ON-LINE IDENTIFICATION OF A ROBOT MANIPULATOR USING NEURAL NETWORK WITH AN ADAPTIVE LEARNING RATE

Jasmin Velagic, Mujo Hebibovic and Bakir Lacevic

University of Sarajevo, Faculty of Electrical Engineering
Zmaja od Bosne bb, 71000 Sarajevo, Bosnia and Herzegovina
e-mails: jasmin.velagic@etf.unsa.ba, mujohebi@hotmail.com, bakir.lacevic@etf.unsa.ba

Abstract: This paper proposes an extension of neural network identification capabilities for on-line identification of a nonlinear closed-loop control system. The neural network (NN) is trained on-line using the backpropagation optimization algorithm with an adaptive learning rate. The optimization algorithm is performed at each sample time to compute the optimal control input. The results confirm the effectiveness of the proposed neural network based identification scheme and control architecture.

Keywords: fuzzy controller, neural network, on-line identification, robot dynamics

1. INTRODUCTION

In the last few years, a growing interest in the study of nonlinear systems in control theory has been observed. This interest stems from the need to give new solutions to some long standing necessities of automatic control; to work with more and more complex systems, to satisfy stricter design criteria, and to fulfill previous points with less and less a priori knowledge of the plant. The example of a nonlinear system is robot manipulator.

Robotic manipulator systems are nonlinear, high coupled, and time varying. Robots have to face many uncertainties in their dynamics, in particular structured uncertainties, which are caused by imprecision in the manipulator link properties, unknown loads, and unstructured one, such as nonlinear friction, disturbances, and the high-frequency part of the dynamics (Peng. and Woo, 2002). The control performance of the robotic manipulator is influenced by the mentioned uncertainties of the plant. From these reasons, identification and control of the robot manipulators are challenges.

Recently, much research has been done on applications of NNs for identification of nonlinear dynamic processes (Narenda and Pathasarathy, 1990; Nguyen and Widrow, 1990). These works are supported by two of the most important capabilities of neural networks; their ability to learn and their good performance for the approximation of nonlinear functions (Hornik, *et al*., 1989). At present, most of the works on system identification using neural networks are based on multilayer feedforward neural networks with backpropogation learning or more efficient variations of this algorithm (Narenda and Pathasarathy, 1990). It has been shown (Hornik, *et al*., 1990; Stinchombe and White, 1989) that a neural network with one hidden layer with an arbitrarily large number of neurons in the hidden layer can be approximate any continuous functions over a compact subnet of $\Re^n$. The objective of this paper is identification of a robot dynamics with NN based on backpropagation learning algorithm with adaptive learning rate in open and cloosed loop control systems. For this purpose, the supervised learning capabilities of NNs can be used for identifying process models from input/output data. These data are the training set for the network.

For controlling the robot manipulator in this paper we used fuzzy logic controller (FLC). The usage of FLC is justified from the following reasons: the dynamics of robot is modeled by nonlinear and coupled differential equations, it gives high flexibility, that is it has many degrees of freedom (shape and number of membership functions, aggregation methods, fuzzification and defuzzification methods, etc.). Also, fuzzy systems are suitable for uncertain and approximate reasoning, especially for the system with a mathematical model that is difficult to derive. Applying fuzzy logic to robotic control is currently a popular topic (Peng. and Woo, 2002; Byung and Woon, 2000; Neo and Er, 1996; Velagic, *et al.*, 2003).

2. NEURAL NETWORK IDENTIFICATION STRUCTURE UNDER CONTROL SYSTEM

The designed system consists of two major components: fuzzy control system and neural network identification structure of the plant (Fig. 1). To control the motion of the manipulator means to determine the n components of generalized joint torques that allow execution of a motion $\boldsymbol{q}(t)$ so that

$$\boldsymbol{q}(t) = \boldsymbol{q}_d(t)\,, \tag{1}$$

as close as possible, where $\boldsymbol{q}_d(t)$ denotes the vector of desired joint trajectory variables. Tracking control is needed to make each joint track a desired trajectory.
The manipulator we will be describing consist of a two links joined by rotary joints. Derivation of the dynamic model of a manipulator is based on the Lagrange formulation. It is used a FLC for joint position control. The output of position controller is the reference signal to be applied to the actuator. A dc servomotor used to actuate a robot joint. The outputs of control system are signals, which are directly measured (e.g., joint angles or velocity). At the same time on-line identification of robot manipulator is performed. In this section, we briefly describe components of the control system joint position. The FLC and NN plant model will be designed in Sections 3 and 4, respectively.

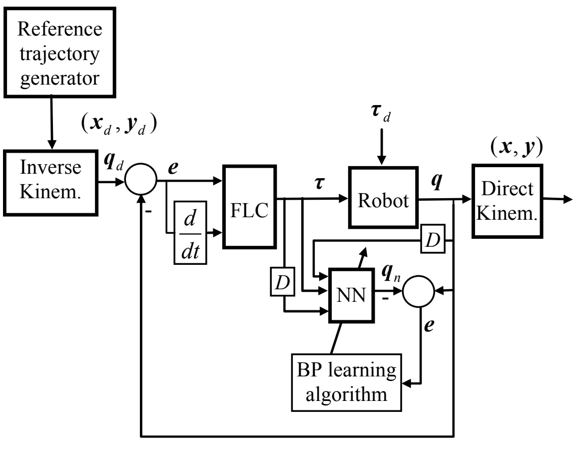

Fig. 1. The structure of the fuzzy/neural model-based robotic system.

2.1 *Dynamic model of robot manipulator and actuators*

A robot manipulator is defined as an open kinematics chain of rigid links. The dynamic model describing the motion of an n-joint robot is a set of n highly nonlinear and coupled differential equations, which relate the actuating, joint forces/torques with the joint positions, velocities, and accelerations respectively. Using Lagrange formulation, the equations of an n-degree-of-freedom manipulator can be written as (Sciavicco and Siciliano, 2000)

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}_r(\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{\tau}_d = \boldsymbol{\tau}\,, \tag{2}$$

where
$\boldsymbol{q} \in R^n$ is the generalized coordinates, $\boldsymbol{D}(\boldsymbol{q}) \in R^{n\times n}$ is the symmetric, positive definite inertia matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in R^n$ presents the centripetal and coriolis torques; $\boldsymbol{G}(\boldsymbol{q}) \in R^n$, $\boldsymbol{F}_r(\dot{\boldsymbol{q}}) \in R^n$, $\boldsymbol{\tau}_d \in R^n$ and $\boldsymbol{\tau} \in R^n$ represent the gravitational torques, friction, disturbance, and applied joint torques, respectively. We used the two-link planar manipulator shown in Fig. 2.

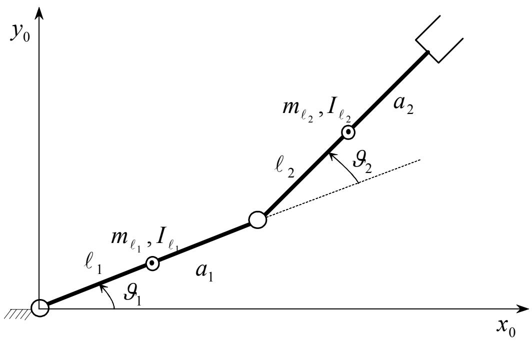

Fig. 2. Two-link planar arm.

The dynamic equation of a two-link planar robot manipulator is derived by using Euler-Lagrange method as follows:

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}\begin{bmatrix} \ddot{\vartheta}_1 \\ \ddot{\vartheta}_2 \end{bmatrix} = \begin{bmatrix} h\dot{\vartheta}_2 & h(\dot{\vartheta}_1 + \dot{\vartheta}_2) \\ -h\dot{\vartheta}_1 & 0 \end{bmatrix}\begin{bmatrix} \dot{\vartheta}_1 \\ \dot{\vartheta}_2 \end{bmatrix} + \begin{bmatrix} g_1 g \\ g_2 g \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{3}$$

where:

$$D_{11}(\vartheta_2) = I_{l_1} + m_{l_1} l_1^2 + k_{r_1}^2 I_{m_1} + I_{l_2} + m_{l_2}(a_1^2 + l_2^2 + 2a_1 l_2 c_2) + I_{m_2} + m_{m_2} a_1^2$$

$$D_{12}(\vartheta_2) = D_{21}(\vartheta_2) = I_{l_2} + m_{l_2}(l_2^2 + a_1 l_2 c_2) + k_{r_2} I_{m_2}$$

$$D_{22} = I_{l_2} + m_{l_2} + k_{r_2}^2 I_{m_2},\ h(\vartheta_2) = -m_{l_2} a_1 l_2 s_2,\ g = 9.81$$

$$g_1(\vartheta_1,\vartheta_2) = (m_{l_2} l_1 + m_{m_2} a_1 + m_{l_2} a_1) g c_1 + m_{l_2} l_2 g c_{12}$$

$$g_2(\vartheta_1,\vartheta_2) = m_{l_2} l_2 g c_{12},\ c_2 = \cos\vartheta_2,\ c_{12} = \cos(\vartheta_1 + \vartheta_2)$$

where l_1, l_2 are the distances of the centers of mass of the two links from the respective joint axes, m_{l_1}, m_{l_2} are the masses of two links and m_{m_1}, m_{m_2} are the masses of the rotors of the two joints motors. The moments of inertia with respect to the axes of the two rotors and the moments of inertia relative to the centers of mass of the two links are denoted by I_{m_1}, I_{m_2} and I_{l_1}, I_{l_2}, respectively. Also, it is assumed that the motors are located on the joints axes with centers of mass located at the origins of the respective frames.

2.2 Direct and inverse kinematics

A manipulator can be represented as an open kinematic chain of the links connected by means of revolute or prismatic joints which constitute the degrees of mobility of the structure. The resulting motion of the structure is obtained by composition of the elementary motions of each link with respect to the previous one. Direct kinematics problem describes the end effector position and orientation as a function of the joint variables of the mechanical structure with respect to a reference frame. The result of direct kinematics function is expressed by the homogeneous transformation matrix:

$$\boldsymbol{T}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{n}(\boldsymbol{q}) & \boldsymbol{s}(\boldsymbol{q}) & \boldsymbol{a}(\boldsymbol{q}) & \boldsymbol{p}(\boldsymbol{q}) \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

where $\boldsymbol{q}$ is the (nx1) vector of joint variables, $\boldsymbol{n}$, $\boldsymbol{s}$, $\boldsymbol{a}$ are the unit vectors of a frame attached to the end-effector, and $\boldsymbol{p}$ is the position vector of the origin of such frame with respect to the origin of the reference frame. The solution for two planar manipulator is:

$$\boldsymbol{T}(\boldsymbol{q}) = \begin{bmatrix} 0 & s_{12} & c_{12} & a_1c_1 + a_2c_{21} \\ 0 & -c_{12} & s_{12} & a_1s_1 + a_2s_{21} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

On the contrary, the inverse kinematics algorithm consists of the determination of joint variables corresponding to a given end-effector. The solution of the inverse kinematics problem for two-link planar manipulator (Fig. 3) is

$$\begin{aligned} \theta_1 &= \arccos(d/2) + \arccos(x/d) \\ \theta_2 &= -2\arccos(d/2) \end{aligned}, \tag{6}$$

where

$$d = \sqrt{x^2 + y^2} . \tag{7}$$

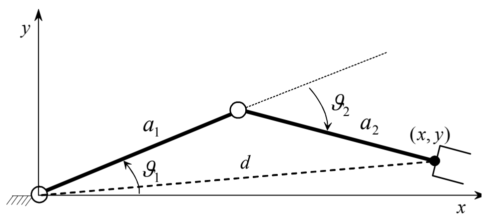

Fig. 3. Two-link planar arm for the inverse problem solving.

3. STRUCTURE OF THE FUZZY LOGIC CONTROLLER

In this paper we used the Mamdani fuzzy logic controller with two control inputs: error position of the links and their derivations (e, de), and the output is the joint torque derivative $d\tau/dt$. This controller has only seven rules and obtained good trajectory tracking performance in the both joint and operational spaces (Velagic, *et al*., 2003; Velagic and Hebibovic, 2004). Fuzzy controller is used for each link, separately, with the same inputs and output, membership functions and control rules. More details about this controller can be shown in (Velagic, *et al*., 2003; Velagic and Hebibovic, 2004).

4. NEURAL NETWORK DESIGN

Neural networks have been applied very successfully in the identification of dynamic systems. The universal approximation capabilities of the multilayer neural network make it popular choice for modeling nonlinear systems. For this purpose a neural network for representing the forward dynamics of the plant was trained.

4.1 Learning in Neural Network

The neural network used in this paper is composed of one hidden static layer with feedback (Fig. 4). The feedback signals are robot manipulator outputs (joint angles). This network is trained using the backpropagation algorithm. In the training phase the network is presented with a series of (input, answer) pairs. The network's current output o_k is compared with the desired input d_k, and the error is used to correct the weights so as to reduced the network's error on this input.

$$e(x) = \frac{1}{2}\sum_k (d_k - o_k)^2 . \tag{8}$$

The backpropagation is probably the most commonly used method for updating synaptic weights (Werbos, 1995). The propagation learning algorithm is briefly described with the following:

1. Weight initialization

Set all weights to some initial random value.

2. Activation calculation

Given an input d_k compute an output computation of the network y_k and compute the output of each hidden units of the network. That is, each individual unit computes an output $g(h_j)$, where $h_j = \sum_i w_{j,i}x_i$.

3. Weight training

Start at the output units and work backwards to the hidden layers recursively. Synaptic weights are update by

$$w_{j,i}(t+1) = w_{j,i}(t) + \eta\delta_j o_i , \tag{9}$$

where η is the learning rate, δ_j is the error gradient at unit j, and o_i is the output of unit i.

The gradient of the error δ_j is given by

a) For an output unit:

$$\delta_j = (d_j - o_j)g'(h_j) , \tag{10}$$

where d_j is the activation for output unit j that is required for output k.

b) For a hidden unit:

$$\delta_j = g'(h_j)\sum_k \delta_k w_{k,j} , \tag{11}$$

where δ_k is the error at unit k to which a connection points from hidden unit j.

4. Iterate until the units converge.

The input and output data of the neural network (Fig. 4) are driving torques (τ_1, τ_2) and joint angles (q_1, q_2).

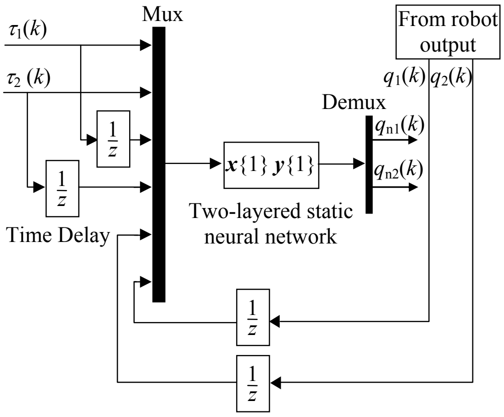

Fig. 4. The simulink model of dynamic NN.

4.2 On-line Identification of Process with Neural Networks

The main idea of this paper is design a neural network that emulates process (robot manipulator + actuators). For this purpose we used identification with neural network. The supervised learning capabilities of neural networks can be used for identifying process models from input/output data.

A unifying framework for neural networks that encompasses process identification concept is to view neural network training as a nonlinear optimization problem:

$$\min_{\mathbf{w}} J(\mathbf{w}) \cdot \tag{12}$$

That is, we need to find values for neural network parameters w (weight vector) for which some cost function $J(\mathbf{w})$ is minimized.

Let us assume that the process (robot) is described by the following nonlinear discrete time difference equation:

$$q(t) = f(q(t-1),...,q(t-n);\tau(t),...,\tau(t-m)), \tag{13}$$

where $q(t)$ is the process output at time t depends on the past n output values and on the past m values of the input τ. For identification plant model the neural network is used in the following form:

$$q_n(\mathbf{w},t) = f_w(q(t-1),...,q(t-n);\tau(t),...,\tau(t-m)), \tag{14}$$

where $n \geq m$ for physically realizable systems.

Here $f_w(\cdot)$ represents the nonlinear input-output map of the neural network which aproximates the process mapping $f(\cdot)$. Note that the input to the neural network includes the past values of the process output but not the past values of the network output, because the neural network has no feedback.

The training process for neural network nonparametric modeling can be expressed uniformly as the minimization of an error measure, typically sum-squared error, between the neural network output and the process output. If the sampled process data are collected over a period $[0,t]$, the cost function $J(\mathbf{w})$ in equation (12) is the following:

$$J(\mathbf{w},t) = \sum_{t=n}^{T}[q(t) - q_n(\mathbf{w},t)]^2 . \tag{15}$$

The minimization is effected with backpropagation algorithm through time, which is needed for the parallel identification form. For this purpose we define relationship between $\Delta J(\mathbf{w},t)$ and $J(\mathbf{w},t)$ as the relative factor $\chi(t)$:

$$\chi(t) = \frac{\Delta J(\mathbf{w},t)}{J(\mathbf{w},t)} = \frac{J(\mathbf{w},t)\text{-}J(\mathbf{w},t-1)}{J(\mathbf{w},t)} . \tag{16}$$

and then, we determine how to adjust learning rate term according to this relative factor χ. The adjustments of the learning rate is given as following:

$$\eta(t+1) = \eta(t)[1 - \mathrm{sgn}(\chi(t))\upsilon \cdot e^{-\chi(t)}], \quad \upsilon \in (0,1) . \tag{17}$$

The proposed algorithm is based on the conventional BP algorithm by employing an adaptive learning rate, where the learning rate is adjusted at each iteration (Eq. 9).

The algorithm proceeds as follows. First, we select the number of neurons in hidden and output layers, initial value of learning rate and the parameter υ. Then, the training process in the closed control loop is performed for various values of parameter υ, $\upsilon \in [0,1]$. We adopt the value of υ for which a satisfactory identification performance is achieved.

5. SIMULATION RESULTS

In the previous sections, we have described how to design fuzzy logic controller and neural network for robot trajectory tracking and on-line identification of robot manipulator. Efficiency of them will be demonstrated in this section. Consider two joint planar manipulator as shown in Fig. 2. The kinematics and inertial parameters of the manipulator and actuators are (Velagic, *et al*., 2003):

$$a_1 = a_2 = 1[m], m_{l_1} = m_{l_2} = 50[kg], I_{l_1} = I_{l_2} = 10[kg \cdot m^2]$$

$$m_{m_1} = m_{m_2} = 5[kg], I_{m_1} = I_{m_2} = 0.01[kg \cdot m^2], k_{r_1} = k_{r_2} = 100$$

The simulation study is organized into three parts. Part 5.1 focuses on fuzzy control of the robot manipulator. Tracking performance in both operational and joint spaces are considered . Parts 5.2 and 5.3 of the study are concerned with the on-line identification of the robot manipulator using feedforward neural network in open loop and closed loops, respectively. In both cases all disturbances are considered, instead of friction.

5.1 Fuzzy motion control system of robotic manipulator

The desired trajectory in operational space is given by way points, as shown in Fig. 5. Actual robotic trajectory achieved by fuzzy logic controller is presented in the same figure with a solid line. Transition between two intermediate way points is governed by trapezoidal joint velocity profile.

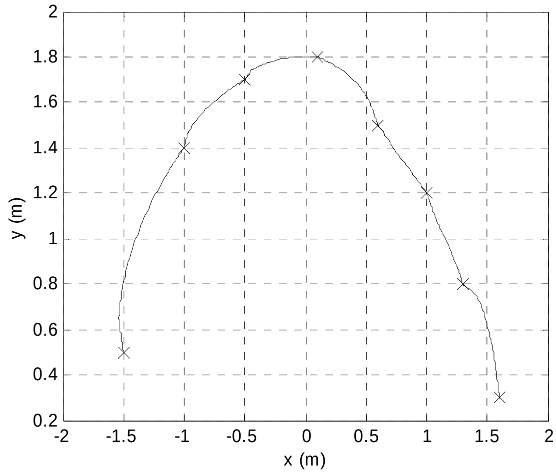

Fig. 5. Desired and actual position trajectory.

Corresponding joint position errors are negligible, which are presented in Fig. 6. Small deviations are presented in the case of joint 1, because the joint 1 senses the joint 2 as payload.

From the above figures it can be seen that a quite satisfactory control result is obtained, even though only seven rules are used to design the fuzzy control law. The proposed fuzzy controller is able to track well any paths. Also, this fuzzy controller demonstrates robustness in performance against adverse effects such as structured and unstructured uncertainties (e.g., robot inertia, Coriolis effect, and gravity). These effects influence the velocity, position, and acceleration of the robotic joints and thus negatively impact the controller's performance and the life span of the manipulator itself.

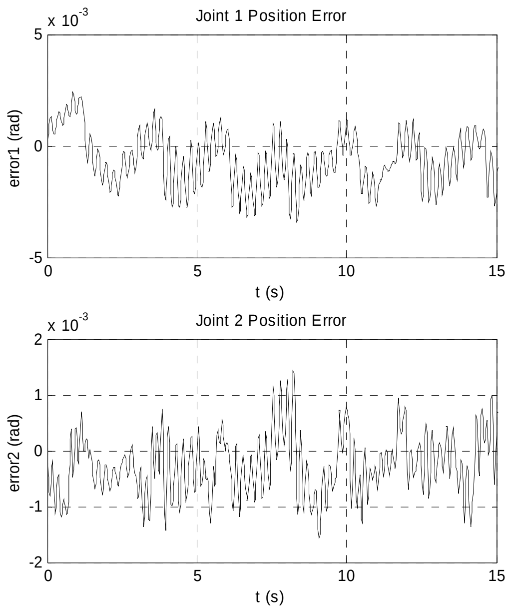

Fig. 6. Joints position errors.

5.2 On-line identification of manipulator using neural network

The identification of the robot manipulator is performed in on-line mode under uncontrolled plant (Fig. 7). The input signal is sinusoidal shaped. Neural network contains 10 tansig neurons in hidden layer and 5 purelin neurons in its output layer. The proposed algorithm started from the same initial learning rate η=0.04 for both layers (hidden and output). The satisfactory results were obtained with υ=0.8.

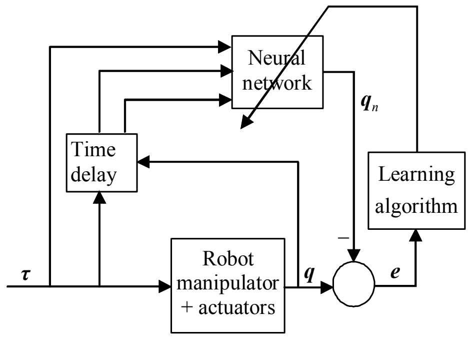

Fig. 7. Identification structure.

Comparison between actual and reference trajectory for both joints are shown in Figures 8 and 9.

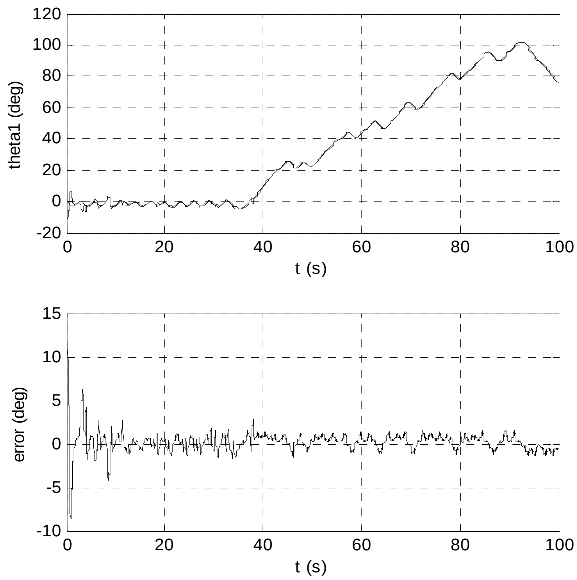

Fig. 8. On-line learning process of the joint angle 1. Solid line: actual values; dotted line: desired values.

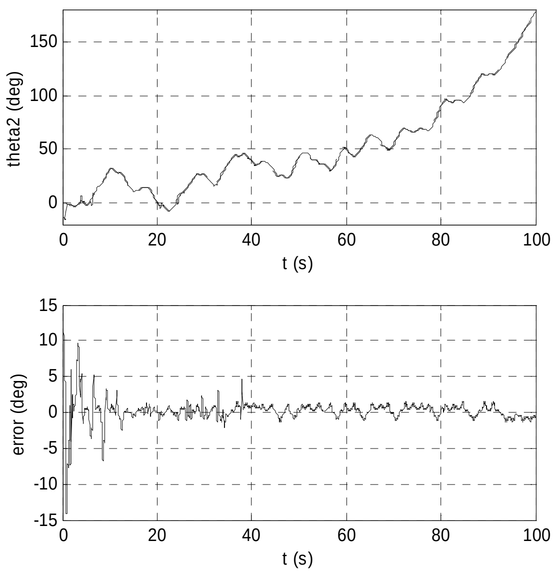

Fig. 9. On-line learning process of the joint angle 2. Solid line: actual values; dotted line: desired values.

The errors tracking have bigger values in the starting faze. The reason for this lies in initialization of the neural network with random number values. However, after short time (about 10 s) the neural network output follows the robot trajectory with small error. The neural network parameters (weights and biases) can be extracted during the on-line learning process.

5.3 On-line identification of manipulator using neural network in closed-loop control system

The identification structure under fuzzy closed control system is depicted in Fig. 10.

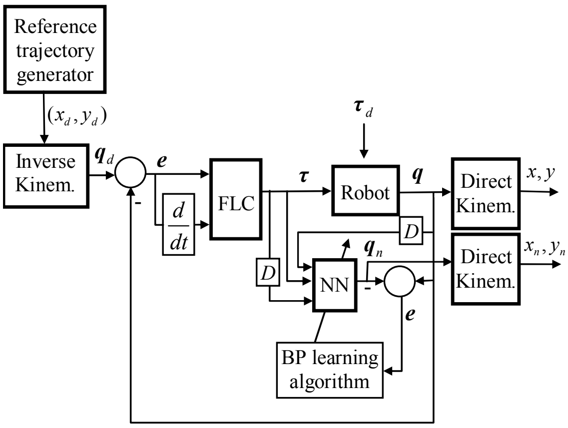

Fig.10. On-line Identification NN structure under closed loop control system.

On-line identification of the robot manipulator and actuators is performed by using the same neural network with the same parameters, which is mentioned described. The satisfactory results is also obtained with υ=0.8. The outputs of neural network plant model and robot manipulator are shown and compared in Fig. 11. From the obtained results can be concluded that the neural network mimics the robot and actuators complex system very well under control system in closed loop.

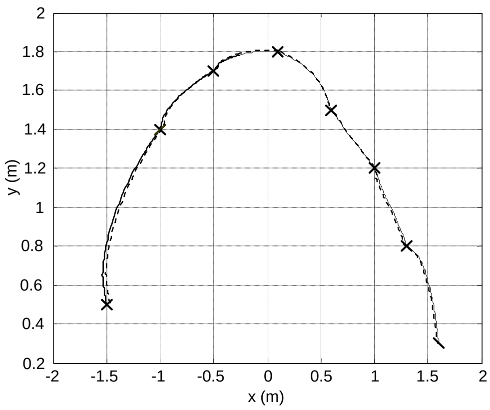

Fig. 11. Comparison among desired trajectory, neural network(--) and robot manipulator(–) outputs.

6. CONLUSIONS

This paper has successfully demonstrated the applications of neural network and fuzzy logic system to the identification and control of a robot manipulator. First, the fuzzy logic controller was used to robot position control. Then, a neural network was proposed to on-line identification of the robotic manipulator dynamics during the motion control. On-line parameter training is derived using the backpropagation method with adaptive learning rate. The effectiveness of the proposed hybrid identification and control scheme has been confirmed by simulated results. The conclusion is that the designed fuzzy controller and neural network are able to provide satisfactory performance for both trajectory tracking and identification capabilities.

REFERENCES

Peng, L. and P.Y. Woo (2002). Neural-Fuzzy Control System for Robotic Manipulators. *IEEE Control Systems*, **Vol. 22**, pp. 53-63.

Narenda, K.S. and K. Pathasarathy (1990). Identification and control of dynamic systems using neural network. *IEEE Transaction on Neural Networks*, **Vol. 1**, pp. 4-27.

Nguyen, D. and B. Widrow (1990). Neural Networks for Self- Learning Control Systems. *IEEE Control System Magazine*, **Vol. 10**, pp. 18-23.

Hornik, K., M. Stinchombe and H. White (1989). Multilayer Feedforward Networks are Universal Approximator. *Neural Networks*, **Vol. 2**, pp. 359-366.

Hornik, K., M. Stinchombe and H. White (1990). Universal Approximation of an Unknown mapping and its Derivatives Using Multilayer Feedforward Networks. *Neural Networks*, **Vol. 3**.

Stinchombe, M. and H. White (1989). Universal Approximation Using Feedforward Networks with Non-sigmoid Hidden Layer Activation Functions. In: *Proceeding of the International Joint Conference on Neural Networks*, Washington, USA, pp. 613-617.

Byung, K.Y. and C.H. Woon (2000). Adaptive Control of Robot Manipulator Using Fuzzy Compensator. *IEEE Transaction on Fuzzy Systems*, **Vol. 8**, pp. 186-199.

Neo, S. S. and M. J. Er (1996). Adaptive Fuzzy Controllers of a Robot Manipulator. *International Journal for System Science*, **Vol. 27**, pp. 519-532.

Velagic, J., A. Adzemovic and J. Ibrahimagic (2003). Switching Fuzzy Force/Position Control of Robot Manipulator. In: *Proceeding of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2003)*, Kobe, Japan, pp. 613-617.

Sciavicco, L. and B. Siciliano (2000). *Modeling and Control of Robot Manipulators*. Springer Verlag, Berlin.

Velagic, J. and M. Hebibovic (2004). Neuro-Fuzzy Architecture for Identification and Tracking Control of a Robot. In: *Proceeding of the 4th World Automation Congress, 5th International Symposium on Soft Computing for Industry (ISSCI2004)*, Sevilla, Spain, Paper no. ISSCI-032 (pp. 1-9).

Werbos, P. (1995). Backpropagation: Basics and New Developments. In: *The Handbook of Brain Theory and Neural Networks*, MIT Press, Cambridge, pp. 134-139.