JOURNAL OF ENGINEERING RESEARCH AND TECHNOLOGY, VOLUME 1, ISSUE 2, JUNE 2014

Improving Trajectory Tracking Performance of Robotic Manipulator Using Neural Online Torque Compensator

Mahmoud M. Al Ashi[1], Iyad Abu Hadrous[2], and Hatem Elaydi[3]

Abstract— This paper introduces an intelligent adaptive control strategy called Neural Online Torque Compensator (NOTC) based on the learning capabilities of artificial neural networks (ANNs) in order to compensate for the structured and unstructured uncertainties in the parameters of a robotic manipulator trajectory tracking control system. A two-layered neural perceptron was designed and trained using an Error Backpropagation Algorithm (EBA) to learn the difference between the actual torques generated by the joints of a 2-DOF robotic arm and the torques generated by the computed torque disturbance rejection controller that was designed previously. An objected oriented approach based on Modelica was adopted to develop a model for the whole robotic arm trajectory tracking control system. The simulation results obtained proved the effectiveness of the NOTC compensator in improving the performance of the computed torque disturbance rejection controller by compensating for both structured and unstructured uncertainties.

Index Terms—Trajectory Tracking Control, Robotic Manipulator, Neural Network Control, Computed Torque Method, backpropagation algorithm

I Introduction

Robotic Manipulators are widely used in different applications of industrial automation such as car manufacturing, space exploration, search and rescues, waste treatment in nuclear plants, in addition to their different applications in medical surgery. Due to their increasing versatile and complex tasks, the development of intelligent control mechanisms to optimize the trajectory tracking capability of robotic manipulators has become a necessity and an important research area. One of the commonly-used methods for controlling the trajectory tracking of robotic manipulators is the computed-torque method [1]. This method was used in [2] as a disturbance torque rejection method to cancel out the coupled torques generated by the joints of a 2-DOF robotic arm due to its inverse dynamics. Despite the proved effectivenss of the computed-torque method in improving the trajectory tracking of the robotic arm, it showed a poor performance in compensating for both structured and unstructured uncertainties. A structured uncertainty means that the mathematical models for both the manipulator and the actuators are accurate but the values of the parameters used in these models are imprecise. An unstructured uncertainity represents any unmodeled dynamics such as gear frictional forces, sudden changes in the payloads held by the end-effector during the online operation, etc. [3].
Several adaptive approaches have been used to maintain the accuracy of the trajectory tracking of robotic manipulators in the presence of structured uncertainties [4, 5]. However, the adaptive control approaches may not be effective to compensate for the unstructured uncertainties. This led to the development of more intelligent adaptive control strategies that help to maintain the trajectory tracking capability of robotic manipulators even in the presence of unstructured uncertainties.
Artificial Neural networks (ANNs) are one of the modern intelligent tools that have been utilized in position trajectory tracking applications of robotic manipulators. This is due to their simple structure and model as well as their universal complex function approximation and learning capabilities gained through simple training algorithms [6]. The primary goal of incorporating a neural network in a robotic manipulator trajectory tracking control system is to either learn only the unknown factors in the system such as the structured and unstructured uncertainties in what so called a model-based control system configuration, or to identify the whole plant dynamics and adaptively compensate for any online changes in such dynamics in what so called a non-model-based control system configuration [7]. In [8], a radial basis neural network was used in a new adaptive control approach that aims to improve the transient response of a robotic manipulator by formulating an objective performance bound. This performance bound enables the robot end-effector to track a desired set-point or trajectory within a unified bound.
In this paper, a two-layered neural network has been designed and utilized in a model-based trajectory tracking control system configuration involving a computed-torque controller that was designed and tested on a 2-DOF robotic manipulator

1. *Mahmoud M. Alashi is a Postgraduate Researcher with the Electrical Engineering Department, Islamic University of Gaza, Gaza, Palestine.*
2. *IyadAbu Hadrous is with the Department of Engineering Professions, Palestine Technical College, Dier El Balah, Gaza, Palestine.*
3. *Hatem Elaydi is with the Electrical Engineering Department, Islamic University of Gaza, Gaza, Palestine.*

in a previous work [2]. The neual network will be trained online using the steepest descent error backpropagation algorithm to compensate for both structured and unstructured uncertainties that caused the poor performance of the computed-torque controller.
This paper is organized in the following manner: section 2 consists of two subsections the first of which will show how the neural network is incorporated into the Dymola model of the computed-torque trajectory tracking control system for a 2-DOF robotic arm. The second subsection is devoted to explain the steepest descent error backpropagation algorithm used in the online training of the neural network. In section 3, the results of simulating the neural online torque compensator are analyzed and compared to those obtained by the computed torque control system. The paper ends with concluding remarks and future work in section 4.

II Neural Network Controller

A System Configuration

A two-layered neural network was designed and incorporated into the computed torque trajectory tracking control system of the 2-DOF robotic arm that was designed in [2] in order to learn the uncertainties involved in the system during its operation. The neural network contains two neurons in the output layer with a pure-linear activation function and three neurons in the hidden layer with a log-sigmoid activation function as shown in figure 1.

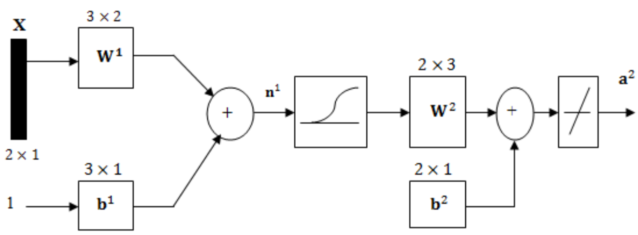

Figure 1: Structure of the neural network controller

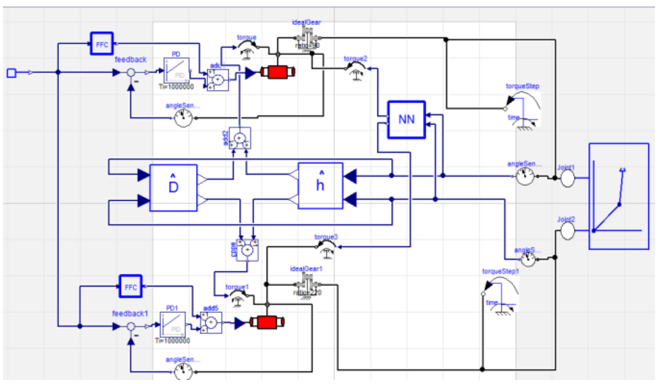

Figure 2: Dymola model of the trajectory tracking control system

Figure 2 shows the configuration of the computed torque trajectory tracking control system involving the neural network controller. The blocks $\widehat{\mathrm{D}}$ and $\widehat{\mathrm{H}}$ represent the estimated inertia matrix and the estimated centrifugal, corioilis and gravitational forces, respectively. These estimated matrices were computed using the nonlinear inverse dynamics equations of the 2-DOF robotic arm. The designed neural network controller receives the actual angular positions of the joints as input signals and generates two corresponding output signals to compensate for the torque difference caused by the structured and unstructured uncertainties.

B Learning Algorithm

As mentioned earlier, the designed neural network must be trained to learn the changing behavior of the system dynamics in the presence of both structured and unstructured uncertainties. The algorithm used for the training of the designed neural network in this paper is the steepest descent error backpropagation algorithm. This algorithm aims to continuously change the weights and biases of the neural network according to a specified error performance index function which is usually called the teaching signal of the network [6]. Figure 3 shows a flow chart summarizing the steps of the steepest descent error backpropagation algorithm to train the neural network.

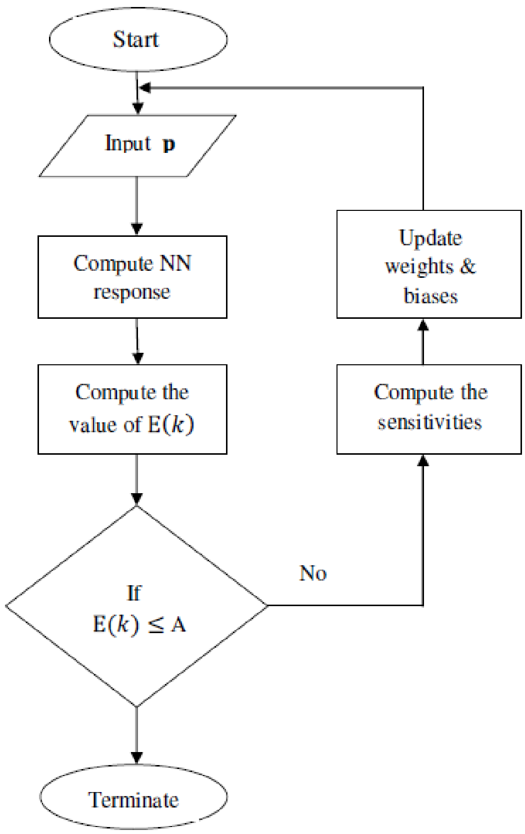

Figure 3: Flow chart of the steepest descent algorithm

Where $E(k)$ in the flow chart denotes the value of the error teaching signal at the k^{th} iteration. In this paper, we will use the teaching signal of the designed neural network to be the vector containing the outputs of the linear PD controllers of the joint actuating motors shown in figure 2. The weights and biases of the designed neural network controller are updated using the following rules:

$$\mathbf{W}^{m}(k+1)=\mathbf{W}^{m}(k)-\alpha\mathbf{S}^{\mathrm{m}}(\mathbf{a}^{m-1})^{\mathbf{T}} \qquad (1)$$

$$\mathbf{b}^{m}(k+1)=\mathbf{b}^{m}(k)-\alpha\mathbf{S}^{\mathrm{m}} \qquad (2)$$

Where $\mathbf{W}^{m}(k)$ and $\mathbf{b}^{m}(k)$ denote the weight matrix and bias vector of layer m of the network at the k^{th} iteration, respectively. $\mathbf{S}^{\mathrm{m}}$ is the sensitivity vector associated with layer m of the network and is computed using the following formulae:

$$\mathbf{S}^{\mathrm{M}}=-2\dot{\mathbf{F}}^{\mathrm{M}}(n^{\mathrm{M}})\,\mathbf{e}(k) \qquad (3)$$

$$\mathbf{S}^{m}=\dot{\mathbf{F}}^{m}(n^{m})(\mathbf{W}^{m+1})^{\mathbf{T}}\mathbf{S}^{m+1} \qquad (4)$$

Where (M) is the order number referring to the output layer of the neural network, and $\dot{\mathbf{F}}^{\mathrm{M}}(n^{\mathrm{M}})$ is the diagonal matrix containing the derivatives of the activation functions used in the output layer with respect to their corresponding net inputs. Likewise, $\dot{\mathbf{F}}^{m}(n^{m})$ denotes the diagonal matrix containing the derivatives of the activation functions used in layer m of the neural network.

III SIMULATION RESULTS

In order to test the effectiveness of the proposed neural network online torque compensator to compensate for both structured and unstructured uncertainties, the following simulation is conducted:

The structured uncertainty will be simulated by changing the values of the parameters of the motors and the robotic arm from their inaccurate values used in [2] to their accurate values mentioned in Table 1.

TABLE I

Parameters for the motor and the robotic arm

DC-motor parameters		Robotic arm parameters	
Inaccurate	Accurate	Inaccurate	Accurate
$R=3.5\Omega$	$R=4\Omega$	$a_1=0.25\ m$	$a_1=0.35m$
$L=1.3mH$	$L=2.4mH$	$a_2=0.15m$	$a_2=0.30m$
$K_b=K_m=0.047$	$K_b=K_m=0.053$	$m_1=1.95Kg$	$m_1=2.3Kg$
$J_m=3.3\times10^{-6}$	$J_m=4.2\times10^{-6}$	$m_2=0.93Kg$	$m_2=1.2Kg$
$B_m=0.0001$	$B_m=0.001$		

a constant disturbance torque of $\tau_d=2\ N.m$ on each joint at the time instant $t=5\ sec.$ as shown in figure (2). Figures (4) and (5) show the simulation results of the computed torque control system without the neural network compensator in the presence of the structured and unstructured uncertainties, respectively. It is clearly shown in these figures that the computed torque and the feed-forward (FFC) controllers were not capable of compensating for neither the parameter uncertainties nor the externally applied disturbance torques. This is because both the computed torque and the feed-forward (FFC) controllers are designed based on mathematical models involving the uncertain parameters of the robotic manipulator and the actuating motor. Figure 4 shows a steady-state error in the horizontal parts of the figure of about 0.5 rad for the shoulder joint motor position and about 0.04 rad for the elbow joint motor position. The steady state errors in the vertical parts are about 0.5 rad for the shoulder joint motor position and about 0.12 rad for the elbow joint motor position.

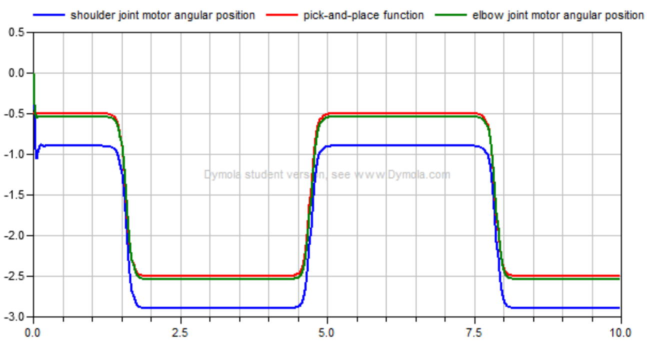

Figure 4: Simulation of computed torque control in the presence of structured uncertainties

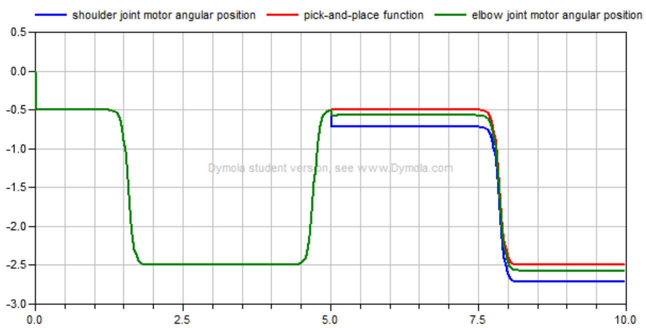

Figure 5: Simulation of computed torque control in the presence of unstructured uncertainties

Figure 6 shows the simulation results of the trajectory tracking control system involving the designed neural network online torque compensator in the presence of both structured and unstructured uncertainties. It can be observed from figure 6 that the designed neural network compensator has effectively learned both the uncertain and unknown factors involved in the system model and was capable of adaptively compensating for such uncertainties in order to maintain the desired trajectory tracking capability of the joint actuating motors which is clearly reflected by the results in figure 6.

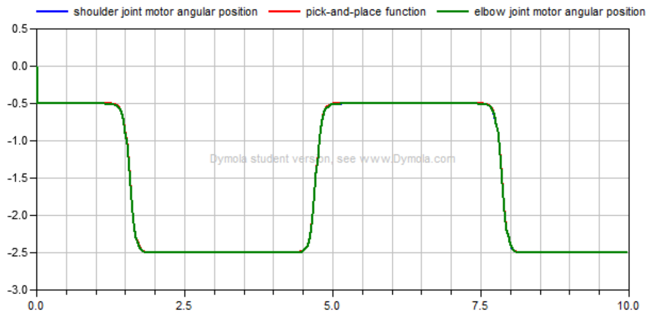

Figure 6: Simulation of trajectory tracking control system with NOTC

In figure 6, the steady state errors in both the horizontal and vertical parts are both equal to approximately 0.006 rad for both shoulder joint and elbow joint motor positions. Figures 7 and 8 show another indicator of the effectiveness of the proposed neural network controller by plotting the total disturbance torque applied on the inertia of each actuating motor before and after adding the neural network compensator, respectively.

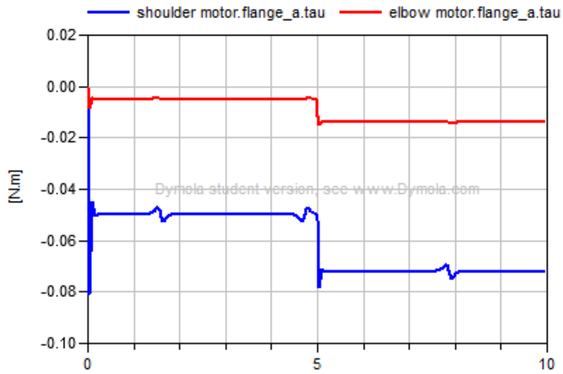

Figure 7: Disturance torques on the motors before adding the NOTC

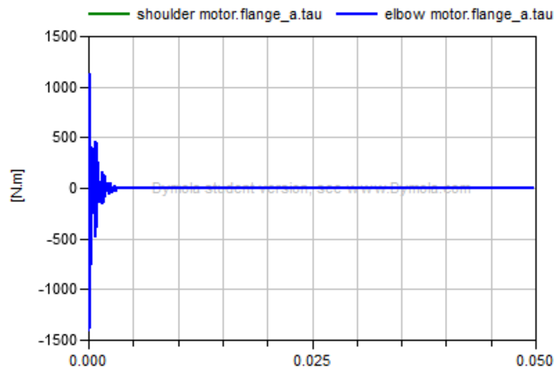

Figure 8: Disturance torques on the motors after adding the NOTC

In figure 7, it is shown that before the time instant $t = 5\ sec$, the total disturbance torques were about $-0.05\ N.m.$ for the shoulder joint motor and about $-0.005\ N.m.$ for the elbow joint motor. These disturbance torques were generated due to the structured uncertainties represented by the uncertain parameters of both arm and the actuating motors. At the time instant $t = 5\ sec$when the unstructured uncertainties began to have its effect, the disturbance torques applied on the actuating motors have increased to become about $-0.07\ N.m.$ for the shoulder joint motor and about $-0.014\ N.m.$ for the elbow joint motor. When the designed neural network online compensator was added to the system, it successfully learned the disturbance torque differences and hence eliminated the total disturbance torque applied on each joint driving motor within a period of 9 $mSec$ as shown in figure 8. Figures 9 and 10 show the sum of the outputs of the linear PD and feedforward (FFC) controllers for each motor control system before and after adding the neural network online torque compensator (NOTC), respectively. It is clearly seen in figure 9 that the PD and FFC controllers supply a non-descreasing control signal to each actuating motor. Before the time instant $t = 5\ sec$, the control signal supplied to the shoulder joint motor has the value of 3.8 and the control signal supplied to the elbow joint motor has the
value of 0.4. After the time instant $t = 5\ sec$, the PD and FFC controllers supplied a signal of about 5.4 to the shoulder joint motor and a signal of about 1.06 to the elbow joint motor.

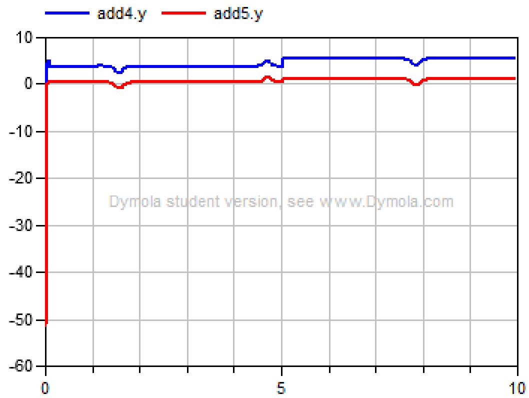

Figure 9: Sum of the outputs of PD and FFC controllers after adding the NOTC

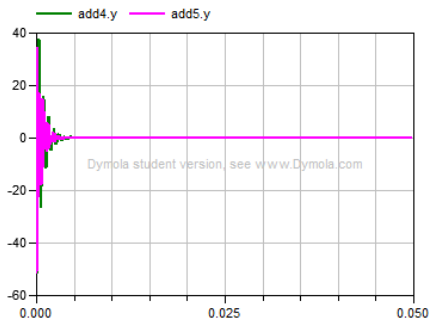

Figure 10: Sum of the outputs of PD and FFC controllers after adding the NOTC

These non-decreasing control signals supplied by the linear PD and FFC controllers to the actuating motors indicate their incapability of compensating for the structured and unstructured uncertainties in system model. However, after adding the neural online torque compensator, the control signals supplied by the linear PD and FFC controllers to the joint driving motors began decreasing dramatically until they both have reached the value of 0.006 within a period of about 9 $mSec$ as shown in figure 10.

IV Conclusion and future work

This paper presented an adaptive control strategy called neural online torque compensator (NOTC) to enhance the trajectory tracking capability of a robotic manipulator using a neural network. A two-layered neural network was designed and trained using the steepest descent error backpropagation algorithm to identify and compensate for the structured and unstructured uncertainties involved in a 2-DOF robotic manipulator trajectory tracking control system which is driven by a computed-torque controller. An object-oriented model was developed using modelica for the system components and was simulated on Dymola. The simulation results proved the superior effectiveness of the designed neural network controller in adaptively identifying and compensating for both structured and unstructured uncertainties. In this paper, we designed the neural network controller to handle the challenges faced by the computed-torque controller through training it to learn only the unknown factors and the uncertain parameters of the system model. Such a control system seems to be simple for a 2-DOF robotic manipulator. However, for higher DOF manipulators, the design of a computed torque controller requires the derivation of highly nonlinear and more complicated inverse dynamics equations. Therefore, we aim in a future paper to avoid this problem by designing a neural network controller which works as both a system identifier to learn the complex inverse dynamics of the manipulator and as a compensator for the structured and unstructured uncertainties involved in the system.

References

[1] Mark W. Spong, Seth Hutchinson, M. Vidyasagar, *Robot Modeling and Control.* First Edition, John Wiley and Sons, pp. 244-247.

[2] M. Alashi, H. Alaydi, I. Abu Hadrous, "Object-Oriented Modeling, Simulation, and Control of a Robotic Manipulator Using PD-Computed Torque Method", submitted for publishing.

[3] S. Okuma, A. Ishiguro, T. Furuhashi, Y. Uchikawa, "A Neural Network Compensator for Uncertainties of Robotic Manipulators", Proceedings of the 29th conference on Decision and Control, Honolulu, Hawaii, December 1990.

[4] J. J. Craig, "*Adaptive Control of Mechanical Manipulators,*"Addison Wesley Publishing Company, 1988.

[5] J.E. Slotine, W. Li, "Adaptive Manipulator Control: A Case Study". IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. AC-33, NO. 11, NOVEMBER 1988.

[6] Martin T. Hagan, Howard B. Demuth, Mark Beale, *Neural Network Design.* PWS Publishing Company.

[7] Suel Jung, "Neural Network Controllers for Robot Manipulators", PhD Dissertation, University of California, 1996.

[8] Xiang Li, Chien Chern Cheah, "Adaptive Neural Network Control of Robot Based on a Unified Objective Bound". IEEE TRANSACTIONS ON CONTROL SYSTEMS AND TECHNOLOGY, VOL. 22, NO. 3, APRIL 2014.