

Neural Network Based Trajectory Planning for Tip Tracking of a Two-Link Flexible Robot Manipulator

Gülay Öke and Yorgo İstefanopulos

Department of Electrical-Electronic Engineering, Boğaziçi University

Bebek, İstanbul, 34342, Turkey

e-mail: oke@boun.edu.tr, istef@boun.edu.tr

Abstract—The main problem in the control of flexible manipulators is to provide precise tip tracking in the operational space. Even if joint angles are controlled successfully, the end effector of the manipulator deviates from the desired position because of link deflections and vibrations. In this study, a neural network based trajectory planning method is applied to calculate modifications in the command values of joint angles for tracking control problem, so that the position error of the tip of the manipulator in the operational space is minimized. A PD controller is applied for the joint angle control, where the reference inputs are the modified values calculated by the neural network. Simulations are performed to evaluate the performance of the trajectory planning method and the control procedure.

Index terms—flexible manipulator, neural network, trajectory planning.

I. INTRODUCTION

Research on flexible manipulators is being carried out for the last two decades, because of the application potential they offer due to the advantages they have over rigid robots. Flexible manipulators have a higher ratio of the load to arm weight. They require less power to produce the same acceleration as rigid link manipulators which have the same load carrying capacity, therefore smaller and cheaper actuators are sufficient. On the other hand, modelling and control of flexible link manipulators are more difficult than for rigid links. The resulting model is a distributed parameter system of infinite dimensions and it is non-minimum phase. The number of control inputs is less than the number of variables to be controlled since the actuators are colocated at the joints. This means that the link deflections can be suppressed only indirectly. The gross motion and the deformational behaviour of the link interact, causing the dynamical equations to be complicated and highly nonlinear. As the spatial boundary conditions of the links change, the characteristic frequencies and modes are modified.

In the technical literature, most of the work on control of flexible manipulators aims at suppression of link deflections, while controlling joint angles simultaneously. Link deflections cannot be controlled directly because of the colocated nature of the actuators, leading to imperfect vibration suppression. Another possible approach is to calculate new joint angles, to make the tip of the manipulator track the desired trajectory in the operational space, irrespective of link deflections.

In this study, a neural network is used to calculate an incremental modification on the command values of joint angles, so that the error of the tip position in the operational space is minimized. A neural network is utilized to make use of the learning properties of intelligent structures. With the modified trajectory as the new command trajectory in the joint space, a PD controller is applied for joint angle control. Simulations are performed to illustrate the performance of the method on the joint motion, deflections, tip motion and errors in operational space.

II. ROBOT DYNAMICS AND KINEMATICS

The manipulator which is to be controlled in this study is a planar, two-link flexible manipulator. The model has been derived by De Luca and Siciliano [1] and a sketch of it is given in figure 1. Rotary joints are subject only to bending deformation in the plane of motion; torsional and gravitational effects are neglected.

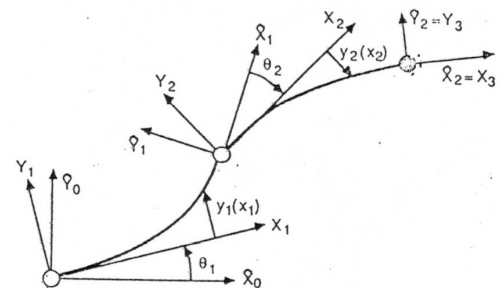


FIGURE 1. Two-link, planar flexible manipulator

For the two-link flexible manipulator clamped at the origin, depicted in figure 1, the following coordinate frames are established: The inertial frame, (\hat{X}_0, \hat{Y}_0) ; the rigid body moving frame associated to link i , (X_i, Y_i) and the flexible body moving frame associated to link i , (\hat{X}_i, \hat{Y}_i) . The rigid motion is described by the joint angles θ_i and $y_i(x_i)$ denotes the transversal deflection of link i at abscissa x_i . ($0 \leq x_i \leq l_i$; where l_i is the length of link i).

The closed-form dynamic equations of the robot can be obtained by calculating the kinetic and potential energies and applying Lagrange equations:

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{K}\mathbf{q} = \mathbf{Q}\mathbf{u} \quad (1)$$

where $\mathbf{B}(\mathbf{q})$ is the positive definite symmetric inertia matrix, $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$ is the vector of Coriolis and centrifugal forces, \mathbf{K} is the stiffness matrix, \mathbf{Q} is the input weighting matrix, \mathbf{u} is the n -vector of joint (actuator) torques.

$\mathbf{q} = (\theta_1, \dots, \theta_n, \delta_{1,1}, \dots, \delta_{1,m_1}, \dots, \delta_{n,1}, \dots, \delta_{n,m_n})^T$ describes the N -vector of generalized coordinates ($N = n + \sum_i m_i$). Input weighting matrix \mathbf{Q} , is in the form $[\mathbf{I}_{n \times n} \quad \mathbf{0}_{n \times (N-n)}]^T$ due to the clamped link assumption. This form of the \mathbf{Q} matrix implies that inputs can be applied through actuators and they affect only the joint angles directly but not the deformation of the links.

To describe the tip position of the robot in operational space, the kinematic equations of the robot are utilized. There are two kinds of rotation matrices for the flexible robot manipulator: The joint (rigid) rotation matrix \mathbf{R}_i , and the rotation matrix of the flexible link at the end-point \mathbf{E}_i .

$$\mathbf{R}_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \quad (2)$$

$$\mathbf{E}_i = \begin{bmatrix} 1 & -y'_{ie} \\ y'_{ie} & 1 \end{bmatrix} \quad (3)$$

In these formulas, $y'_{ie} = \left(\frac{\partial y_i}{\partial x_i} \right)_{x_i=l_i}$ and the linear approximation $\arctan(y'_{ie}) \cong y'_{ie}$ is valid for small deflections. \mathbf{T}_i denotes the global transformation matrix from (\hat{X}_0, \hat{Y}_0) to (X_i, Y_i) , where i represents the i^{th} link, and obeys the following recursive equation:

$$\mathbf{T}_i = \mathbf{T}_{i-1} \mathbf{E}_{i-1} \mathbf{R}_i = \hat{\mathbf{T}}_{i-1} \mathbf{R}_i \quad \hat{\mathbf{T}}_0 = \mathbf{I} \quad (4)$$

By using (2)-(4), the kinematics of any point along the arm can be fully characterized with respect to the base frame. For the two-link, planar, flexible arm, the position of the tip of the robot in the operational space is given by:

$$\mathbf{p} = \mathbf{T}_1 \begin{bmatrix} l_1 \\ y_1(l_1) \end{bmatrix} + \mathbf{T}_1 \mathbf{E}_1 \mathbf{T}_2 \begin{bmatrix} l_2 \\ y_2(l_2) \end{bmatrix} \quad (5)$$

where l_1 and l_2 are the lengths of shoulder and elbow links respectively. The \mathbf{p} vector denotes the x and y coordinates of the tip of the manipulator:

$$\mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix} \quad (6)$$

Using (5), the x and y coordinates of the tip of the manipulator are calculated as follows:

$$x = l_1 \cos \theta_1 - y_1 \sin \theta_1 + l_2 \cos(\theta_1 - \theta_2) - l_2 y'_{1e} \sin(\theta_1 - \theta_2) - y_2 \sin(\theta_1 - \theta_2) - y_2 y'_{1e} \cos(\theta_1 - \theta_2) \quad (7)$$

$$y = l_1 \sin \theta_1 + y_1 \cos \theta_1 + l_2 \sin(\theta_1 - \theta_2) + l_2 y'_{1e} \cos(\theta_1 - \theta_2) + y_2 \cos(\theta_1 - \theta_2) - y_2 y'_{1e} \sin(\theta_1 - \theta_2) \quad (8)$$

It is obvious from (7) and (8) that, the direct kinematic equations of the two-link flexible manipulator are highly nonlinear functions of several variables and it is not an easy task to obtain inverse kinematic equations out of them. Therefore, it is logical to utilize a computationally intelligent agent for control in the operational space.

III. NEURAL NETWORK BASED TRAJECTORY PLANNING

The main difficulty in the control of flexible link manipulators is that because of the colocated nature of the actuators, the vibrations cannot be controlled directly. Even if the joint angles are controlled perfectly, the tip of the manipulator will not follow the desired trajectory in the operational space, because of the deflections and vibrations. A possible approach in the control of flexible manipulators is to leave the link deflections as they are and to calculate new joint angles for a precise command of manipulator tip in the operational space. Some iteration based techniques utilizing this method can be found in [2],[3], [4] and [5]. In [6], a gradient descent based trajectory planning method has been proposed for the regulation of two-link flexible robotic arm. In [7], a neural network based technique has been discussed for the regulation of flexible log cranes. The modification of the reference trajectory for flexible manipulators when the tip is required to track a trajectory in the operational space is therefore a challenging problem. In [8], neural network inverse control techniques are applied for trajectory tracking of a PD controlled rigid robot manipulator.

In this study, a neural network based scheme is applied to modify the reference trajectories of the joint angles of the two-link flexible robot manipulator depicted in figure 1, for the tracking control problem. The control scheme is shown in figure 2. The neural network operates online to calculate the incremental change in the command values of joint angles, while in the control loop a PD controller is applied to the manipulator.

The reference trajectory for the joint angles can be decomposed into two parts:

$$\theta_{ref} = \theta_{ref1} + \theta_{ref2} \quad (9)$$

In figure 2, \mathbf{p}_{des} denotes the desired trajectory for the tip of the manipulator in the operational space, θ_{ref1} is calculated from \mathbf{p}_{des} using inverse kinematic equations of a two-link, planar rigid robot. θ_{ref2} is calculated by the neural network utilizing the information of error in the operational space.

$$\theta_{ref2} = NN(\mathbf{W}, \mathbf{e}_p) \quad (10)$$

$$\mathbf{e}_p = \mathbf{p}_{ref} - \mathbf{p} = \begin{bmatrix} e_x \\ e_y \end{bmatrix} = \begin{bmatrix} x_{des} - x \\ y_{des} - y \end{bmatrix} \quad (11)$$

\mathbf{W} is the vector of neural network weights and \mathbf{e}_p denotes the error of the tip in the Cartesian coordinates. Separate neural networks are designed for each joint angle and they are composed of three layers; the input, hidden and output layer. Hyperbolic tangent functions are used as activation functions in the hidden layer, and linear functions are utilized in the output layer. An auxiliary variable, e_{xy} , is defined to measure the error between the actual and desired positions of the tip of the robot manipulator.

$$e_{xy} = \frac{1}{2} (e_x^2 + e_y^2) \quad (12)$$

Error backpropagation method is used for training the neural networks and the weights are updated according to (13) in order to minimize the cost function given in (14).

$$\Delta \mathbf{W} = -\eta \frac{\partial J}{\partial \mathbf{W}} \quad (13)$$

$$J = \frac{1}{2} e_{xy}^2 \quad (14)$$

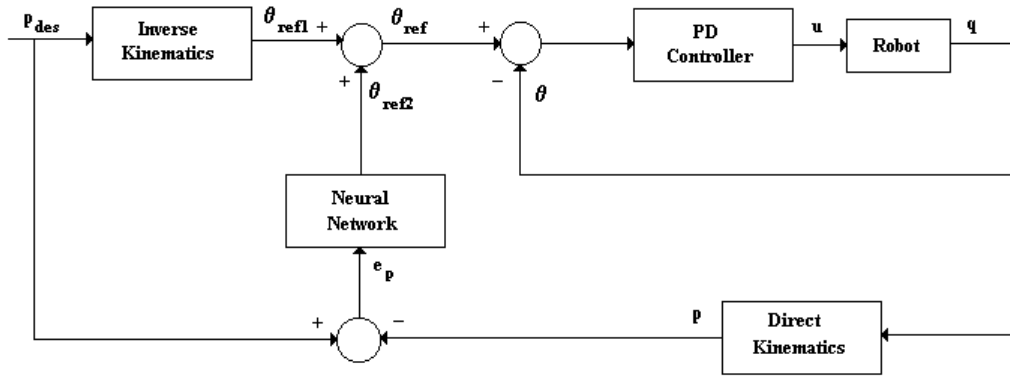


FIGURE 2. The control scheme

IV. PD CONTROLLER

The incremental changes to be made in the reference values of the joint angles to minimize the tip error in Cartesian coordinates are calculated in the neural networks as described in section III, and added to the value obtained from inverse kinematic equations, hence the final value of the reference angles are obtained. In the control loop, a PD controller is utilized to obtain good tracking in the joint space. The actuator torques to be applied to each joint is given by:

$$\tau_i = -k_{p_i} \tilde{\theta}_i(t) - k_{v_i} \dot{\tilde{\theta}}_i(t) \quad (15)$$

In equation (15), $\tilde{\theta}_i(t)$ and $\dot{\tilde{\theta}}_i(t)$ represent the error in the joint angle and velocity for the i^{th} link. k_{p_i} and k_{v_i} are the proportional and derivative gains, respectively. In the simulations, the controller gains are chosen as: $k_{p_1} = 5$, $k_{v_1} = 3$, $k_{p_2} = 1.2$, $k_{v_2} = 0.8$.

V. SIMULATION RESULTS

Simulations are performed on the two-link, planar, flexible manipulator depicted in figure 1, to test the performance of the trajectory planning method described in section III and the PD controller explained in section IV. It is required that the tip of the manipulator tracks a rectangular trajectory in the operational space. Figures 3-8 illustrate the results of the simulations. In figure 3, the modification in reference joint angles obtained from neural networks are shown. In figure 4, the reference and

actual values of joint angles are depicted, figure 5 shows the deflection of each link. Figure 6 illustrates the comparison of the error of the tip of the manipulator in the operational space, with and without the incremental change calculated by the neural network. Similarly, figure 7 depicts the auxiliary variable e_{xy} , given by (14), and figure 8 shows the tip of the manipulator in the operational space, with and without the incremental change calculated by the neural network.

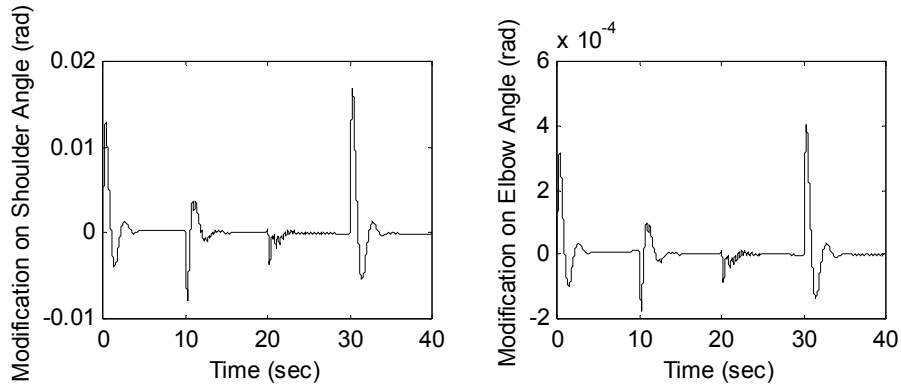


FIGURE 3. The modification on reference values of joint angles calculated by neural networks

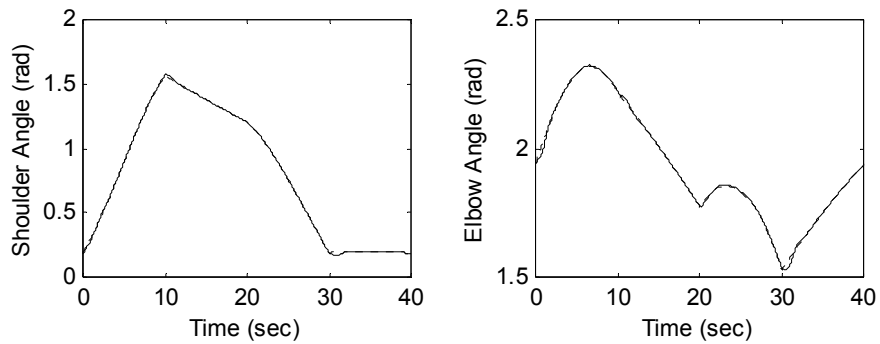


FIGURE 4. Reference (dotted lines) and actual (solid lines) angles for the shoulder and elbow angles

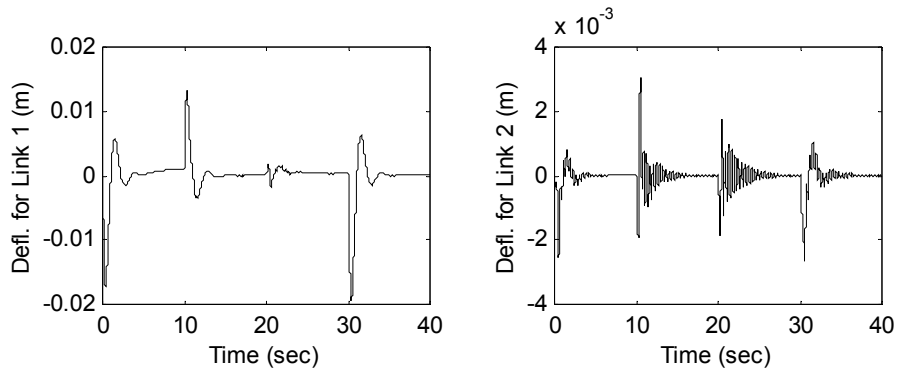


FIGURE 5. Deflection of the shoulder link and the elbow link

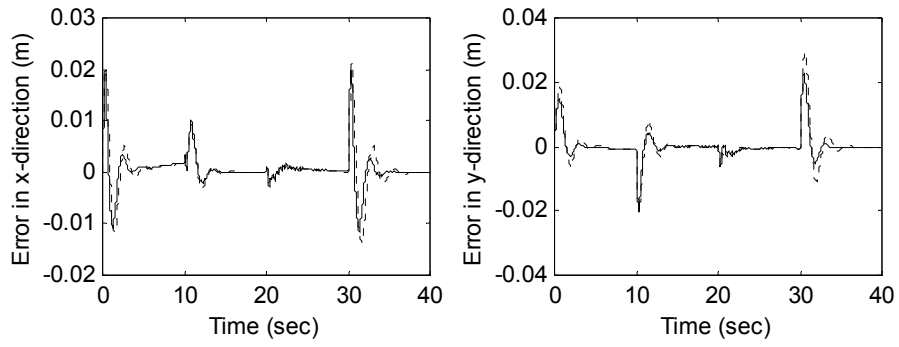


FIGURE 6. The errors in x- and y-coordinates of the tip of the manipulator in the operational space with (solid line) and without (dotted line) modification of reference joint angles

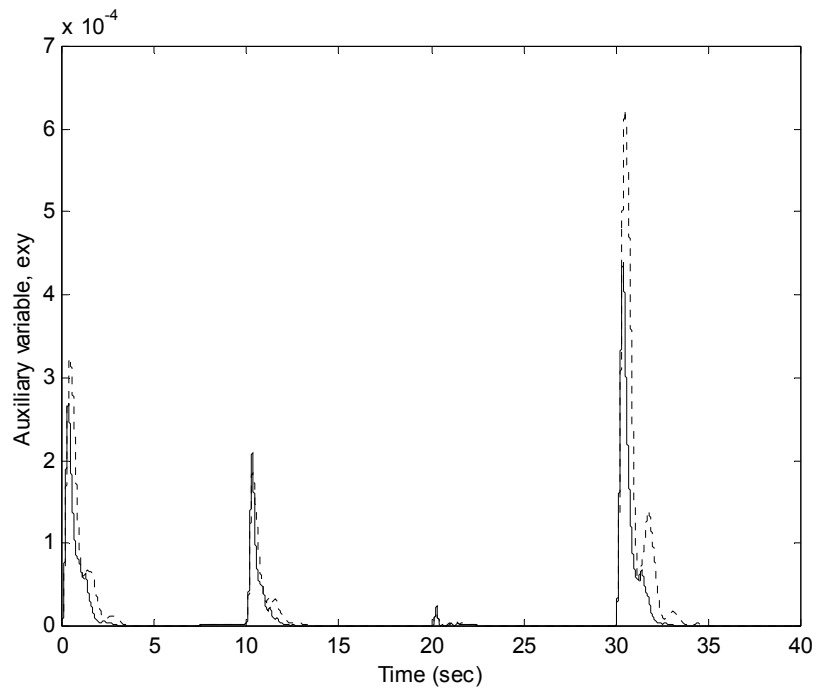


FIGURE 7. The auxiliary variable e_{xy} with (solid line) and without (dotted line) modification of reference joint angles

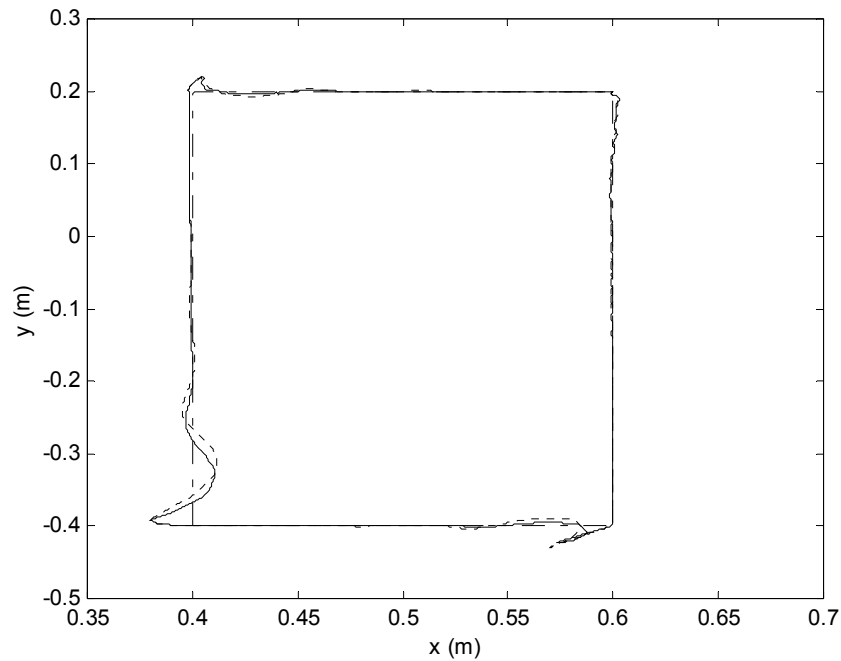


FIGURE 8. Desired trajectory (dash-dotted line) and the position of the tip of the manipulator in the operational space with (solid line) and without (dotted line) modification of reference joint angles

VI. CONCLUSIONS

In this study, a neural network based trajectory planning method is applied to a planar, two-link flexible manipulator to provide a more precise tracking for the tip trajectory in the operational space. The reference values of joint angles are calculated by inverse kinematic equations of rigid manipulators. Then an incremental change for these joint angles is computed by a neural network which takes the errors in x- and y- coordinates of the tip in the operational space as inputs and updates its weights in order to minimize the error in the operational space. The incremental value is added to the reference value of the joint angle. In the control loop, the control of the joint angles is provided by a PD controller. As we can observe from the simulation results, in comparison to the case where there is no reference joint angle compensation for link flexibility, there is really a pronounced improvement in the operational space tracking performance of the tip of the manipulator.

REFERENCES

- [1] De Luca A, Siciliano B, "Closed-Form Dynamic Model of Planar Multilink Lightweight Robots," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 21, No. 4 s. 826-839, 1991.
- [2] Asada H, Ma ZD, Tokumaru H., "Inverse Dynamics of Flexible Robot Arms: Modelling and Computation for Trajectory Control," *ASME Journal of Dynamic Systems Measurement and Control*, Vol. 112, s. 177-185, 1990.
- [3] Bayo E, Papadopoulos P, Stubbe J, Serna MA "Inverse Dynamics and Kinematics of Multi-Link Elastic Robots: An Iterative Frequency Approach," *International Journal of Robotics Research*, Vol. 8, No. 6, s. 49-62, 1989.
- [4] Sarkar PK, Yamamoto M, Mohri A, "On the Trajectory Planning of a Planar Elastic Manipulator Under Gravity," *IEEE Transactions on Robotics and Automation*, Vol. 15, No. 2, s. 357-362, 1999.
- [5] Ubertini F, "A Contribution to the Analysis of Flexible Link Systems," *International Journal of Solids and Structures*, Vol. 37, s.969-990, 2000.
- [6] Öke G, İstefanopulos Y, "Gradient-Descent Based Trajectory Planning for Regulation of a Two-Link Flexible Robotic Arm" *2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, (AIM'01) Proceedings*, Vol.II, s.948-952.
- [7] Rouvinen A, Handroos H, "Deflection Compensation of a Flexible Hydraulic Manipulator Utilizing Neural Networks," *Mechatronics*, Vol. 7, No. 4, s. 355-368,1997.
- [8] Jung S., Hsia T.C., "Neural Network Inverse Control Techniques for PD Controlled Robot Manipulator", *Robotica*, Vol.18, pp. 305-314, 2000.