from the output by correcting input based on the output error repeatedly, as shown in Figure 3. By this way, the inverse problem for estimating input x from output y is solved using the multilayer neural network by using forward relation inversely [28].

The technique is implemented in the two phases: *forward learning* and *inverse estimating* to solve the inverse problem by the network inversion. The procedure is shown in Figure 4. In the learning phase, the learning input x is provided with the learning output y and calculates the output error E . Then, the weight is updated until the error is for each input the output produced by the network is the same as (or sufficiently close to) the desired output, this procedure is based on the usual backpropagation method.

In the estimation phase, the relation obtained in the learning is fixed, provides the random input x and the test output y , and calculates the output error E . Then, the input is updated. By using this procedure, the input is estimated from the output [30].

4. ANN Implementation

The network inversion technique was implemented in two phases. The first phase was the forward learning phase in

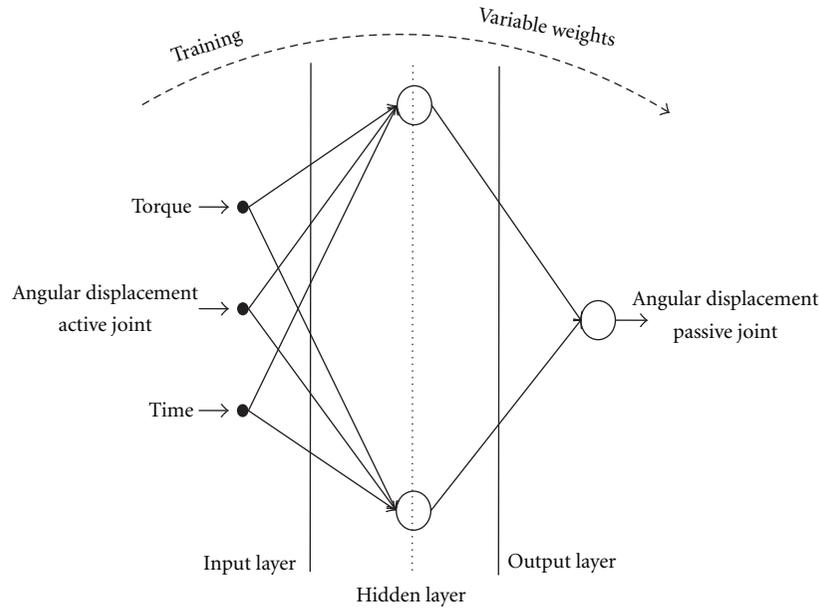


FIGURE 5: The artificial neural network used in the first phase (forward learning).

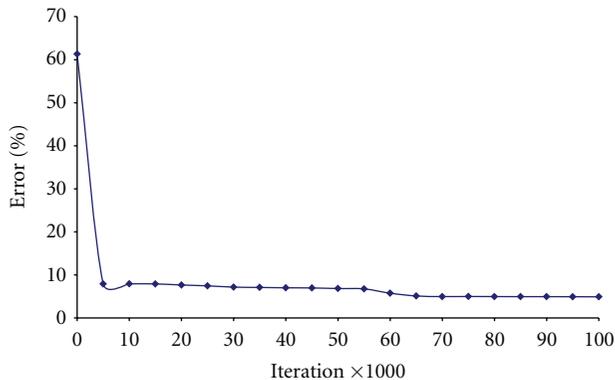


FIGURE 6: The learning curve of the forward learning phase.

order to find the weights; the weight is updated until for each input the output produced by the network is the same as (or sufficiently close to) the desired output. Then, the weight is going to be fixed in the second phase, which is the inverse estimating phase, in order to estimate the passive joint's path.

4.1. Forward Learning Phase. 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 others, 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. Figure 6 shows the building knowledge process for the system. The average absolute error was 4.9% after 100,000 iterations.

4.2. Inverse Estimating Phase. To verify the success of the control algorithm, a new data set that represents new path, a path that the network was not previously trained for, was introduced to the inverted network that can be seen in Figure 7. The weights from the previous stage were fixed and the desired path for the passive joint was applied as an excitation signal to the inverted network in order to estimate the necessary input parameters to drive the passive joint to follow the desired trajectory.

The estimated input parameters were then used as excitation signal to the network used in the first phase to obtain the estimated path of the passive joint. To show the success of this control approach, the experimental trajectory tracking of the estimated passive joint is shown in Figure 8. As this figure clearly shows, the estimated trajectory was closely corresponding to the target trajectory.

5. Conclusions

In this research, we proposed to introduce the network inversion technique to the problem of underactuated robot control that estimating the input parameters necessary to

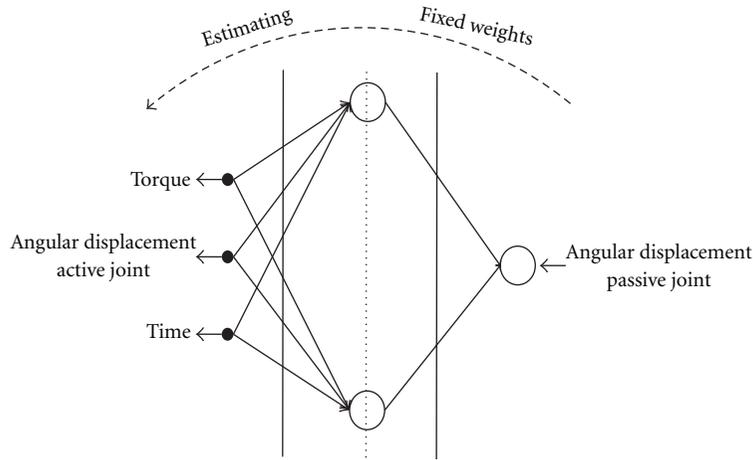


FIGURE 7: The artificial neural network used in the second phase (estimating).

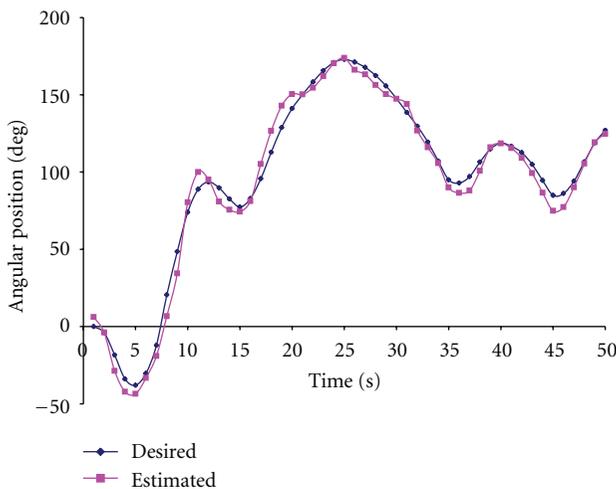


FIGURE 8: Estimated trajectory tracking of the passive joint.

drive the passive joint to follow a desired trajectory from the output parameters of the network. The effect of the network inversion was confirmed by the simulation of a 2R underactuated robot arm.

As a conclusion, it was shown that using the network inversion technique to estimate the input parameters from the output parameters of the robot arm was a successful technique. Consequently, it was shown that the network inversion could solve the problem of underactuated robot control.

References

[1] K.-H. Yu, Y. Shito, and H. Inooka, "Position control of an underactuated manipulator using joint friction," *International Journal of Non-Linear Mechanics*, vol. 33, no. 4, pp. 607–614, 1998.

[2] K. M. Lynch, N. Shiroma, H. Arai, and K. Tanie, "Collision-free trajectory planning for a 3-DoF robot with a passive joint,"

International Journal of Robotics Research, vol. 19, no. 12, pp. 1171–1184, 2000.

[3] G. Oriolo and Y. Nakamura, "Control of mechanical systems with second-order nonholonomic constraints: underactuated manipulators," in *Proceedings of the 30th IEEE Conference on Decision and Control*, pp. 2398–2403, Brighton, UK, December 1991.

[4] O. J. Sordalen, Y. Nakamura, and W. J. Chung, "Design of a nonholonomic manipulator," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 8–13, May 1994.

[5] J. Hauser and R. M. Murray, "Nonlinear controllers for non-integrable systems: the Acrobot example," in *Proceedings of the American Control Conference*, pp. 669–671, May 1990.

[6] M. W. Spong, "Swing up control of the acrobot," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pp. 2356–2361, May 1994.

[7] M. D. Berkemeier and R. S. Fearing, "Tracking fast inverted trajectories of the underactuated Acrobot," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 4, pp. 740–750, 1999.

[8] S. Takashima, "Control of gymnast on a high bar," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1424–1429, Osaka, Japan, 1991.

[9] F. Saito, T. Fukuda, and F. Arai, "Swing and locomotion control for a two-link Brachiation robot," *IEEE Control Systems Magazine*, vol. 14, no. 1, pp. 5–12, 1994.

[10] A. D. Luca, R. Mattone, and G. Oriolo, "Stabilization of an Underactuated Planar 2R Manipulator," *International Journal of Robust and Nonlinear Control*, pp. 181–198, 2000.

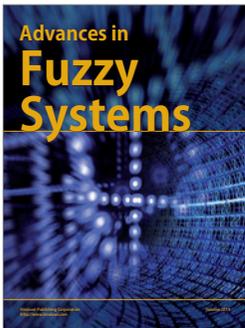
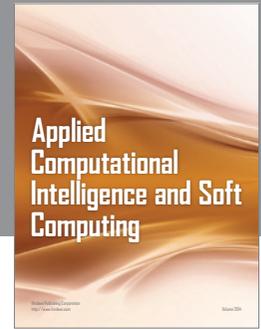
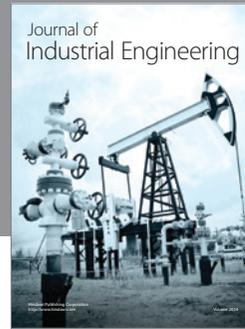
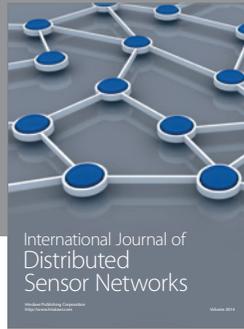
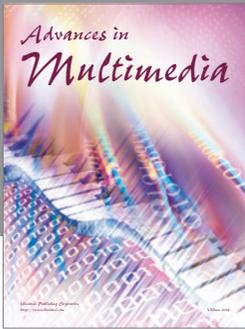
[11] A. De Luca and G. Oriolo, "Trajectory planning and control for planar robots with passive last joint," *International Journal of Robotics Research*, vol. 21, no. 5-6, pp. 575–590, 2002.

[12] H. Arai and S. Tachi, "Position control of manipulator with passive joints using dynamic coupling," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 4, pp. 528–534, 1991.

[13] R. Mukherjee and D. Chen, "Control of free-flying underactuated space manipulators to equilibrium manifolds," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 561–570, 1993.

[14] K.-H. Yu, T. Takahashi, and H. Inooka, "Dynamics and motion control of a two-link robot manipulator with a passive

- joint,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 311–316, August 1995.
- [15] M. Bergerman, C. Lee, and Y. Xu, “Experimental study of an underactuated manipulator,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 317–322, August 1995.
- [16] A. D. Mahindrakar, S. Rao, and R. N. Banavar, “Point-to-point control of a 2R planar horizontal underactuated manipulator,” *Mechanism and Machine Theory*, vol. 41, no. 7, pp. 838–844, 2006.
- [17] A. T. Hasan, A. M. S. Hamouda, N. Ismail, I. Aris, and M. H. Marhaban, “Trajectory tracking for a serial robot manipulator passing through singular configurations based on the adaptive kinematics Jacobian method,” *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 223, no. 3, pp. 393–415, 2009.
- [18] A. T. Hasan, N. Ismail, A. M. S. Hamouda, I. Aris, M. H. Marhaban, and H. M. A. A. Al-Assadi, “Artificial neural network-based kinematics Jacobian solution for serial manipulator passing through singular configurations,” *Advances in Engineering Software*, vol. 41, no. 2, pp. 359–367, 2010.
- [19] O. Begovich, E. N. Sanchez, and M. Maldonado, “Takagi-Sugeno fuzzy scheme for real-time trajectory tracking of an underactuated robot,” *IEEE Transactions on Control Systems Technology*, vol. 10, no. 1, pp. 14–20, 2002.
- [20] H. M. A. A. Al-Assadi, A. M. S. Hamouda, N. Ismail, and I. Aris, “An adaptive learning algorithm for controlling a two-degree-of-freedom serial ball-and-socket actuator,” *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 221, no. 7, pp. 1001–1006, 2007.
- [21] S. A. Kalogirou, “Artificial neural networks in renewable energy systems applications: a review,” *Renewable and Sustainable Energy Reviews*, vol. 5, no. 4, pp. 373–401, 2000.
- [22] A. T. Hasan, A. M. S. Hamouda, N. Ismail, and H. M. A. A. Al-Assadi, “An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 D.O.F serial robot manipulator,” *Advances in Engineering Software*, vol. 37, no. 7, pp. 432–438, 2006.
- [23] B. Karlik and S. Aydin, “Improved approach to the solution of inverse kinematics problems for robot manipulators,” *Engineering Applications of Artificial Intelligence*, vol. 13, no. 2, pp. 159–164, 2000.
- [24] R. Köker, “Reliability-based approach to the inverse kinematics solution of robots using Elman’s networks,” *Engineering Applications of Artificial Intelligence*, vol. 18, no. 6, pp. 685–693, 2005.
- [25] A. T. Hasan, A. M. S. Hamouda, N. Ismail, and H. M. A. A. Al-Assadi, “A new adaptive learning algorithm for robot manipulator control,” *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 221, no. 4, pp. 663–672, 2007.
- [26] Y. Kuroe, Y. Nakai, and T. Mori, “New neural network learning of inverse kinematics of robot manipulator,” in *Proceedings of the 1994 IEEE International Conference on Neural Networks*, vol. 7, pp. 2819–2824, June 1994.
- [27] P. Martín and J. D. R. Millán, “Robot arm reaching through neural inversions and reinforcement learning,” *Journal of Robotics and Autonomous Systems*, vol. 31, no. 4, pp. 227–246, 2000.
- [28] T. Ogawa, H. Matsuura, and H. Kanada, “A solution of inverse kinematics of robot arm using network inversion,” in *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation*, pp. 858–862, November 2005.
- [29] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics Control, Sensing, Vision and Intelligence*, McGraw-Hill, New York, NY, USA, 1987.
- [30] A. Linden and J. Kindermann, “Inversion of Multilayer Networks,” in *Proceedings of the International Joint conference on Neural Networks*, vol. 3, p. 188, 1993.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

