ADAPTIVE NEURO-FUZZY CONTROLLER FOR HYBRID POSITION/FORCE CONTROL OF ROBOTIC MANIPULATORS

Arash Fanaei **Mohammad Farrokhi**

Faculty of Electrical Engineering
Iran University of Science and Technology
Tehran16844 - Iran

Abstract: In this paper, an adaptive control method for hybrid position/force control of robot manipulators, based on neuro-fuzzy modelling, is presented. Also, an adaptive neuro-fuzzy compensator compensates the friction force between the endeffector and the surface of the object. Due to the adaptive neuro-fuzzy modelling, the proposed controller is independent of the robot dynamics. Also, the stability of the controller is guaranteed, since the adaptation law is based on Lyapunov theory. The simulation results show good performance of the proposed controller as compared with other conventional control schemes such as computed torque method.

Keywords: Manipulators – Robot control – Force Control – Position control – Fuzzy control – Adaptive control – Friction – Compensation

1. INTRODUCTION

When the endeffector of a robot manipulator comes in contact with a surface with friction and unknown stiffness, there must be force control in addition to the position control of the endeffector. This task is well known as the position/force control of robot manipulators. Two force control methods have been extensively studied by many researchers in the past two decades: 1) the hybrid position/force control method (Hsu and Fu, 2000) and 2) the impedance control method (Hogan, 1985). In both methods a precise control scheme is necessary in order to move the endeffector on the right path, exerting the correct amount of force on the object.

In recent years, intelligent methods and algorithms, like neural networks and fuzzy logic have been employed for position/force control of robot manipulators. Hsu and Fu (2000) have proposed an adaptive fuzzy hybrid position/force controller, in which the adaptive parameters are the centers of the membership functions in the tally part of the fuzzy IF-THEN rules, but the friction force has not been taken into account. Kiguchi et al. (1999) have also used a neuro-fuzzy method for position/force control, but with a simple and static model for the friction between the endeffector and the contact surface. The friction compensation has been performed with one neuron, whose output is multiplied to a weight, which is the coefficient of the static friction of the surface. But the weight is not trained adaptively. The proposed controller by Xiao et al. (2000) has the advantage of handling the position/force control in an uncalibrated environment. But, the controller needs an optical sensors and a visual system to observe the conditions of the surface.

In this paper an adaptive neuro-fuzzy controller with an adaptive neuro-fuzzy friction compensator is proposed for hybrid position/force control of robot manipulators. Thanks to the adaptive neuro-fuzzy modelling, both for the controller and the friction compensator, the proposed method is independent of the robot dynamics as well as the conditions of the environment. The main advantage of the proposed control method is that the adaptation law is based on the Lyapunov stability theory, which guarantees the stability of the controller. Although the simulations are performed on a robot manipulator with three degrees of freedom with revolute joints, the proposed controller can be extended to robot manipulators with more degrees of freedom and different kind of joints. Moreover, the structure of the controller and the compensator is very simple, making it a very fast and appropriate method for different applications of robot manipulators. The simulation results show good performance of the proposed method as compared with other conventional control methods such as computed torque method.

2. MANIPULATOR DYNAMICS AND THE MODEL OF CONTACT SURFACE

2.1 Manipulator Dynamics

The dynamic equation of a robot manipulator can be expressed in Cartesian space as (Lewis et al., 1993):

$$\mathbf{M_x}(\mathbf{x})\ddot{\mathbf{x}} + \mathbf{C_x}(\mathbf{x},\dot{\mathbf{x}}) + \mathbf{G_x}(\mathbf{x}) + \mathbf{D_x}(\dot{\mathbf{x}}) = \mathbf{f} - \mathbf{f_e} - \mathbf{f_s} \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^{n\times 1}$ is the position vector of the endeffector, $\mathbf{M_x}(\mathbf{x}) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $\mathbf{C_x}(\mathbf{x},\dot{\mathbf{x}}) \in \mathbb{R}^{n\times 1}$ is the centrifugal and coriolis force vector, $\mathbf{G_x}(\mathbf{x}) \in \mathbb{R}^{n\times 1}$ is the gravity force vector, $\mathbf{D_x}(\dot{\mathbf{x}}) \in \mathbb{R}^{n\times 1}$ is the vector for joint friction force of the robot arms, $\mathbf{f} \in \mathbb{R}^{n\times 1}$ is the required force vector to move the endeffector, $\mathbf{f_e} \in \mathbb{R}^{n\times 1}$ is the applied force vector to the surface by the endeffector, and $\mathbf{f_s} \in \mathbb{R}^{n\times 1}$ is the force vector for the surface friction in Cartesian coordinate. Equation (1) can also be written as

$$\mathbf{M_x}(\mathbf{x})\ddot{\mathbf{x}} + \mathbf{C_{xm}}(\mathbf{x},\dot{\mathbf{x}})\dot{\mathbf{x}} + \mathbf{G_x}(\mathbf{x}) + \mathbf{D_x}(\dot{\mathbf{x}}) = \mathbf{f} - \mathbf{f_e} - \mathbf{f_s} \quad (2)$$

where $\mathbf{C_{xm}} \in \mathbb{R}^{n\times n}$. The robot manipulator used in this paper has 3DOF, consisting of waist, shoulder and elbow.

2.2 The Characteristics of the Contact Surface

In order to characterize the surface, which comes in contact with the endeffector, two mathematical models are considered here: the perpendicular model and the tangent model.

The Perpendicular Model. The mathematical model in the perpendicular direction of the contact surface is a spring model (Kiguchi et al. 1999). If the external applied force to the surface is shown with f_e, then $f_e = k_e(x_e - x)$, where k_e is the stiffness coefficient of the surface, in N/m, and x and x_e are the position of the endeffector and the surface, respectively.

The Tangent Model. When an object is moving on a surface, there will be friction force between the object and the surface. This force is due to the applied perpendicular force to the surface and is in the opposite direction of the movement. There are many models available in the literature to represent friction forces. Bona and Indri (1995) have shown that a comprehensive mathematical model for friction force f_s can be given in terms of applied perpendicular force f_e as

$$f_s = \left[g_0 + g_1 e^{-b_1|v|} + g_2(1-e^{-b_2|v|})\right]|f_e|\,\mathrm{sgn}(v) \quad (3)$$

where v is the speed of the object, $(g_0+g_1)|f_e|$ is the stiction friction, $(g_0+g_2)|f_e|$ is the dynamic friction at higher speeds, $(g_i = a_i f_e, i = 0,1,2)$, b_1 shows the damping rate of friction at lower speeds and also is an indication of negative slope in the stiction region (Bona and Indri, 1995), and b_2 specifies the ascending rate of friction at higher speeds.

3. ADAPTIVE NEURO-FUZZY CONTROLLER

The task of the controller is to move the endeffector along the desired path $\mathbf{x}_d = [x_{d1}\ \ x_{d2}\ \ x_{d3}]^T$, on a surface in the $x_1 - x_2$ plane, and since the force is applied to the surface along the x_3 axis, then $f_e = k_e(x_{e3} - x_3)$. Therefore, the desired path and the desired force can be defined as

$$\mathbf{x}_\mathrm{d} = [x_{d1}\ \ x_{d2}\ \ f_d/k_e]^T, \mathbf{f}_\mathrm{d} = [0\ \ 0\ \ f_e]^T \quad (4)$$

The errors between the desired and the actual vectors for the position, velocity and acceleration of the endeffector are

$$\mathbf{e} = \mathbf{x}_d - \mathbf{x} = \begin{bmatrix} x_{d1} - x_1 & x_{d2} - x_2 & \dfrac{f_d - f_e}{k_e}\end{bmatrix}^T \quad (5)$$

$$\dot{\mathbf{e}} = \dot{\mathbf{x}}_\mathrm{d} - \dot{\mathbf{x}}, \ddot{\mathbf{e}} = \ddot{\mathbf{x}}_\mathrm{d} - \ddot{\mathbf{x}}$$

Now, let define the sliding vector $\mathbf{s} \in \mathbb{R}^{n\times 1}$ as

$$\mathbf{s} = \dot{\mathbf{e}} + \mathbf{\Lambda e} \quad (6)$$

where $\mathbf{\Lambda} = \mathrm{diag}\{l_1,...,l_n\} \in \mathbb{R}^{n\times n}$ is a diagonal and positive definite matrix. Differentiating (6) with respect to time and calculating $\ddot{\mathbf{e}}$ in terms of $\mathbf{s}$ using (2) yields

$$\mathbf{M_x(\dot{s} - \Lambda\dot{e}) = M_x\ddot{x}_d + C_{xm}\dot{x}_d - C_{xm}(s - \Lambda e) + G_x + D_x + f_e + f_s - f} \quad (7)$$

Then, selecting the control law as

$$\mathbf{f} = \mathbf{M_x(\ddot{x}_d + \Lambda\dot{e}) + C_{xm}(\dot{x}_d + \Lambda e) + G_x + D_x} \\ \mathbf{+ f_e + f_s + Ks} \quad (8)$$

where $\mathbf{K} \in \mathbb{R}^{n\times n}$ is a positive definite matrix, and substituting (8) into (7) and rearranging terms gives

$$\mathbf{M_x\dot{s} + (C_{xm} + K)s = 0} \quad (9)$$

In this equation, since the coefficients are not zero, the sliding vector $\mathbf{s}$ approaches zero, which means that Eq. (9) shows the stability of the closed-loop system as well. But the main obstacle in realizing this control law is that matrices $\mathbf{G_x}, \mathbf{C_{xm}}, \mathbf{M_x}$, and $\mathbf{D_x}$ in (8) are usually undefined and unknown. Some of the terms of the control law (8) can be written as (Craig, 1989):

$$\mathbf{Y(x, \dot{x}, x_d, \dot{x}_d)\boldsymbol{\varphi} = M_x(\ddot{x}_d + \Lambda\dot{e}) + C_{xm}(\dot{x}_d + \Lambda e)} \\ \mathbf{+ G_x + D_x} \quad (10)$$

where $\boldsymbol{\varphi} \in \mathbb{R}^{r\times 1}$ is a vector of the unknown parameters and $\mathbf{Y}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{x_d}, \dot{\mathbf{x}}_\mathbf{d}) \in \mathbb{R}^{n\times r}$ is a regression matrix and known. Vector $\boldsymbol{\varphi}$ can be considered as the adaptive parameter, which should be adjusted in such a way to guarantee the stability and the minimum error of the closed loop system. Eq. (10), which is the nonlinear part of control law (8), can be written using robust control method (Hsu and Fu, 2000) as

$$\begin{aligned}\mathbf{Y}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{x_d}, \dot{\mathbf{x}}_\mathbf{d})\boldsymbol{\varphi} = \mathbf{h}(\|\mathbf{x}\|, \mathbf{s}) = \mathbf{M_x}(\ddot{\mathbf{x}}_\mathbf{d} + \boldsymbol{\Lambda}\dot{\mathbf{e}}) \\ +\mathbf{C_{xm}}(\dot{\mathbf{x}}_\mathbf{d} + \boldsymbol{\Lambda}\mathbf{e}) + \mathbf{G_x} + \mathbf{D_x}\end{aligned} \tag{11}$$

where $\|\mathbf{x}\|$ is the norm of the position vector, $\mathbf{s}$ is the sliding vector defined in (6), and $\mathbf{h}(\|\mathbf{x}\|, \mathbf{s}) \in \mathbb{R}^{n\times 1}$ is an unknown vector whose elements can be defined as

$$h_i(\|\mathbf{x}\|, \mathbf{s}) = \mathbf{w}_i^{*T} \boldsymbol{\eta}_i(\|\mathbf{x}\|, s_i) \tag{12}$$

where $\boldsymbol{\eta}_i(\|\mathbf{x}\|, s_i) \in \mathbb{R}^{r\times 1}$ and $\mathbf{w}_i^*$ is the optimal value of $\mathbf{w}_i \in \mathbb{R}^{r\times 1}$. Hence, the control law (8) becomes

$$\begin{aligned}\mathbf{f} = \left[\mathbf{w}_1^{*T} \boldsymbol{\eta}_1(\|\mathbf{x}\|, s_1) \;\; \mathbf{w}_2^{*T} \boldsymbol{\eta}_2(\|\mathbf{x}\|, s_2) \;\; \mathbf{w}_3^{*T} \boldsymbol{\eta}_3(\|\mathbf{x}\|, s_3)\right]^T \\ +\mathbf{f_e} + \mathbf{f_s} + \mathbf{Ks}\end{aligned} \tag{13}$$

where $\mathbf{w} = [\mathbf{w}_1^T \;\; \mathbf{w}_2^T \;\; \mathbf{w}_3^T]^T$ is the adaptive parameter of the controller for 3 3DoF robot and must be adaptively trained as on-line to minimize the tracking error and to maintain the stability of the closed-loop system as well. Since the proposed controller in this paper is of neuro-fuzzy type, therefore, $\mathbf{w}$ is considered as the weight vector of the neuro-fuzzy network. Also, another adaptive neuro-fuzzy estimator, whose design will be explained in the next section, will compensate the friction force.

Each subcontroller (i.e. one controller for every joint) has two inputs, $\|\mathbf{x}\|$, the norm of the position vector, and s_i, the *i*th component of the sliding vector $\mathbf{s}$ in (6). The fuzzy rules of these subcontrollers can be defined as

$$\mathrm{R}_i^{j_1 j_2}: \text{IF } \|\mathbf{x}\| \text{ is } A_i^{j_1} \text{ and } s_i \text{ is } A_i^{j_2} \text{ THEN } y_i \text{ is } B^{j_1 j_2} \tag{14}$$

where $A_i^{j_1}$ and $A_i^{j_2}$ $(i = 1, 2, 3)$, $(j_1 = 1, \ldots, l_1)$, $(j_2 = 1, \ldots, l_2)$ are the j_1th and the j_2th fuzzy sets defined for the fuzzy input variables $\|\mathbf{x}\|$ and s_i, in the *i*th subcontroller, respectively, and $B^{j_1 j_2}$ are the fuzzy sets for the tally part of the fuzzy rules. Using the Mamdani product inference engine, singleton fuzzifier, and center average defuzzifier, the output of the *i*th subcontroller can be calculated as

$$\begin{aligned}f_{\mathrm{Nf}_i} &= \mathbf{w}_i^T \boldsymbol{\eta}_i(\|\mathbf{x}\|, s_i) \\ &= \frac{\sum_{j_1=1}^{l_1}\sum_{j_2=1}^{l_2} \mu_{A^{j_1}}(\|\mathbf{x}\|)\;\mu_{A^{j_2}}(s_i)\;\bar{y}^{j_1 j_2}}{\sum_{j_1=1}^{l_1}\sum_{j_2=1}^{l_2} \mu_{A^{j_1}}(\|\mathbf{x}\|)\;\mu_{A^{j_2}}(s_i)}\end{aligned} \tag{15}$$

where $\bar{y}^{j_1 j_2}$ are the center of the membership functions defined for fuzzy sets $B^{j_1 j_2}$ in (14).

The adaptive parameters of the subcontrollers are the center of the membership functions defined for fuzzy sets $B^{j_1 j_2}$ (i.e. $\bar{y}^{j_1 j_2}$s), which are the elements of $\mathbf{w}$ in the adaptive control law (13). Therefore the three subcontrollers must be trained in such a way that the nonlinear functions $\mathbf{w}_i^T \boldsymbol{\eta}_i(\|\mathbf{x}\|, s_i)$ (i = 1, 2, 3) are estimated with good accuracy. This adaptation law is based on Lyapunov stability theory and will be derived in section 5. In the next section, an adaptive neuro-fuzzy compensator will be developed for surface friction.

4. ADAPTIVE NEURO-FUZZY COMPENSATOR FOR SURFACE FRICTION

When the tip of the endeffector moves on the surface of an object, the surface friction force affects the performance of the controller and deters the correct task of the robot manipulator. Hence, it is important to compensate this friction properly; otherwise there will be large steady-state error. The friction force is affected by the movement of the endeffector on the surface and cannot be directly measured. The models given in section 2 have uncertainties and their parameters are unknown in practice. Moreover, the model may not precisely define the surface friction. Several methods have been proposed in literature for surface friction force compensation. The proposed method by Park et al. (2003) estimates the surface friction force using an adaptive neuro-fuzzy estimator. The free parameters of this estimator are updated adaptively using the tracking errors until the friction force is completely compensated. Since the surface friction force is directly proportional to the applied force to the surface and has a nonlinear relationship with the velocity of the endeffector, the appropriate input to the proposed estimator should be the velocity of the joints in the Cartesian space. Therefore, three subcompensators, one for every direction in the Cartesian Space, must be designed. The fuzzy rules for subcompensators are defined as

$$\mathrm{R}_i^j: \text{IF } \dot{x}_i \text{ is } A_i^j \text{ THEN } y_i \text{ is } B_i^j \tag{16}$$

where $\dot{x}_i$ $(i = 1, 2, 3)$ is the *i*th component of the velocity vector of the endeffector in the Cartesian coordinates, A_i^j and B_i^j $(i = 1, 2, 3,\; j = 1, \ldots, l)$ are the *j*th fuzzy sets defined for the input and the output for the *i*th subcompensator, respectively, and y_i is the output of the *i*th subcompensator, which must be multiplied to f_e and added to the output of the neuro-fuzzy controller obtained form the equations in the previous section. Using singleton fuzzifier, product inference engine, and center average defuzzifier, the output of the *i*th neuro-fuzzy subcompensator cab be expressed as

$$y_i = \sum_{j=1}^{l} \mu_{A_i^j}(\dot{x}_i)\,\bar{y}^j \Big/ \sum_{j=1}^{l} \mu_{A_i^j}(\dot{x}_i) \tag{17}$$

where $\bar{y}^{j}$ is the center of the *j*th membership function of the tally part of the fuzzy rules in (16) and also the free parameters of the neuro-fuzzy subcompensators. Hence, The compensated force for surface friction in the *i*th direction is

$$\hat{f}_{si} = y_i \, f_e \tag{18}$$

where $\hat{f}_{si}$ is the estimation of the *i*th component for surface friction and f_e is the perpendicular force, applied to the surface of the object, and measured by the force sensor. Eq. (17) can be written in the following vector notation form

$$y_i = \boldsymbol{\theta}_i^T \, \boldsymbol{\xi}_i(\dot{x}_i) \tag{19}$$

where $\boldsymbol{\theta}_i = \left[\bar{y}^1 \; \ldots \; \bar{y}^l\right]^T$ is a vector containing the free parameters of the *i*th neuro-fuzzy subcompensator, $\boldsymbol{\xi}_i(\dot{x}_i) = [m_{A_i^l}(\dot{x}_i) \;\; \ldots \;\; m_{A_i^l}(\dot{x}_i)] \Big/ \sum_{j=1}^{l} m_{A_i^j}(\dot{x}_i)$ and *l* is the number of fuzzy rules. Therefore, the estimated force for surface friction compensation is

$$\hat{\mathbf{f}}_{\mathbf{s}} = \hat{\mathbf{f}}_{\mathbf{s}}(\dot{\mathbf{x}}|\boldsymbol{\theta}) = [\boldsymbol{\theta}_1^T \boldsymbol{\xi}_1(\dot{x}_1) f_e \quad \boldsymbol{\theta}_2^T \boldsymbol{\xi}_2(\dot{x}_2) f_e \quad 0]^T \tag{20}$$

where $\boldsymbol{\theta} = \left[\boldsymbol{\theta}_1^T \;\; \boldsymbol{\theta}_2^T \;\; 0\right]^T$. The third component of the friction force is zero, because there are no desired movements in this direction. Hence, the control law given in Eq. (13) can be written as

$$\mathbf{f} = \mathbf{w}^{*T} \boldsymbol{\eta} + \mathbf{f}_{\mathbf{e}} + \hat{\mathbf{f}}_{\mathbf{s}}(\dot{\mathbf{x}}|\boldsymbol{\theta}) + \mathbf{Ks} \tag{21}$$

where $\mathbf{w}^* = \left[\mathbf{w}_1^{*T} \;\; \mathbf{w}_2^{*T} \;\; \mathbf{w}_3^{*T}\right]^T$.

5. ADAPTATION ALGORITHM FOR THE CONTROLLER AND THE COMPENSATOR

There are two sets of parameters, $\mathbf{w}$ and $\boldsymbol{\theta}$, which must be trained during operation of the robot to reach their optimal values $\mathbf{w}^*$ and $\boldsymbol{\theta}^*$, respectively. Therefore, the errors for subcontrollers and subcompensators can be defined as

$$\begin{cases} \boldsymbol{\psi}_i = \mathbf{w}_i^* - \mathbf{w}_i \\ \boldsymbol{\varphi}_i = \boldsymbol{\theta}_i^* - \boldsymbol{\theta}_i \end{cases}, \quad i = 1, 2, 3 \tag{22}$$

As a result, there are three sets of errors in the whole system, namely $\boldsymbol{\psi} = [\boldsymbol{\psi}_1^T \;\; \boldsymbol{\psi}_2^T \;\; \boldsymbol{\psi}_3^T]^T$, $\boldsymbol{\varphi} = [\boldsymbol{\varphi}_1^T \;\; \boldsymbol{\varphi}_2^T \;\; \boldsymbol{\varphi}_3^T]^T$, and $\mathbf{s} = (\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \boldsymbol{\Lambda}(\mathbf{x}_d - \mathbf{x})$, which can be incorporated into one error function in quadratic form, which is the candidate for the Lyapunov function

$$V = \frac{1}{2}\mathbf{s}^{\mathbf{T}}\mathbf{M}_{\mathbf{x}}\mathbf{s} + \frac{1}{2}\sum_{i=1}^{n} \boldsymbol{\psi}_i^T \boldsymbol{\Gamma}_i^{-1} \boldsymbol{\psi}_i + \frac{1}{2}\sum_{i=1}^{n} \boldsymbol{\varphi}_i^T \boldsymbol{\Phi}_i^{-1} \boldsymbol{\varphi}_i \tag{23}$$

where *n*=3 for a 3DoF robot, $\mathbf{M}_{\mathbf{x}}$ is the same as in Eq. (1), and $\boldsymbol{\Gamma}_i = \operatorname{diag}\{g_{1i}, \ldots, g_{ri}\} \in \mathbb{R}^{r \times r}$ and $\boldsymbol{\Phi}_i = \operatorname{diag}\{f_{1i}, \ldots, f_{ri}\} \in \mathbb{R}^{l \times l}$ are diagonal and positive definite matrices, in which *r* and *l* are the number of fuzzy rules in the *i*th subcontroller and ith subcompensator, respectively. Differentiating (23) with respect to time yields

$$\dot{V} = \mathbf{s}^T \mathbf{M}_{\mathbf{x}} \dot{\mathbf{s}} + \frac{1}{2}\mathbf{s}^T \dot{\mathbf{M}}_{\mathbf{x}} \mathbf{s} + \sum_{i=1}^{n} \boldsymbol{\psi}_i^T \boldsymbol{\Gamma}_i^{-1} \dot{\boldsymbol{\psi}}_i + \sum_{i=1}^{n} \boldsymbol{\varphi}_i^T \boldsymbol{\Phi}_i^{-1} \dot{\boldsymbol{\varphi}}_i \tag{24}$$

Using Eqs. (22) and (21) and noting that $\dot{\mathbf{M}}_{\mathbf{x}} - 2\mathbf{C}_{\mathbf{xm}}$ is a skew symmetric matrix

$$\dot{V} = \mathbf{s}^T \left\{ \begin{matrix} \begin{bmatrix} \mathbf{w}_1^{*T} \boldsymbol{\eta}_1(\|\mathbf{x}\|, s_1) \\ \mathbf{w}_2^{*T} \boldsymbol{\eta}_2(\|\mathbf{x}\|, s_2) \\ \mathbf{w}_3^{*T} \boldsymbol{\eta}_3(\|\mathbf{x}\|, s_3) \end{bmatrix} - \begin{bmatrix} \mathbf{w}_1^{T} \boldsymbol{\eta}_1(\|\mathbf{x}\|, s_1) \\ \mathbf{w}_2^{T} \boldsymbol{\eta}_2(\|\mathbf{x}\|, s_2) \\ \mathbf{w}_3^{T} \boldsymbol{\eta}_3(\|\mathbf{x}\|, s_3) \end{bmatrix} \\ + \mathbf{f}_{\mathbf{s}} - \hat{\mathbf{f}}_{\mathbf{s}}(\dot{\mathbf{x}}|\boldsymbol{\theta}) - \mathbf{Ks} \end{matrix} \right\}$$

$$+ \sum_{i=1}^{n} \boldsymbol{\psi}_i^T \boldsymbol{\Gamma}_i^{-1} \dot{\boldsymbol{\psi}}_i + \sum_{i=1}^{n} \boldsymbol{\varphi}_i^T \boldsymbol{\Phi}_i^{-1} \dot{\boldsymbol{\varphi}}_i \tag{25}$$

Now, let $\boldsymbol{\varepsilon} = \mathbf{f}_{\mathbf{s}}(\dot{\mathbf{x}}) - \hat{\mathbf{f}}_{\mathbf{s}}(\dot{\mathbf{x}} \,|\, \boldsymbol{\theta}^*)$ be the error between the estimated friction force and its actual value. Then

$$\dot{V} = -\mathbf{s}^T \mathbf{Ks} + \mathbf{s}^T \boldsymbol{\varepsilon} + \sum_{i=1}^{n} \boldsymbol{\psi}_i^T \boldsymbol{\Gamma}_i^{-1} \dot{\boldsymbol{\psi}}_i + s_i \boldsymbol{\psi}_i^T \boldsymbol{\eta}_i(\|\mathbf{x}\|, s_i)$$

$$+ \sum_{i=1}^{n} \boldsymbol{\varphi}_i^T \boldsymbol{\Phi}_i^{-1} \dot{\boldsymbol{\varphi}}_i + s_i \boldsymbol{\varphi}_i^T \boldsymbol{\xi}_{\mathrm{i}}(\dot{x}_i) f_e \tag{26}$$

If matrix $\mathbf{K}$ is a positive definite matrix, then $\mathbf{s}^T \mathbf{Ks} > \mathbf{s}^T \boldsymbol{\varepsilon}$ since $\boldsymbol{\varepsilon}$ is a small number. Consequently, the following conditions must be hold in order to have $\dot{V} < 0$:

$$\begin{matrix} \boldsymbol{\psi}_i^T \boldsymbol{\Gamma}_i^{-1} \dot{\boldsymbol{\psi}}_i + s_i \boldsymbol{\psi}_i^T \boldsymbol{\eta}_i(\|\mathbf{x}\|, s_i) = 0 \\ \boldsymbol{\varphi}_i^T \boldsymbol{\Phi}_i^{-1} \dot{\boldsymbol{\varphi}}_i + s_i \boldsymbol{\varphi}_i^T \boldsymbol{\xi}_{\mathrm{i}}(\dot{x}_i) f_e = 0 \end{matrix} \quad \forall i = 1, \ldots, n \tag{27}$$

Since $\dot{\boldsymbol{\psi}}_i = \dot{\mathbf{w}}_i^* - \dot{\mathbf{w}}_i$ and $\dot{\boldsymbol{\varphi}}_i = \dot{\boldsymbol{\theta}}_i^* - \dot{\boldsymbol{\theta}}_i$, and also since at the optimal point $\dot{\mathbf{w}}_i^* = 0$ and $\dot{\boldsymbol{\theta}}_i^* = 0$, it can be concluded that

$$\begin{matrix} \dot{\mathbf{w}}_i = s_i \boldsymbol{\Gamma}_i \boldsymbol{\eta}_i(\|\mathbf{x}\|, s_i) \\ \dot{\boldsymbol{\theta}}_i = s_i f_e \boldsymbol{\Phi}_i \boldsymbol{\xi}_{\mathrm{i}}(\dot{x}_i) \end{matrix} \quad \forall i = 1, \ldots, n \tag{28}$$

Integrating these equations gives the adaptive algorithms for updating the free parameters of the neuro-fuzzy subcontrollers and the neuro-fuzzy subcompensators

$$\begin{matrix} \mathbf{w}_i(t_2) = \mathbf{w}_i(t_1) + \int_{t1}^{t2} s_i \boldsymbol{\Gamma}_i \boldsymbol{\eta}_i(\|\mathbf{x}\|, s_i) dt \\ \boldsymbol{\theta}_i(t_2) = \boldsymbol{\theta}_i(t_1) + \int_{t1}^{t2} s_i f_e \boldsymbol{\Phi}_i \boldsymbol{\xi}_{\mathrm{i}}(\dot{x}_i) dt \end{matrix} \quad \forall i = 1, \ldots, n \tag{29}$$

Using these equations, the parameters are updated until they reach their optimal values in order to bring the position and the force errors to minima. One important point is that equations (29) are independent of the robot dynamics and the surface friction. They also guarantee the stability of the closed-loop system, since they have been derived in such a way that the derivative of the Lyapunov function remains always negative. Substituting Eqs. (28) in (26) yields

$$\dot{V} = -\mathbf{s}^T \mathbf{Ks} + \mathbf{s}^T \boldsymbol{\varepsilon} \tag{30}$$

Hence, it can be concluded that $\dot{V} < 0$ if and only if $\mathbf{s}^T \mathbf{Ks} > \mathbf{s}^T \boldsymbol{\varepsilon}$, which intuitively should be true since $\boldsymbol{\varepsilon}$ is small. But there cannot be any guarantee that the estimated friction force is close enough to its actual value, resulting to $\mathbf{s}^T \mathbf{Ks} \le \mathbf{s}^T \boldsymbol{\varepsilon}$ and $\dot{V} \ge 0$, yielding an unstable system. To overcome this problem, the following robust control law will be used (Sugie, 1988):

$$\mathbf{f} = \begin{bmatrix} \mathbf{w}_1^T \boldsymbol{\eta}_1(\|\mathbf{x}\|, s_1) \\ \mathbf{w}_2^T \boldsymbol{\eta}_2(\|\mathbf{x}\|, s_2) \\ \mathbf{w}_3^T \boldsymbol{\eta}_3(\|\mathbf{x}\|, s_3) \end{bmatrix} + \mathbf{f_e} + \hat{\mathbf{f}}_\mathbf{s}(\dot{\mathbf{x}}|\boldsymbol{\theta}) + \mathbf{K}\,\mathbf{s} + \mathbf{Q}\,\mathrm{sgn}(\mathbf{s}) \tag{31}$$

where $\mathbf{Q} = \mathrm{diag}\{q_1, ..., q_n\} \in \mathbb{R}^{n\times 1}$ is a positive definite matrix. Substituting (31) in (25) gives

$$\dot{V} = -\mathbf{s}^T \mathbf{K}\,\mathbf{s} + \mathbf{s}^T \boldsymbol{\varepsilon} - \mathbf{Q}\,\mathrm{sgn}(\mathbf{s}) \tag{32}$$

Now, let define the elements of matrix $\mathbf{Q}$ to be

$$q_i \ge |e_i|, \qquad i = 1, ..., n \tag{33}$$

Then, $\dot{V} < 0$, and $\dot{V} = 0$ only if $\mathbf{s} = 0$, which means that the closed-loop system is asymptotically stable.

6. SIMULATION RESULTS

The membership functions for the input variables of the subcontrollers are 3 gaussian functions equally spaced over intervals {0, 1} and {-1, 1} for $\|\mathbf{x}\|$ and for s_i, respectively. Also, the membership functions for the output variable are 9 triangular functions equally spaced over interval {-4, 4}. The adaptive parameters for the controller (i.e. the elements of vector $\mathbf{w}$) are the center of the membership functions for the output variable. The fuzzy rules for subcontrollers have been given in Table 1. Therefore, each subcontroller has only 9 rules with only 9 adaptive parameters, which must be trained adaptively to minimize the tracking errors. The membership functions for the input variables of the subcompensators ($\dot{x}_1$ and $\dot{x}_2$) are 7 gaussian functions equally spaced over intervals {-2, 2}. Also, the membership functions for the output variable are 7 triangular functions equally spaced over the interval {-0.6, 0.6}. The adaptive parameters of the compensator (i.e. the elements of vector $\boldsymbol{\theta}$) are the center of the membership functions for the output variable. The fuzzy rules for subcompensators have been given in Table 2. As this table shows, each compensator has only 7 rules with only 7 adaptive parameters, which must be updated adaptively for a better estimation of the actual surface friction.

Although the designed controller in this paper is adaptive, but for faster convergence, the initial values of the adaptive parameters (the centers of membership functions for the tally part of the fuzzy rules) are defined using prior knowledge of the system, which is one of many advantages of fuzzy systems.

The 3DoF robot manipulator, used in simulations, has waist, shoulder and elbow with the following lengths and masses for its three arms:

$$l_1 = 0.5\mathrm{m}\ ,\ l_2 = 0.8\mathrm{m}\ ,\ l_3 = 0.3\mathrm{m}$$

$$m_1 = 2\mathrm{kg}\ ,\ m_2 = 1\mathrm{kg}\ ,\ m_3 = 1\mathrm{kg}$$

Also, the following parameters have been chosen for the neuro-fuzzy controller and the neuro-fuzzy compensator:

$$\mathbf{K}=\mathrm{diag}\{30\ \ 30\ \ 30\},\quad \boldsymbol{\Lambda}=\mathrm{diag}\{0.4\ \ 0.4\ \ 0.4\}$$

$$\boldsymbol{\Gamma}_i=\mathrm{diag}\{50\ \ 50\ \ 50\}\ \forall i,\quad \boldsymbol{\Phi}_i=\mathrm{diag}\{50\ \ 50\ \ 50\}\ \forall i$$

$$\mathbf{Q}=\mathrm{diag}\{10\ \ 10\ \ 10\}$$

The desired path is a circle in $x_1 - x_2$ plane with a radius of one meter and the center at the origin. The endeffector must exert 10 N force on the surface while following the desired path. In order to show the adaptation ability of the proposed controller and the proposed compensator, the mass of the arms are increased by 100% of their initial values at $t = 5\,\mathrm{s}$. Figs. 1 shows the position and force control for 12 s of real time. As this Fig. shows, the proposed method in this paper is perfectly capable of controlling the position and the applied force of the endeffector to the desired values even with large amount of changes in the system parameters.

To compare the performance of the proposed controller with another method, simulations have been performed for the same robot with computed torque method, which is dynamic dependent and nonadaptive (Lewis et al., 1993). The simulation results have been shown in Fig. 2. As this figure shows, when there are no changes in the system parameters, the controller can perform a very task with almost no steady state error. But, since the computed torque method is dynamic dependent, any changes in the robot parameters can create large errors, especially for 100% increase in the arms masses it becomes unstable.

Fig. 3 shows the changes in the parameters of one of the adaptive neuro-fuzzy subcontrollers, for the case of 100% increase in the arms masses. As this figure shows, the initial values of the adaptable weights are very close to the desired values, since these weights have been defined using the prior knowledge from the system. Also, as the parameters of the system change at *t*=5 s, the weights quickly adapt themselves to these changes, maintaining small amount of tracking errors for the position control as well as the force control. Fig. 4 indicates that the derivative of the Lyapunov function, defined in Eq. (23), remains negative for the case of 100% increase in the arms masses of the robot, meaning that the system is stable at all time.

7. CONCLUSIONS

An adaptive neuro-fuzzy controller for hybrid position/force control of robot manipulators and an adaptive surface friction compensator was presented in this paper. The proposed algorithms were designed to be independent of the robot dynamics, resulting into an exceedingly robust closed loop system. This is because, the adaptation law of the free parameters was derived based on the Lyapunov stability theory. The other advantage of the proposed method is the simple structure of the controller and the compensator. There are only 9 and 7 fuzzy rules for every subcontroller and every subcompensator, respectively, making it simple, fast and appropriate method for different applications of robot manipulators.

REFERENCES

Bona, B., Indri, M., 1995. Friction compensation and robustness issues in force/position controlled manipulators. IEE Proceedings: Control Theory and Applications 142 (6), 569-574.

Craig, J.J., 1989. Introduction to Robotic: Mechanics and Control. Addison-Wesley Publishing Co.

Hogan, N., 1985. Impedance control: an approach to manipulator, part i, ii, iii. ASME Journal of Dynamic Measurements and Control 3, 1-24.

Hsu, F.Y., Fu, L.C., 2000. Intelligent robot deburring using adaptive fuzzy hybrid position/force control. IEEE Transactions on Robotics and Automation 16 (4), 325-335.

Huang, S.N., Tan, K.K., Lee, T.H., 2000. Adaptive friction compensation using neural network approximations. IEEE Transactions on System, Man and Cybernetics: Part C 30 (4), 551-557.

Kiguchi K., Watanabe, K., Izumi, K., Fukuda, T., 1999. Two-stage adaptation of a position/force robot controller –application of soft computing techniques. IEEE International Conference on Knowledge-Based Intelligent Information Engineering Systems, 141-144.

Kiguchi, K., Fukuda, T., 2000. Position/Force control of robot manipulators for geometrically unknown objects using fuzzy neural networks. IEEE Transactions on Industrial Electronics 47 (3), 641-649.

Lewis, F.L., Abdallah, C.T., Dawson, D.N., 1993. Control of Robot Manipulators. Macmillan Publishing Co.

Park, E.C., Lim, H., Choi, C.H., 2003. Position control of x–y table at velocity reversal using presliding friction characteristics. IEEE Transactions on Control System Technology 11 (1), 24-31.

Sugie, T., 1988. Robust controller design for robot manipulators. ASME Transactions on Dynamics, Measurements and Control 110 (1), 94-96.

Xiao, D., Ghosh, B.K., Xi, N., Tran, T.J., 2000. Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment. IEEE Transactions on Control Systems Technology 8 (4), 635-645.

Table 1: Fuzzy rules for the *i*th subcontroller

$\|\mathbf{x}\|$ / s_i	P	PM	PB
N	M9	M8	M7
Z	M6	M5	M4
P	M3	M2	M1

Table 2: Fuzzy rules for the *i*th subcompensator

$\dot{x}_i$	y_i
NB	NB
NM	NM
N	N
Z	Z
P	P
PM	PM
PB	PB

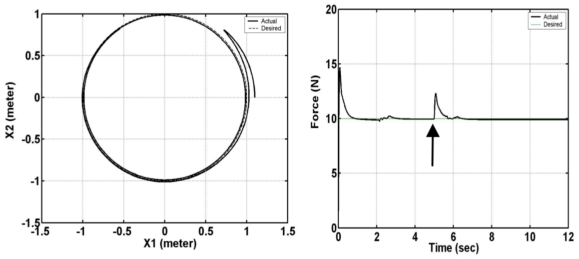

Fig. 1- Position and force control with 100% increase in arms masses at *t*=5 s (proposed method).

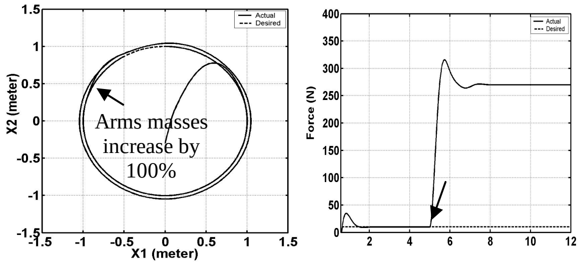

Fig. 2- Position and force control with 100% increase in arms mass at *t*=5 s (computed torque method).

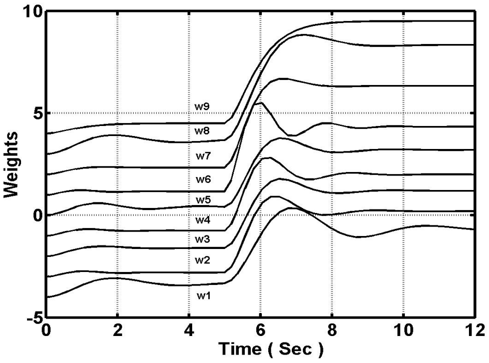

Fig. 3. Adaptive weight changes of one of the subcontrollers during operation of the robot, when 100% increase in arms masses occurs at *t*=5 s.

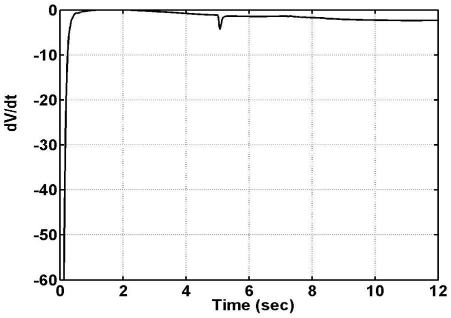

Fig. 4. Derivative of the Lyapunov function, defined in Eq. (23), during operation of the robot, when 100% increase in arms masses occurs at *t*=5 s.