International Journal of Digital Information and Wireless Communications (IJDIWC) 2(1): 33-39
The Society of Digital Information and Wireless Communications, 2012(ISSN 2225-658X)

Practical Neural Controller for Robotic Manipulator

Baker Al-Bahri[#1] and Zainab Al-Qurashi[#2]

[#1] Electrical and Computer Engineering, Faculty of Engineering, Sohar University, Oman
baker_albahri@yahoo.com, bbahri@soharuni.edu.om
[#2] Higher College of Technology, Department of Information Technology, Muscat, Oman
Zainab.alqurashi@gmail.com, zainab_alqurashi@hct.edu.om

ABSTRACT

This paper presents the design and practical implementation of inverse neural controller which is used to control the operation of six Degree Of Freedom (6DOF) robotic manipulator. An efficient off-line training method has been proposed which is used to train the neural network controller to be used as fed forward controller in the real time applications without need to the on-line training which is time consumption method. All the control algorithms and real time programming had been written with the aid of the MATLAB software.

KEYWORDS

Artificial Intelligent AI, Neural Networks, Neural Controller, Fed Forward Controller, and Robotic Manipulator.

1 INTRODUCTION

Since neural networks have the ability to comprehend and learn about complex plant structures, disturbances, environment and different operating conditions, they (NN) are used in the artificial intelligent controllers. One of the most important features of the neural networks their ability to learn or model unknown systems even when they are complex systems. Updating of the neural network is the process of modifying the weights and biases of the neural network to minimize the error between its actual and desired outputs. Updating can be done by using off-line training which is evaluated before using neural network controller in real time operation or on-line training which is evaluated during real time operation of the neural network controller, however sometimes both off-line and on-line training are used to get good performance for the neural controller. On-line training can map changes in the inputs with the outputs during the real time operation better than off-line training of the neural network controller [1,2]. On the other hand on-line training has heavy computations which require long time during real time operation and this is the main problem with the on-line training of neural netw845ork controller. This time becomes very critical when it is greater than sampling time of the system operation and sometime makes the system unstable. To avoid this problem, efficient off-line training is used to train neural controller to model a system even it is non-linear system, this can be done by selecting efficient training algorithm, good training data and efficient controller scheme.

2 INVERSE NEURAL NETWORK CONTROLLER (INNC)

There are several controller schemes that use neural networks as an intelligent component. One of these controller schemes uses the neural networks as a FFC [1,3]. In this control scheme, the neural network is trained to identify (learn) the inverse of the plant dynamic with the aid of a set of training data then the trained neural network is used as FFC. In this paper an INNC has been designed, trained, implemented and tested practically to control the operation of the MA2000 robotic manipulator.

MA2000 is 6DOF articulated type robotic manipulator has six revolute joints; these are base, shoulder, elbow, pitch, yaw, and roll. Figure 1 shows MA2000 robotic manipulator structure. For each one of the six actuators of the MA2000 manipulator there is an INNC which is used to achieve fast response with minimum steady state error in that joint to reach its desired joint variable which is computed using the inverse kinematic of the manipulator system [4]. All the six INNCs have three layers but they differ in their inputs, number of hidden neurons and training data set. There are nine inputs to the INNCs of the miner joints in the wrist structure which are pitch, yow and roll axes; while there is only one output from it which is the control signal of that joint in the MA2000 manipulator. The 1st input of the INNC of the i-th joint of the miner joints is its desired value $\theta_{id}(k+1)$. The 2nd to 7th inputs are the current and previous values of that joint variable i.e. $\theta_{ia}(k)$, $\theta_{ia}(k-1)$, $\theta_{ia}(k-2)$, $\theta_{ia}(k-3)$, $\theta_{ia}(k-4)$ and $\theta_{ia}(k-5)$. The 8th input is the current change in the i-th joint variable, i.e. $\theta_{ia}'(k) =\theta_{ia}(k)-\theta_{ia}(k-1)$; while the last input to the INNC is $u_i(k-1)$ which is the previous value of the control signal of the joint i. In the major joints (Base, shoulder and elbow) the coupling effects between these parts become significant and they should be considered in the controller design. Thus for the first three joints (major joints) there are four extra inputs for their controllers, these extra inputs are the previous joint variable and the pervious control signal of the other two major joints. Figure 2 shows the INNC of the base joint of the robotic manipulator. There are thirteen inputs and one output. The same controller scheme is used for the shoulder and elbow joints but $\theta_{Sa}(k-1)$, $u_S(k-1)$ are replaced by $\theta_{Ba}(k-1)$, $u_B(k-1)$ in the shoulder controller and $\theta_{Ea}(k-1)$, $u_E(k-1)$ are replaced by $\theta_{Ba}(k-1)$, $u_B(k-1)$ in the elbow INNC. While the last four inputs $\theta_{Sa}(k-1)$, $u_S(k-1)$, $\theta_{Ea}(k-1)$, $u_E(k-1)$ are not used for the miner joints (pitch, yaw and roll) because their joints effects can be neglected.

Figure 1. MA2000 robotic manipulator structure.

3 TRAINING OF INNC

In this controller scheme, the neural network is trained off-line to learn the inverse dynamic of the i-th joint in the MA2000 manipulator. The off-line training phase is done before using the neural network as Fed Forward Controller (FFC). The training of INNC of the base joint is shown in Figure 3 which provides a method to minimize the overall Mean Square Error (MSE) of the training data. The off-line training of INNC of joint i is achieved by using hard training signal which is taken from the open loop response of joint i. The training data must be extended to fill all the range of the input and output variables of that joint. Otherwise, the neural network cannot learn the inverse dynamic of joint exactly which may cause bad system response especially in the range of the output or input variables that were not included in the training signal even MSE of the training process is small. In this paper only off-line training is used to train INNC, so that it is important to use a hard and complex training signal to get good system response with the INNC. The on-line training is not used in this controller because it is time consume during the real time operation of MA2000 manipulator which sometimes makes system unstable due to heavy mathematical computations in the on-line training phase which take a time more than the sampling interval. However practical results show good off-line training of INNC can give good system response. The off-line training of the INNC is achieved by using batch training method [5]. This training technique gives better and faster convergence from single train step. In this training technique, a batch of training data is used, which includes input vectors and the corresponding output vectors. The output of the neural network is computed according to each input vector, then the MSE is computed which is equal to:

$$MSE = \sqrt{e_1^2 + e_2^2 + e_3^2 + ... + e_n^2} \quad (1)$$

$$e_j(k) = \hat{u}_j(k) - u_j(k) \quad (2)$$

Where $e_j(k)$, u_j^(k) and $u_j(k)$ are the error due to the j^{th} vector in the training data at sample k, actual output of INNC in the training phase at sample k ,and desired output of INNC in the training phase at sample k. The new network weights are computed and updated in order to minimize the MSE. Then the batch training method applies the inputs to the new network, calculates outputs, compares them to the associated output values in the training data and calculates the MSE. If the error goal is satisfactory, then the training is stepped. Otherwise training goes through another loop.

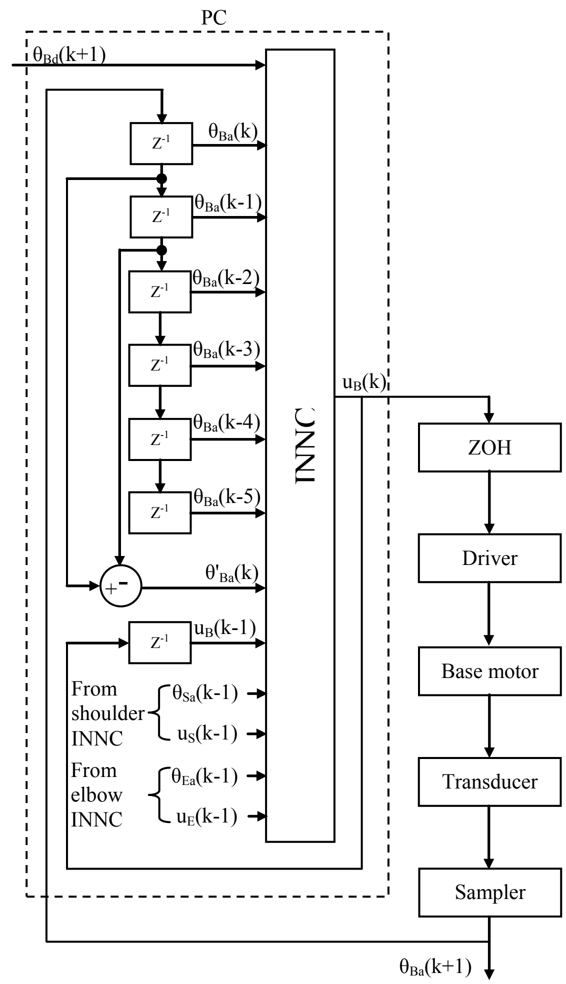

Figure 2. INNC of the base joint in the robotic manipulator.

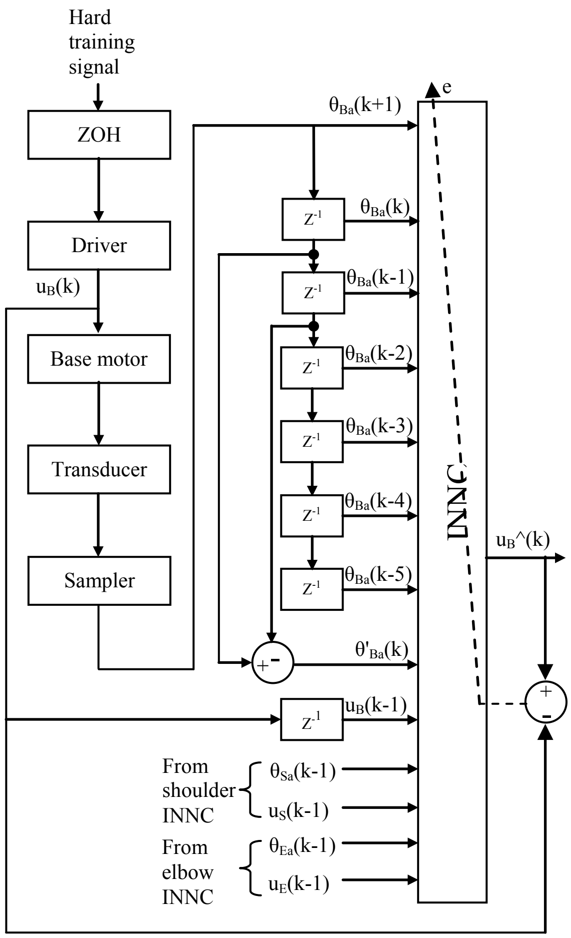

Figure 3. Training of the INNC of the base joint.

4 OPEN LOOP RESPONSE OF MA2000 MANIPULATOR

Open loop response of a system can be obtained from the system under condition of no control action. Simply this can be done by applying an input signal to the certain joint in the MA2000 manipulator and read the corresponding joint variable of the joint without using any controller. Figure 4 illustrates the open loop response of the base joint due to a square wave (±7.5 Volt) as an input signal to the base motor. Base joint in the MA2000 manipulator is the largest joint in this manipulator structure, so it is the most critical link in the manipulator structure because it has largest effective mechanical time varying load (i.e. shoulder, elbow, pitch, yaw, roll, gripper, and load structures). Any change in the manipulator structure will directly effect on the base joint and link. The load inertia of the base motor may change at any time because it depends on the shape and the mass of the base effective load of manipulator. The shape of the effective load may change if any one of the other links in the manipulator changes its joint variable or if there is any change in the gripper's load in this case, the mass of the effective load of the base joint will change too. For these reasons, the base joint becomes the most sensitive joint in the manipulator so that the performance of INNC will be tested practically on this joint.

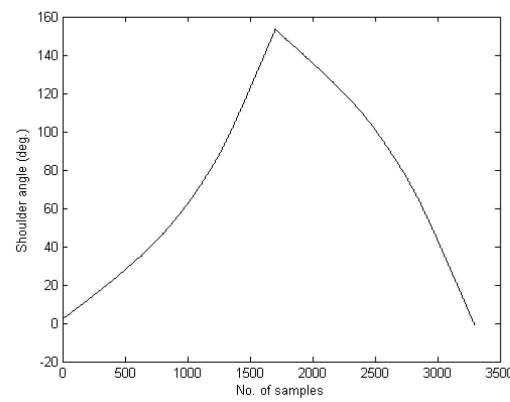

Figure 4. Open loop response of the base joint in MA2000 due to (±7.5Volt) square input.

5 TESTING WITH THE INNC

A multilayer neural network contains 23 neurons in the hidden layer is used as INNC for each joint in the MA2000 manipulator. First the neural network is trained off-line by using batch training technique with Levenberg Marquaidt [5] training algorithm is used to learn the inverse dynamic model of a certain joint in the MA2000 manipulator. Then it is used as FFC for that joint. The neural network inverse model has been trained by using hard training signal taken from the open loop response of that joint of the manipulator. Figure 5 illustrates the hard training signal which is used to train the neural network to learn the inverse of the base joint.

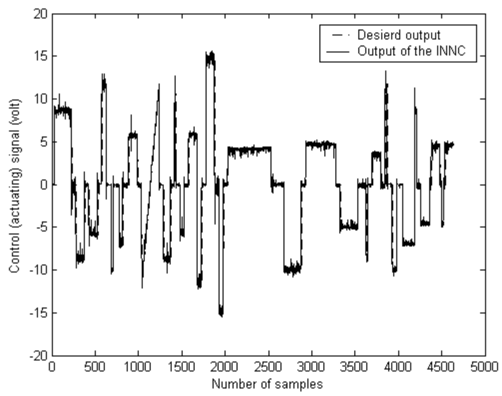

Figure 5. The training signal of the INNC to learn the inverse model of the base joint.

The batch size of the training data (n) is 4700 input/output vectors. The neural network is trained using Levenberg Marquaidt algorithm until an acceptable MSE is reached, which is equal to 161 and then the neural network is connected as INNC. Note that $\theta_{Ba}(k+1)$ in the training phase is replaced by $\theta_{Bd}(k+1)$; while other inputs to the INNC remain as they were used in the off-line training phase. The base step response using the INNC with thirteen inputs is shown in Figure 6; while Figure 7 illustrates the output of base INNC (control signal). It is clear that by using efficient off-line training for INNC, the system response reaches the desired value. It combines with low overshoot and decayed oscillation until the desired value is reached. The steady state system response with the INNC is good and the actual value reaches the desired value without (or with small) steady state error. Also the INNC can move the joint for small step change as it is shown in Figure 8. It is important to note that, in this figure, to explain the system performance for small change in the joint variables, the system response is drawn in steps of ADC (not as angle in degree or radian). Each step equals to:

$$\begin{aligned} step/\deg. &= \frac{steps \quad of \quad ADC}{base \quad angle \quad range} \\ &= \frac{2^{12}}{270^{o}} = 15.17 step/\deg. \end{aligned} \quad (3)$$

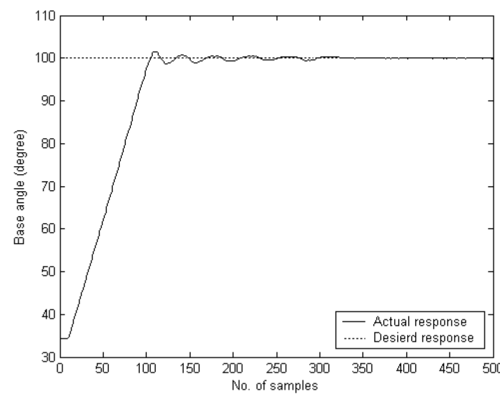

Figure 6. Step response of the base joint using INNC.

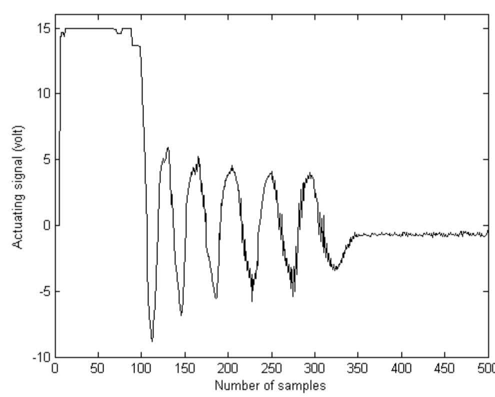

Figure 7. Control signal of base actuator using INNC.

The oscillation that appears in the system response in Figure 8 is not a mechanical oscillation in the base structure but it represents the noise due to the small step size of the analog to digital converter which is 2.4mV.

There are two important points must be investigated during training INNC. The first one is MSE which is given in equation (1). For same INNC structure as the MSE decreases the INNC gives better performance in the real time operation. Figure 9 shows base step response using INNC with the same internal structure, number of the hidden neurons, and training data but they differ in their MSE, the first INNC has MSE of 161 while the second controller has MSE of 304. It can be shown that the system response with the first INNC (MSE=161) is better than the system response with the second INNC (MSE=304). In the first case the system response reaches the desired value with zero steady state error while with the second INNC there is a steady state error in the system response.

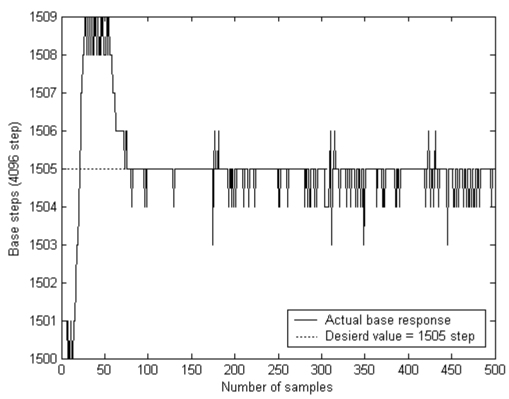

Figure 8. Base step response (small change) of the MA2000 robotic manipulator using INNC.

The second important point which affects the performance of INNC is the number of neurons in the hidden layer. From practical results, it was found that, if two neural networks with different number of neurons in the hidden layer and same training data are trained to learn the INNC for a certain joint in the MA2000 manipulator until they reach the same value of the MSE, the INNC with the largest number of the neurons in the hidden layer gives better system response than that of smaller number of neurons in the hidden layer even the two INNC have the same MSE and same training data. This is due to that INNC with largest number of neurons in the hidden layer can map the input vectors with the output vectors in more details (due to the large number of the neurons in the hidden layer) than the other uses small number of neurons in the hidden layer. Figure 10 shows the base step response using INNC with the same MSE but differ in the number of the neurons in the hidden layer (23 and 9 neurons).

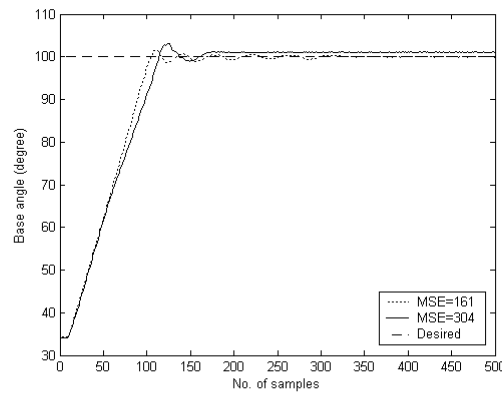

Figure 9. Step response of the base joint using two INNCs with different MSE.

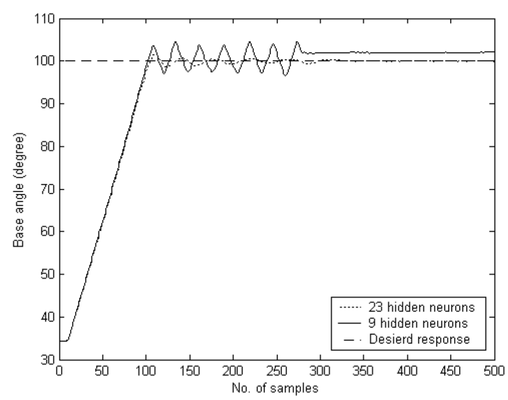

Figure 10. Step response of the base joint using two INNCs with different number of neurons in the hidden layer

As expected system response with the INNC of largest number of neurons in the hidden layer is better than that of small number of neurons in the hidden layer. However, there is a limit for the number of neurons in the hidden layer because if the number of neurons in the hidden layer becomes large this may push the system to be unstable.

7 CONCLUSIONS

The following guide points are concluded form practical design, implementation and results of INNC trained by off-line training. The INNC is used to control the operation of robotic manipulator in the real time operation:

- The off-line training signal for the INNC must cover all ranges of the input and output variables of the system to get best training of the neural network in the off-line training phase.
- Increasing the number of neurons in the hidden layer (to certain value found by trial) that are used in the hidden layer of the INNC gives better controller performance in the real time operation.
- Good off-line training of the neural network to learn inverse dynamic of complex plant structure by using good hard training signal which is taken from the open loop response of the system, lead to good learning of the neural network and on-line training is not required in the real time operation of the INNC.
- System steady state error can be zero with the use of the INNC.
- INNC can be used for small step change in the system output.
- As MSE in the off-line training decreases, the performance of the INNC will be better in the real time operation.

8 REFERENCES

1. B. Al-Bahri.: Hybrid neuro-fuzzy controller for practical laser detection and tracking system, Second Minia International Conference for Advanced Trends in Engineering, Minia, Egypt (2002).
2. Omatu S., Khalid M., and Yusof R.: Neuro-control and its Applications. Second edition, Springer (1996).
3. Wolpert D. and Kawato M.: Multiple Paired Forward and Inverse Models for Motor Control. Pergamion, Neural Networks, pp 1317-1329 (1998).
4. B. Al-Bahri and M. Alfazi.: An algorithm to solve the inverse kinematics problem of robotic manipulator based on the rotation vectors, Third IEEE-GCC, Gulf International Convention and Exhibition Center, Manama, Bahrain (2006).
5. Demuth H. and Beale M.: Neural Network Toolbox for Use with MATLAB. MathWorks (2002).