Robust adaptive neuro-fuzzy controller for hybrid position/force control of robot manipulators in contact with unknown environment

Arash Fanaei and Mohammad Farrokhi[*]
Iran University of Science and Technology, Department of Electrical Engineering, Narmak, Tehran 16844, Iran
Tel.: +9821 7808022; Fax: +9821 7454055; E-mail: farrokhi@iust.ac.ir
** Corresponding author*

Abstract - In this paper, an adaptive control method for hybrid position/force control of robot manipulators, based on neuro-fuzzy modeling, is presented. Since the force control involves applying certain amount of force on the surface of an object, it is important to consider the friction force between the end-effector and the surface into account. In order to compensate this friction force, a robust and adaptive neuro-fuzzy compensator will be designed and incorporated into the close-loop system. Moreover, to determine stiffness coefficient of the surface, an on-line estimator will be designed for more precise computation of the desired force. Due to the adaptive neuro-fuzzy modeling, the proposed controller is independent of the robot dynamics, since the free parameters of the neuro-fuzzy controller are adaptively updated to cope with the changes in the system and the environment. As a result, the tracking error, both for position and force, will always remain small. Also, the stability of the controller is guaranteed, since the adaptation law is based on Lyapunov theory. In addition to that, the convergence of the adaptive parameters will be proved in this paper. The simulation results show good performance of the proposed controller as compared with other conventional control schemes for robot manipulators such as computed torque method.

Keywords: robot – hybrid force/position control – adaptive control – neuro-fuzzy control – surface friction compensator

1. Introduction

Neuro-fuzzy controllers have been successfully applied in the past two decades by many researchers to nonlinear systems. The motivation for this utilization stems from two fundamental properties of neuro-fuzzy systems: 1) the ability to incorporate the knowledge of experts into a controller as fuzzy *IF-THEN* rules, which makes the design of the controller independent of the system dynamics or at least there is no need for exact mathematical model of the system, and 2) the unique property of the neural networks, which is called training, to adapt the free parameters of the controller to the changes in the system parameters and the environment, which makes the controller more robust.

When the end-effector of a robot manipulator is moving in free space, the main goal is to control the arms of robot such that it follows the desired trajectory. On the other hand, when robot must hold and move an object, or when the end-effector must come in contact with a surface with friction and unknown stiffness, then there must be force control in addition to the position control of the end-effector. 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 [1] and 2) the impedance control method [2]. In both methods a precise control scheme is necessary in order to move the end-effector on the right path, exerting the correct amount of force on the object. If the force is not controlled with certain degree of precision, there either will be no contact between the end-effector and the object, or the object and the arms of the robot will be damaged. Hence, the position/force control of robot manipulators requires accurate controllers, especially when the stiffness of the surface is unknown in advance, which is common in most applications such as grinding, pounding, polishing, deburring and twisting.

In recent years, intelligent methods and algorithms, like neural networks and fuzzy logic have been employed for position/force control of manipulators. Hsu and Fu in [3], [4] and [5], 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 and Fukuda in [6] and [7], and Kiguchi et al. in [8] have also used a neuro-fuzzy method for position/force control, but with a simple and static model for the friction between the end-effector and the 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. The weight is not trained adaptively. Also, Kiguchi and Fukuda in [9] have used the neuro-fuzzy method to control a 2DoF robot manipulator, assuming that the stiffness coefficient of the surface is known and the friction force is negligible. The proposed controller in [10] by Xiao et al. 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 many applications, such as grinding and deburring, due to the dust and undesired particles in the environment, the output of the visual system can be noisy and imprecise, hence hampering the task.

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 modeling, 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. This is due to the fact that the free parameters of the neuro-fuzzy controller as well as the neuro-fuzzy compensator are adaptively updated to the changes in the system and 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 controller 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 [11]:

$$\mathbf{M}_{\mathbf{x}}(\mathbf{x})\ddot{\mathbf{x}} + \mathbf{C}_{\mathbf{x}}(\mathbf{x},\dot{\mathbf{x}}) + \mathbf{G}_{\mathbf{x}}(\mathbf{x}) + \mathbf{D}_{\mathbf{x}}(\dot{\mathbf{x}}) = \mathbf{f} - \mathbf{f}_{\mathbf{e}} - \mathbf{f}_{\mathbf{s}} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{n\times 1}$ is the position vector of the end-effector, $\mathbf{M}_{\mathbf{x}}(\mathbf{x}) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $\mathbf{C}_{\mathbf{x}}(\mathbf{x},\dot{\mathbf{x}}) \in \mathbb{R}^{n\times 1}$ is the centrifugal and coriolis force vector, $\mathbf{G}_{\mathbf{x}}(\mathbf{x}) \in \mathbb{R}^{n\times 1}$ is the gravity force vector, $\mathbf{D}_{\mathbf{x}}(\dot{\mathbf{x}}) \in \mathbb{R}^{n\times 1}$ 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 end-effector, $\mathbf{f}_{\mathbf{e}} \in \mathbb{R}^{n\times 1}$ is the applied force vector to the surface by the end-effector, and $\mathbf{f}_{\mathrm{s}} \in \mathbb{R}^{n\times 1}$ is the force vector for the surface friction in Cartesian coordinate. Equation (1) can also be written as

$$\mathbf{M}_{\mathrm{x}}(\mathbf{x})\ddot{\mathbf{x}} + \mathbf{C}_{\mathrm{xm}}(\mathbf{x},\dot{\mathbf{x}})\dot{\mathbf{x}} + \mathbf{G}_{\mathbf{x}}(\mathbf{x}) + \mathbf{D}_{\mathrm{x}}(\dot{\mathbf{x}}) = \mathbf{f} - \mathbf{f}_{\mathrm{e}} - \mathbf{f}_{\mathrm{s}} \tag{2}$$

where $\mathbf{C}_{\mathrm{xm}} \in \mathbb{R}^{n\times n}$. Fig. 1 shows the schematic diagram of a 3DoF robot manipulator, consisting of waist, shoulder and elbow. The position of the end-effector in the Cartesian coordinates can be calculated in terms of joint angles as

$$
\begin{aligned}
x_1 &= \left(l_2 \cos q_2 + l_3 \cos\left(q_2 + q_3\right)\right)\cos q_1 \\
x_2 &= \left(l_2 \cos q_2 + l_3 \cos\left(q_2 + q_3\right)\right)\sin q_1 \\
x_3 &= l_2 \sin q_2 + l_3 \sin\left(q_2 + q_3\right)
\end{aligned}
\tag{3}
$$

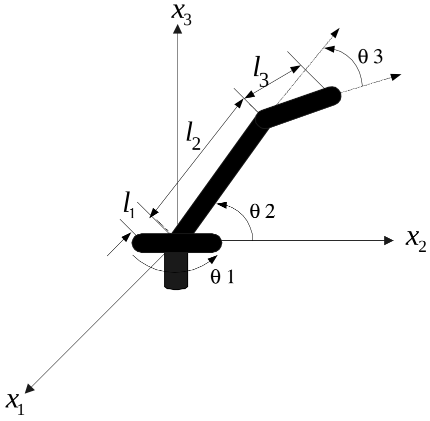


Fig. 1- Schematic diagram of a 3DoF robot manipulator in Cartesian coordinate

Differentiating Eq. (3) w.r.t. time and writing in vector-matrix form, we have

$$\dot{\mathbf{x}} = \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{4}$$

where $\mathbf{J}(\boldsymbol{\theta}) \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$ is called the Jaccobian matrix, which can be used to find the relationship between the joint-torques vector and the end-effector force vector as

$$\mathbf{f} \;=\; \mathbf{J}^{-T}(\boldsymbol{\theta})\;\boldsymbol{\tau} \tag{5}$$

Also, the following equations hold between the matrices and vectors in Cartesian coordinate and in joint-angle coordinate, using the Jaccobian matrix

$$
\begin{gathered}
\mathbf{M}_{\mathbf{x}}(\mathbf{x}) \;=\; \mathbf{J}^{-T}\,\mathbf{M}(\boldsymbol{\theta})\mathbf{J}^{-1} \\
\mathbf{C}_{\mathbf{x}}(\mathbf{x},\dot{\mathbf{x}}) + \mathbf{G}_{\mathbf{x}}(\mathbf{x}) + \mathbf{D}_{\mathbf{x}}(\dot{\mathbf{x}}) \;=\; \mathbf{J}^{-T}\;\left(\mathbf{C}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) + \mathbf{G}(\boldsymbol{\theta}) + \mathbf{D}(\dot{\boldsymbol{\theta}})\;\text{-}\mathbf{M}\,\mathbf{J}^{-1}\dot{\mathbf{J}}\mathbf{J}^{-1}\dot{\mathbf{x}}\right)
\end{gathered}
\tag{6}
$$

where $\mathbf{M}(\boldsymbol{\theta})$, $\mathbf{C}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})$, $\mathbf{G}(\boldsymbol{\theta})$, and $\mathbf{D}(\dot{\boldsymbol{\theta}})$ are the equivalent of $\mathbf{M}_{\mathbf{x}}(\mathbf{x})$, $\mathbf{C}_{\mathbf{x}}(\mathbf{x},\dot{\mathbf{x}})$, $\mathbf{G}_{\mathbf{x}}(\mathbf{x})$, and $\mathbf{D}_{\mathbf{x}}(\dot{\mathbf{x}})$, in joint-angle coordinate, respectively, and can be expressed for a 3DoF robot manipulator as

$$
\mathbf{M}(\boldsymbol{\theta}) = \begin{bmatrix}
m_3 l_3 l_2 c_2 c_{23} + m_3 l_2^2 c_2^2 + \frac{2}{3} m_1 l_1^{\,2} + \frac{1}{3} m_3 l_3^{\,2} c_{23}^{\,2} + \frac{1}{3} m_2 l_2^{\,2} c_2^{\,2} & 0 & 0 \\
0 & m_3 l_3 l_2 c_3 + \frac{1}{3} m_3 l_3^{\,2} + m_3 l_2^{\,2} + \frac{1}{3} m_2 l_2^{\,2} & \frac{1}{2} m_3 l_2 l_3 c_3 + \frac{1}{3} m_3 l_3^{\,2} \\
0 & \frac{1}{2} m_3 l_3 l_2 c_3 + \frac{1}{3} m_3 l_3^{\,2} & \frac{1}{3} m_3 l_3^{\,2}
\end{bmatrix}
$$

$$
\mathbf{C}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) = \begin{bmatrix}
-\dot{q}_1[\frac{1}{3} m_3 l_3^{\,2}(\dot{q}_2 + w_3)\sin 2(q_2 + q_3) + m_3 l_3 l_2(\dot{q}_2 s_2 c_{23} + (\dot{q}_2 + \dot{q}_3)c_2 s_{23}) + m_3 l_2^2 \dot{q}_2 \sin 2q_2 + \frac{1}{3} m_2 l_2^2 \dot{q}_2 \sin 2q_2] \\
-\frac{1}{2} m_3 l_3 l_2 \dot{q}_3(2\dot{q}_2 + \dot{q}_3)s_3 + \frac{1}{2} m_3(\frac{1}{3} l_3^2 \dot{q}_1^2 \sin 2(q_2 + q_3) + l_3 l_2 \dot{q}_1^2 \sin(2q_2 + q_3) + l_2^2 \dot{q}_1^2 \sin 2q_2) + \frac{1}{6} m_2 l_2^2 \dot{q}_1^2 \sin 2q_2 \\
-\frac{1}{2} m_3 l_3 l_2 \dot{q}_3 \dot{q}_2 s_3 + \frac{1}{2} m_3(\frac{1}{3} l_3^2 \dot{q}_1^2 \sin 2(q_2 + q_3) + l_3 l_2 \dot{q}_2(\dot{q}_2 + \dot{q}_3)s_3 + l_3 l_2 \dot{q}_1^2 c_2 s_{23})
\end{bmatrix}
$$

$$\mathbf{G(\theta)} = \begin{bmatrix} 0 \\ m_3(\frac{1}{2} l_3 c_{23} + l_2 c_2) + \frac{1}{2} m_2 l_2 c_2 \\ \frac{1}{2} m_3 l_3 c_{23} \end{bmatrix}, \qquad \mathbf{D}(\dot{\theta}) = \begin{bmatrix} 2\dot{q}_1 + \mathrm{sgn}(\dot{q}_1) \\ 2\dot{q}_2 + \mathrm{sgn}(\dot{q}_2) \\ 2\dot{q}_3 + \mathrm{sgn}(\dot{q}_3) \end{bmatrix} \tag{7}$$

where l_i, m_i, and q_i $(i = 1,2,3)$ are the length, weight and angular position of joints, respectively; and $c_i = \cos q_i$, $s_i = \sin q_i$, $c_{ij} = \cos(q_i + q_j)$, and $s_{ij} = \sin(q_i + q_j)$.

2-2 The Characteristics of the Contact Surface

In order to characterize the surface, which comes in contact with the end-effector, two mathematical models are considered here. The first one is in the direction of the force perpendicular to the surface and the second one is in the direction of movement, tangent to the surface. The perpendicular model describes the stiffness degree of the surface, and is used for force control. The tangent model expresses the friction properties of the surface and is used for force as well as position control of the end-effector.

2-2-1 The Perpendicular Model

The mathematical model in the perpendicular direction of the contact surface is a spring model (Fig. 2) [8], [13], [14], [15]. If the external applied force to the surface is shown with f_e, then

$$f_e = k_e(x_e - x) \tag{8}$$

where k_e is the stiffness coefficient of the surface, in N/m, and x and x_e are the position of the end-effector and the surface, respectively (Fig. 2).

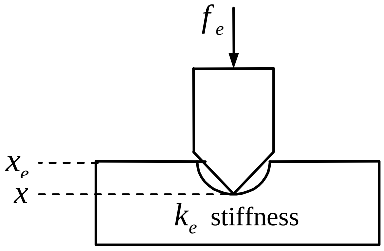


Fig. 2- The perpendicular model of the surface

According to Fig. 2, if the external force f_e is applied to the surface with stiffness coefficient k_e, the shape of the surface will be deformed in form of a hole. The higher the stiffness of the surface is, the smaller the depth of the penetration will be. For instance, 10 N force applied to a surface with $k_e = 10000$ N/m can penetrate 0.001 m. In this paper it is assumed that the surface of the object lies on $x_1 - x_2$ plane, hence $x_e = 0$.

2-2-2 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. Fig. 3 shows the affecting forces applied to an object, moving on a surface. In this Fig., f_{t1} and f_{t2} are moving forces along x_1 and x_2 axis, respectively. The frictions forces f_{s1} and f_{s2} always act in the opposite direction of the moving forces on the surface.

There are many models available in the literature to represent friction forces. Bona and Indri in [16] have shown that a comprehensive mathematical model for friction force f_s can be given in terms of applied perpendicular force f_e as

$$f_s = \left[g_0 + g_1 e^{-b_1|v|} + g_2(1 - e^{-b_2|v|})\right]|f_e|\,\mathrm{sgn}(v) \tag{9}$$

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 (Fig. 4), and b_2 specifies the ascending rate of friction at higher speeds.

Fig. 4 shows the surface friction force versus the speed of the object [17], [18] and [19]. In this Fig., there is a negative slope at lower speeds, which is called the Stribeck effect. If this effect is not compensated, there could be instability around the origin of the friction force-speed characteristics [16].

In position/force control of robot manipulators, the opposing surface friction force must be compensated appropriately; otherwise it could create difficulties in control process, especially in position control, and yet could lead to very large steady-state errors or even can become unstable.

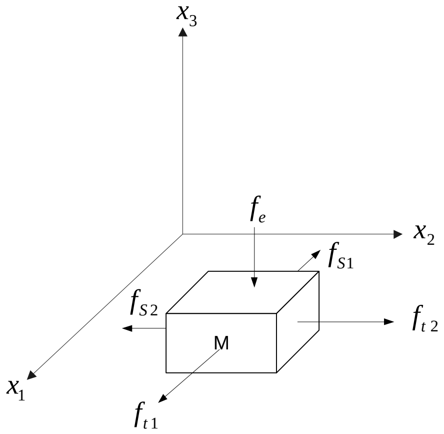


Fig. 3. Applied forces to an object moving on a surface

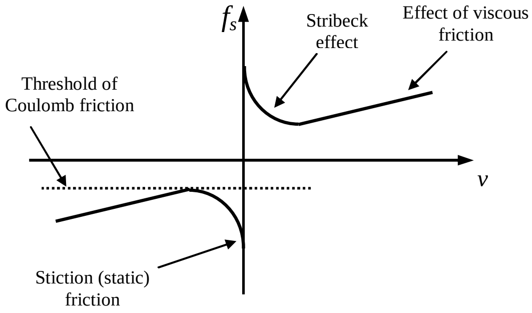


Fig. 4. The complete model for friction with Stribeck effect

2-3 Surface Coefficient Estimation

Usually, the surface friction coefficient is an unknown parameter and should be estimated with good accuracy; otherwise it could lead to large errors in forces applied to the surface or even instability. Since this friction can change with time, the best way to estimate it is to employ an on-line method [24]. The relationship between the applied force to the surface and the amount of the depth of the penetration can be expressed as (Fig. 2):

$$k_e = \frac{f_e}{(x_e - x)} \tag{10}$$

Here, f_e can be measured by a force sensor located at the end-effector and $(x_e - x)$ can be calculated by transforming the angular joint position to the Cartesian space. Since the obtained k_e is not the actual value of the surface coefficient, therefore, the estimated value of this parameter is shown with $\hat{k}_e$ in Fig. 5.

In this paper it is assumed that the end-effector is always in contact with the surface and the force measurement at the end-effector is accurate enough, avoiding zero estimation of the surface stiffness coefficient.

3- Adaptive Neuro-Fuzzy Controller

The goal in this paper is to control position and force of a robot manipulator with 3DoF, with the end-effector moving on a surface in the $x_1 - x_2$ plane (Fig. 1), while applying force in the negative direction of the x_3 axis (i.e. perpendicular to the $x_1 - x_2$ plane). The control action is performed in the Cartesian space $\mathbf{x} = [x_1 \ x_2 \ x_3]^T$, but in practice the robot movement is in the joint space $\boldsymbol{\theta} = [q_1 \ q_2 \ q_3]^T$. Therefore, the Cartesian space is used for controller design, and then using $\boldsymbol{\tau} = \mathbf{J}^{\mathrm{T}}\mathbf{f}$, where $\mathbf{J}$ is the Jaccobian matrix, the control force $\mathbf{f}$ can be converted to the control torques $\boldsymbol{\tau}$ to apply to the robot arms. Hence, the task of the controller is to move the end-effector along the desired path $\mathbf{x}_d = \left[x_{d1} \ x_{d2} \ x_{d3}\right]^T$. Since the force is applied to the surface along the x_3 axis, then

$$f_e = k_e\left(x_{e3} - x_3\right) \tag{10}$$

Also, since we assume that the surface of the object lies on the $x_1 - x_2$ plane, then $x_{e3} = 0$ and we have

$$\mathbf{x} = \left[x_1 \ \ x_2 \ \ f_e/k_e\right]^T \tag{11}$$

Therefore, the desired path and desired force can be defined as

$$\mathbf{x}_{\mathrm{d}} = \left[x_{d1} \ \ x_{d2} \ \ f_d/k_e\right]^T, \quad \mathbf{f}_{\mathrm{d}} = \left[0 \ \ 0 \ \ f_d\right]^T \tag{12}$$

The error between the desired and the actual vectors for the position, velocity and acceleration of the end-effector is

$$\mathbf{e} = \mathbf{x}_{\mathrm{d}} - \mathbf{x} = \begin{bmatrix} x_{d1} - x_1 & x_{d2} - x_2 & \dfrac{f_d - f_e}{k_e} \end{bmatrix}^T, \ \dot{\mathbf{e}} = \dot{\mathbf{x}}_{\mathrm{d}} - \dot{\mathbf{x}}, \ \ddot{\mathbf{e}} = \ddot{\mathbf{x}}_{\mathrm{d}} - \ddot{\mathbf{x}} \tag{13}$$

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

$$\mathbf{s} = \dot{\mathbf{e}} + \boldsymbol{\Lambda}\mathbf{e} \tag{14}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}\left\{l_1, ..., l_n\right\} \in \mathbb{R}^{n\times n}$ is a diagonal and positive definite matrix [20]. Differentiating (14) w.r.t. time and calculating $\ddot{\mathbf{e}}$ in terms of $\mathbf{s}$ using (2), we have

$$\mathbf{M}_{\mathbf{x}}(\dot{\mathbf{s}} - \boldsymbol{\Lambda}\dot{\mathbf{e}}) = \mathbf{M}_{\mathbf{x}}\ddot{\mathbf{x}}_{\mathbf{d}} + \mathbf{C}_{\mathbf{xm}}\dot{\mathbf{x}}_{\mathbf{d}} - \mathbf{C}_{\mathbf{xm}}(\mathbf{s} - \boldsymbol{\Lambda}\mathbf{e}) + \mathbf{G}_{\mathbf{x}} + \mathbf{D}_{\mathbf{x}} + \mathbf{f}_{\mathbf{e}} + \mathbf{f}_{\mathbf{s}} - \mathbf{f} \tag{15}$$

or

$$\mathbf{M}_{\mathbf{x}}\dot{\mathbf{s}} = \mathbf{M}_{\mathbf{x}}(\ddot{\mathbf{x}}_{\mathbf{d}} + \boldsymbol{\Lambda}\dot{\mathbf{e}}) + \mathbf{C}_{\mathbf{xm}}(\dot{\mathbf{x}}_{\mathbf{d}} + \boldsymbol{\Lambda}\mathbf{e}) - \mathbf{C}_{\mathbf{xm}}\mathbf{s} + \mathbf{G}_{\mathbf{x}} + \mathbf{D}_{\mathbf{x}} + \mathbf{f}_{\mathbf{e}} + \mathbf{f}_{\mathbf{s}} - \mathbf{f} \tag{16}$$

Then, selecting the control law as

$$\mathbf{f} = \mathbf{M}_{\mathbf{x}}(\ddot{\mathbf{x}}_{\mathbf{d}} + \boldsymbol{\Lambda}\dot{\mathbf{e}}) + \mathbf{C}_{\mathbf{xm}}(\dot{\mathbf{x}}_{\mathbf{d}} + \boldsymbol{\Lambda}\mathbf{e}) + \mathbf{G}_{\mathbf{x}} + \mathbf{D}_{\mathbf{x}} + \mathbf{f}_{\mathbf{e}} + \mathbf{f}_{\mathbf{s}} + \mathbf{K}\mathbf{s} \tag{17}$$

where $\mathbf{K} \in \mathbb{R}^{n\times n}$ is a positive definite matrix, and substituting (17) into (16) yields

$$\mathbf{M}_{\mathbf{x}}\dot{\mathbf{s}} + (\mathbf{C}_{\mathbf{xm}} + \mathbf{K})\mathbf{s} = \mathbf{0} \tag{18}$$

Eq. (18) shows that using control law in (17), the closed-loop equation is a first order system. Moreover, since the coefficients in (18) are not zero, the sliding vector $\mathbf{s}$ must approach zero in order to satisfy (18). But the main obstacle in realizing this control law is that matrices $\mathbf{G}_{\mathbf{x}}, \mathbf{C}_{\mathbf{xm}}, \mathbf{M}_{\mathbf{x}}$, and $\mathbf{D}_{\mathbf{x}}$ in (17) are usually undefined and unknown. Some of the terms of the control law (17) can be written as [12]

$$\mathbf{Y}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{x}_{\mathbf{d}}, \dot{\mathbf{x}}_{\mathbf{d}})\boldsymbol{\varphi} = \mathbf{M}_{\mathbf{x}}(\ddot{\mathbf{x}}_{\mathbf{d}} + \boldsymbol{\Lambda}\dot{\mathbf{e}}) + \mathbf{C}_{\mathbf{xm}}(\dot{\mathbf{x}}_{\mathbf{d}} + \boldsymbol{\Lambda}\mathbf{e}) + \mathbf{G}_{\mathbf{x}} + \mathbf{D}_{\mathbf{x}} \tag{19}$$

where $\boldsymbol{\varphi} \in \mathbb{R}^{\mathrm{r}\times 1}$ is a vector of the unknown parameters and $\mathbf{Y}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{x}_{\mathbf{d}}, \dot{\mathbf{x}}_{\mathbf{d}}) \in \mathbb{R}^{\mathrm{n}\times \mathrm{r}}$ 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. (19), which is the nonlinear part of control law (17), can be written using robust control method as [5]

$$\mathbf{Y}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{x}_{\mathbf{d}}, \dot{\mathbf{x}}_{\mathbf{d}})\boldsymbol{\varphi} = \mathbf{h}(\|\mathbf{x}\|, \mathbf{s}) = \mathbf{M}_{\mathbf{x}}(\ddot{\mathbf{x}}_{\mathbf{d}} + \boldsymbol{\Lambda}\dot{\mathbf{e}}) + \mathbf{C}_{\mathbf{xm}}(\dot{\mathbf{x}}_{\mathbf{d}} + \boldsymbol{\Lambda}\mathbf{e}) + \mathbf{G}_{\mathbf{x}} + \mathbf{D}_{\mathbf{x}} \tag{20}$$

where $\|\mathbf{x}\|$ is the norm of the position vector, $\mathbf{s}$ is the sliding vector defined in (14), and $\mathbf{h}(\|\mathbf{x}\|,\mathbf{s}) \in \mathbb{R}^{n\times 1}$ is an unknown vector whose elements can be defined as

$$h_i(\|\mathbf{x}\|,\mathbf{s}) = \mathbf{w}_i^{*T} \boldsymbol{\eta}_i(\|\mathbf{x}\|, s_i) \qquad (21)$$

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, equation (16) and the control law (17) become

$$\mathbf{M_x}\dot{\mathbf{s}} = \left[\mathbf{w}_1^{*T} \boldsymbol{\eta}_1(\|\mathbf{x}\|, s_1) \quad \mathbf{w}_2^{*T} \boldsymbol{\eta}_2(\|\mathbf{x}\|, s_2) \quad \mathbf{w}_3^{*T} \boldsymbol{\eta}_3(\|\mathbf{x}\|, s_3)\right]^T - \mathbf{C_{xm}s} + \mathbf{f_e} + \mathbf{f_s} - \mathbf{f} \qquad (22)$$

$$\mathbf{f} = \left[\mathbf{w}_1^{*T} \boldsymbol{\eta}_1(\|\mathbf{x}\|, s_1) \quad \mathbf{w}_2^{*T} \boldsymbol{\eta}_2(\|\mathbf{x}\|, s_2) \quad \mathbf{w}_3^{*T} \boldsymbol{\eta}_3(\|\mathbf{x}\|, s_3)\right]^T + \mathbf{f_e} + \mathbf{f_s} + \mathbf{Ks} \qquad (23)$$

In fact, the nonlinear part of the recent equation has been given as the product of $\boldsymbol{\eta}(\|\mathbf{x}\|,\mathbf{s})$ and $\mathbf{w}$, where $\mathbf{w} = [\mathbf{w}_1^T \ \mathbf{w}_2^T \ \mathbf{w}_3^T]^T$ is the adaptive parameter of the controller and must be adaptively trained as on-line to minimize the tracking error, to cope with the changes in the system parameters and the environment, and to maintain the stability of the closed-loop system. 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, which is trained on-line and adaptively to perform the above three tasks. Also, another adaptive neuro-fuzzy estimator, whose design will be explained in the next section, will compensate the friction force.

The block diagram of the proposed method in this paper has been shown in Fig. 5. The controller block consists of three subcontrollers, one for each joint of the robot. There are also one block for estimation and compensation of surface friction force and another block for on-line estimation of stiffness coefficient of the surface. Since this coefficient is usually not only unknown but also varying, it is important to estimate it with good accuracy. Also, it should be mentioned that sensors, located at the joints of the robot, measure the angle and the angular velocity of joints. Therefore, these quantities need to be converted to position and velocity of the end-effector in Cartesian space using Eqs. (3) and (4).

Each subcontroller has two inputs, $\|\mathbf{x}\|$ (i.e. the norm of the position vector in Cartesian coordinates) and s_i (the *i*th component, *i* =1, 2, 3, of the sliding vector $\mathbf{s}$). The internal connections of each subcontroller are shown in Fig. 6. The fuzzy rules of three subcontrollers can be defined as

$$\mathrm{R}_i^{j_1 j_2}: \mathrm{IF} \|\mathbf{x}\| \text{ is } A_i^{j_1} \text{ and } s_i \text{ is } A_i^{j_2} \text{ THEN } y_i \text{ is } B^{j_1 j_2} \qquad (24)$$

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

$$f_{\mathrm{Nf}_i} = \mathbf{w}_i^T \boldsymbol{\eta}_i(\|\mathbf{x}\|, s_i) = \frac{\sum_{j_1=1}^{l_1}\sum_{j_2=1}^{l_2} \mu_{A^{j_1}}(\|\mathbf{x}\|)\ \mu_{A^{j_2}}(s_i)\ \bar{y}^{j_1 j_2}}{\sum_{j_1=1}^{l_1}\sum_{j_2=1}^{l_2} \mu_{A^{j_1}}(\|\mathbf{x}\|)\ \mu_{A^{j_2}}(s_i)} \qquad (25)$$

where $\bar{y}^{j_1 j_2}$ is the center of the membership function defined for fuzzy sets $B^{j_1 j_2}$ in (24).

The adaptive parameters of the subcontrollers are the center of the memberships 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 (23). 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. In other words, we need an adaptation law for changing the free parameters of the subcontroller (i.e. $\bar{y}^{j_1 j_2}$s) in an on-line fashion subject to: 1) the stability of the close-loop system is guaranteed at all time, 2) the tracking error for position and force remains bounded and small, even for large changes in the system parameters and the environment. 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.

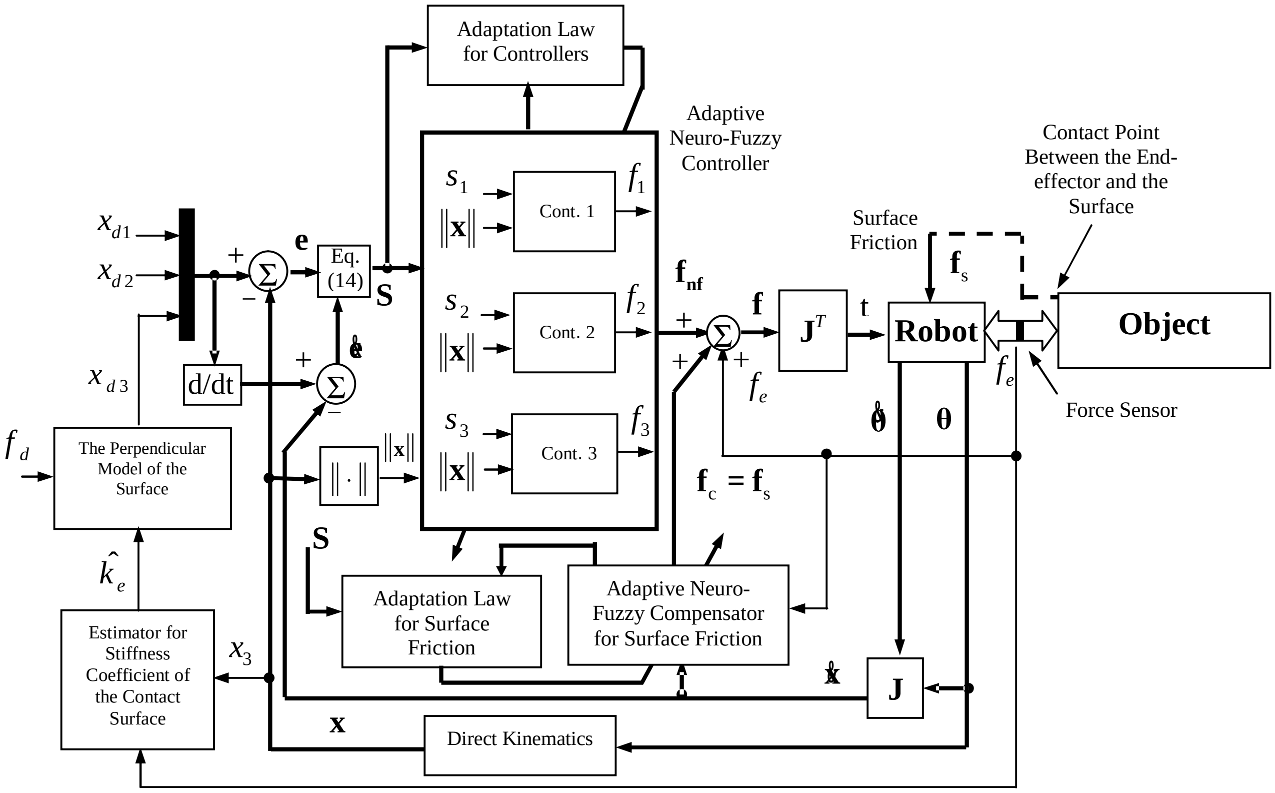

Adaptation Law for Controllers
Adaptive Neuro-Fuzzy Controller
Contact Point Between the End-effector and the Surface
Surface Friction
x_{d1}
x_{d2}
x_{d3}
f_d
+
−
Σ
e
Eq. (14)
S
s_1
$\|\mathbf{x}\|$
Cont. 1
f_1
s_2
Cont. 2
f_2
s_3
Cont. 3
f_3
$\mathbf{f}_{\mathrm{nf}}$
f
$\mathbf{J}^T$
t
Robot
Object
$\mathbf{f}_s$
f_e
Force Sensor
$\dot{\theta}$
θ
d/dt
$\dot{e}$
$\|\cdot\|$
$\|\mathbf{x}\|$
$\mathbf{f}_c = \mathbf{f}_s$
The Perpendicular Model of the Surface
$\hat{k}_e$
Estimator for Stiffness Coefficient of the Contact Surface
x_3
Adaptation Law for Surface Friction
Adaptive Neuro-Fuzzy Compensator for Surface Friction
$\dot{x}$
J
x
Direct Kinematics

Fig. 5- Block diagram of the proposed adaptive neuro-fuzzy controller for hybrid position/force control of a 3DoF robot manipulator along with friction force compensator and the estimator for stiffness coefficient of the contact surface

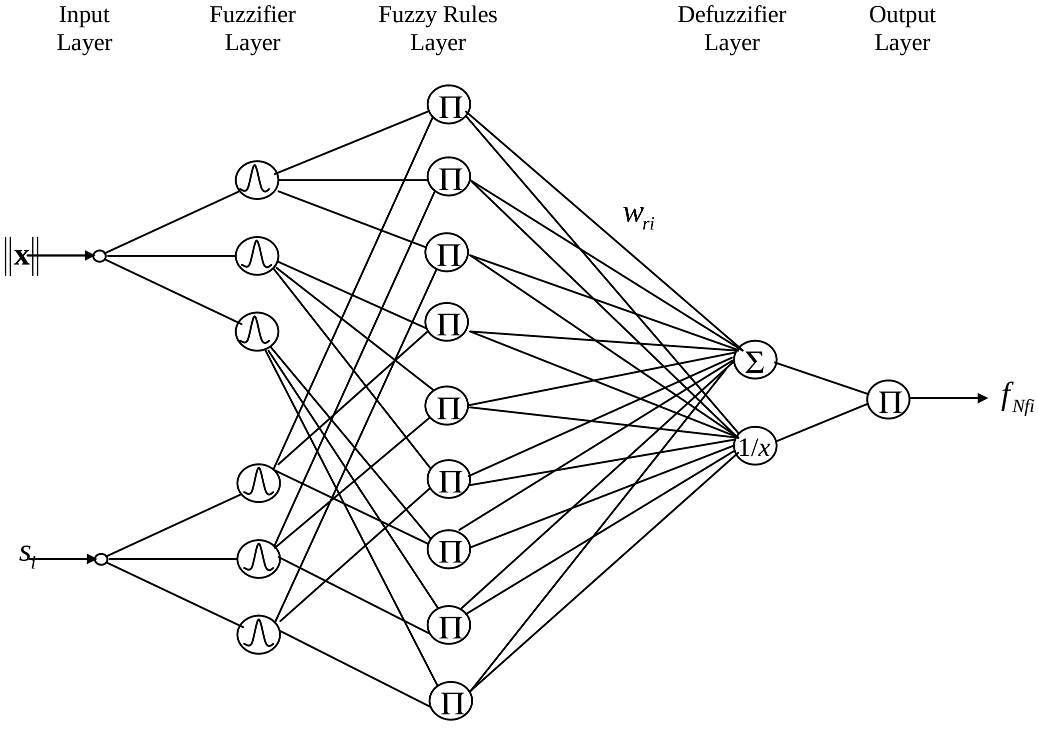


Fig. 6- The neuro-fuzzy structure of the adaptive hybrid position/force control

4- Adaptive Neuro-Fuzzy Compensator for Surface Friction

When the tip of the end-effector moves on the surface of an object, the surface friction force affects the performance of the controller and deters the correct process of the robot manipulator. Hence, it is important to compensate this friction properly; otherwise there will be large steady-state error. Moreover, the system might become unstable for surfaces with large friction coefficients. The friction force is affected by the movement of the end-effector 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. Therefore, in order to compensate the friction force, one must find a good estimate of this force. Several methods have been proposed in literature for surface friction force compensation. One method is to take advantage of high gain in controller. This method compensates the required force to some degrees, but high gain can cause problems like unstability and lack of providing enough force by servomotors. Amirat et al. have proposed a robust compensator, which compensates friction to some good degrees, but due to inadaptability nature of the method, it cannot work well when friction force is increased. Also, It has not been mentioned how to determine the parameters of the compensator [21]. The other method is to use torque sensors in every joint of the robot to measure the actual torques and comparing them with the desired torques (output of the controllers). The difference is the amount of torque consumed for surface friction [22]. This method needs torque sensors in every joint of the robot.

The proposed method in this paper is to estimate 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 end-effector (Fig. 4), the appropriate input to the proposed estimator should be the velocity of joints in Cartesian space. Then, in order to calculate the required force for friction compensation, the output of the neuro-fuzzy compensator will be multiplied to the perpendicular force, applied to the surface. Therefore, three compensator must be designed, one for every direction.

The fuzzy rules of subcompensators are defined as

$$\mathrm{R}_i^j: \text{IF } \dot{x}_i \text{ is } A_i^j \text{ THEN } y_i \text{ is } B_i^j \tag{26}$$

where $\dot{x}_i\ (i=1,2,3)$ is the *i*th component of the velocity vector of the end-effector in Cartesian coordinates, A_i^j and $B_i^j\ (i=1,2,3,\ j=1,\ldots,l)$ are the *j*th membership functions of 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

$$y_i = \frac{\sum_{j=1}^{l} \mu_{A_i^j}(\dot{x}_i)\ \bar{y}^j}{\sum_{j=1}^{l} \mu_{A_i^j}(\dot{x}_i)} \tag{27}$$

where $\bar{y}^j$ is the center of the *j*th membership function of the THEN part of the fuzzy rules in (26) and also the free parameters of the neuro-fuzzy compensators. Hence, The compensated force for surface friction in the *i*th direction is

$$\hat{f}_{si} = y_i\, f_e \tag{28}$$

where $\hat{f}_{si}$ is the estimation of the *i*th component for surface friction and f_e is the perpendicular force, measured by force sensor, and applied to the surface of the object. Eq. (27) can be written in the following vector notation form

$$y_i = \boldsymbol{\theta}_i^T\, \xi_i(\dot{x}_i) \tag{29}$$

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

$$\hat{\mathbf{f}}_{\mathbf{s}} = \hat{\mathbf{f}}_{\mathbf{s}}(\dot{\mathbf{x}}|\boldsymbol{\theta}) = [\boldsymbol{\theta}_1^T \xi_1(\dot{x}_1)\, f_e \quad \boldsymbol{\theta}_2^T \xi_2(\dot{x}_2)\, f_e \quad 0]^T \tag{30}$$

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. (23) can be written as

$$\mathbf{f} = \mathbf{w}^{*T}\boldsymbol{\eta} + \mathbf{f}_{\mathbf{e}} + \hat{\mathbf{f}}_{\mathbf{s}}(\dot{\mathbf{x}}|\boldsymbol{\theta}) + \mathbf{Ks} \tag{31}$$

where $\mathbf{w}^* = \left[\mathbf{w}_1^{*T} \;\; \mathbf{w}_2^{*T} \;\; \mathbf{w}_3^{*T}\right]^T$. Therefore, the free parameters in the controller, $\mathbf{w}$, and in the compensator, $\boldsymbol{\theta}$, must be appropriately adjusted during operation of robot to guarantee stability of the closed-loop system and to ensure small tracking error for position and force as well. Next section gives the proposed algorithm and the stability analysis of that, which are the main contributions of this paper.

5- Adaptation Algorithm for Controller and Compensator

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

$$\begin{cases} \boldsymbol{\psi}_i = \mathbf{w}_i^* - \mathbf{w}_i \\ \boldsymbol{\varphi}_i = \boldsymbol{\theta}_i^* - \boldsymbol{\theta}_i \end{cases}, \quad i = 1, 2, 3 \tag{32}$$

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 our candidate for the Lyapunov function

$$V = \frac{1}{2}\mathbf{s}^{\mathbf{T}}\mathbf{M}_{\mathbf{x}}\mathbf{s} + \frac{1}{2}\sum_{i=1}^{n}\boldsymbol{\psi}_i^T\boldsymbol{\Gamma}_i^{-1}\boldsymbol{\psi}_i + \frac{1}{2}\sum_{i=1}^{n}\boldsymbol{\varphi}_i^T\boldsymbol{\Phi}_i^{-1}\boldsymbol{\varphi}_i \tag{33}$$

where n=3 for a 3DoF robot, $\mathbf{M}_{\mathbf{x}}$ is the same as in Eq. (1), and $\boldsymbol{\Gamma}_i \in \mathbb{R}^{r\times r} = \mathrm{diag}\{g_{1i}, \cdots, g_{ri}\}$ and $\boldsymbol{\Phi}_i \in \mathbb{R}^{l\times l} = \mathrm{diag}\{f_{1i}, \cdots, f_{ri}\}$ are positive definite matrices, in which r and l are the number of fuzzy rules in the *i*th subcontroller and *i*th subcompensator, respectively. Differentiating (33) w.r.t. time yields

$$\dot{V} = \mathbf{s}^T\mathbf{M}_{\mathbf{x}}\dot{\mathbf{s}} + \frac{1}{2}\mathbf{s}^T\dot{\mathbf{M}}_{\mathbf{x}}\mathbf{s} + \sum_{i=1}^{n}\boldsymbol{\psi}_i^T\boldsymbol{\Gamma}_i^{-1}\dot{\boldsymbol{\psi}}_i + \sum_{i=1}^{n}\boldsymbol{\varphi}_i^T\boldsymbol{\Phi}_i^{-1}\dot{\boldsymbol{\varphi}}_i \tag{34}$$

Then, using Eq. (22) gives

$$\begin{aligned}\dot{V} = \mathbf{s}^T\Big([\mathbf{w}_1^{*T}\boldsymbol{\eta}_1(\|\mathbf{x}\|, s_1) \quad \mathbf{w}_2^{*T}\boldsymbol{\eta}_2(\|\mathbf{x}\|, s_2) \quad \mathbf{w}_3^{*T}\boldsymbol{\eta}_3(\|\mathbf{x}\|, s_3)]^T + \mathbf{f}_{\mathbf{e}} + \mathbf{f}_{\mathbf{s}} - \mathbf{f}\Big) + \frac{1}{2}\mathbf{s}^T(\dot{\mathbf{M}}_{\mathbf{x}} - 2\mathbf{C}_{\mathbf{xm}})\mathbf{s} \\ + \sum_{i=1}^{n}\boldsymbol{\psi}_i^T\boldsymbol{\Gamma}_i^{-1}\dot{\boldsymbol{\psi}}_i + \sum_{i=1}^{n}\boldsymbol{\varphi}_i^T\boldsymbol{\Phi}_i^{-1}\dot{\boldsymbol{\varphi}}_i\end{aligned} \tag{35}$$

Since $\dot{\mathbf{M}}_{\mathbf{x}} - 2\mathbf{C}_{\mathbf{xm}}$ is a skew symmetric matrix, we have

$$\mathbf{s}^T(\dot{\mathbf{M}}_{\mathbf{x}} - 2\mathbf{C}_{\mathbf{xm}})\mathbf{s} = \mathbf{0} \tag{36}$$

Substituting $\mathbf{f}$ from (31) into (35) results in

$$\dot V = \mathbf{s}^T \left\{ \begin{bmatrix} \mathbf{w}_1^{*T}\boldsymbol{\eta}_1(\|\mathbf{x}\|,s_1)\\ \mathbf{w}_2^{*T}\boldsymbol{\eta}_2(\|\mathbf{x}\|,s_2)\\ \mathbf{w}_3^{*T}\boldsymbol{\eta}_3(\|\mathbf{x}\|,s_3)\end{bmatrix} - \begin{bmatrix} \mathbf{w}_1^{T}\boldsymbol{\eta}_1(\|\mathbf{x}\|,s_1)\\ \mathbf{w}_2^{T}\boldsymbol{\eta}_2(\|\mathbf{x}\|,s_2)\\ \mathbf{w}_3^{T}\boldsymbol{\eta}_3(\|\mathbf{x}\|,s_3)\end{bmatrix} + \mathbf{f}_s - \hat{\mathbf{f}}_s(\dot{\mathbf{x}}|\boldsymbol{\theta}) - \mathbf{Ks} \right\} + \sum_{i=1}^{n} \boldsymbol{\psi}_i^T \boldsymbol{\Gamma}_i^{-1} \dot{\boldsymbol{\psi}}_i + \sum_{i=1}^{n} \boldsymbol{\varphi}_i^T \boldsymbol{\Phi}_i^{-1} \dot{\boldsymbol{\phi}}_i \tag{37}$$

Now, let the error between the estimated friction force $\hat{\mathbf{f}}_s(\dot{\mathbf{x}}|\boldsymbol{\theta}^*)$ and its actual value $\mathbf{f}_s(\dot{\mathbf{x}})$ be

$$\boldsymbol{\varepsilon} = \mathbf{f}_s(\dot{\mathbf{x}}) - \hat{\mathbf{f}}_s(\dot{\mathbf{x}}|\boldsymbol{\theta}^*) \tag{38}$$

where $\boldsymbol{\varepsilon} \in \mathbf{R}^{n\times 1}$. This error can also be written as

$$\hat{\mathbf{f}}_s(\dot{\mathbf{x}}|\boldsymbol{\theta}^*) - \hat{\mathbf{f}}_s(\dot{\mathbf{x}}|\boldsymbol{\theta}) = [\boldsymbol{\varphi}_1^T \xi_1(\dot{x}_1) f_e \quad \boldsymbol{\varphi}_2^T \xi_2(\dot{x}_2) f_e \quad 0]^T \tag{39}$$

Therefore, combining Eqs. (38) and (39) yields

$$\mathbf{f}_s(\dot{\mathbf{x}}) - \hat{\mathbf{f}}_s(\dot{\mathbf{x}}|\boldsymbol{\theta}) = [\boldsymbol{\varphi}_1^T \xi_1(\dot{x}_1) f_e \quad \boldsymbol{\varphi}_2^T \xi_2(\dot{x}_2) f_e \quad 0]^T + \boldsymbol{\varepsilon} \tag{40}$$

Using (40) in (37) gives

$$\begin{aligned} \dot V &= \mathbf{s}^T \left\{ \begin{bmatrix} \mathbf{w}_1^{*T}\boldsymbol{\eta}_1(\|\mathbf{x}\|,s_1)\\ \mathbf{w}_2^{*T}\boldsymbol{\eta}_2(\|\mathbf{x}\|,s_2)\\ \mathbf{w}_3^{*T}\boldsymbol{\eta}_3(\|\mathbf{x}\|,s_3)\end{bmatrix} - \begin{bmatrix} \mathbf{w}_1^{T}\boldsymbol{\eta}_1(\|\mathbf{x}\|,s_1)\\ \mathbf{w}_2^{T}\boldsymbol{\eta}_2(\|\mathbf{x}\|,s_2)\\ \mathbf{w}_3^{T}\boldsymbol{\eta}_3(\|\mathbf{x}\|,s_3)\end{bmatrix} + \boldsymbol{\varepsilon} - \mathbf{Ks} \right\} + \sum_{i=1}^{n} \boldsymbol{\psi}_i^T \boldsymbol{\Gamma}_i^{-1} \dot{\boldsymbol{\psi}}_i + \sum_{i=1}^{n} \boldsymbol{\varphi}_i^T \boldsymbol{\Phi}_i^{-1} \dot{\boldsymbol{\phi}}_i \\ &= -\mathbf{s}^T \mathbf{Ks} + \mathbf{s}^T \boldsymbol{\varepsilon} + \sum_{i=1}^{n} \left[\boldsymbol{\psi}_i^T \boldsymbol{\Gamma}_i^{-1} \dot{\boldsymbol{\psi}}_i + s_i \boldsymbol{\psi}_i^T \boldsymbol{\eta}_i(\|\mathbf{x}\|,s_i) \right] + \sum_{i=1}^{n} \left[\boldsymbol{\varphi}_i^T \boldsymbol{\Phi}_i^{-1} \dot{\boldsymbol{\phi}}_i + s_i \boldsymbol{\varphi}_i^T \xi_i(\dot{x}_i) f_e \right] \end{aligned} \tag{41}$$

Now, let

$$\begin{aligned} \boldsymbol{\psi}_i^T \boldsymbol{\Gamma}_i^{-1} \dot{\boldsymbol{\psi}}_i + s_i \boldsymbol{\psi}_i^T \boldsymbol{\eta}_i(\|\mathbf{x}\|,s_i) = 0 \\ \boldsymbol{\varphi}_i^T \boldsymbol{\Phi}_i^{-1} \dot{\boldsymbol{\phi}}_i + s_i \boldsymbol{\varphi}_i^T \xi_i(\dot{x}_i) f_e = 0 \end{aligned} \qquad \forall i = 1, \dots, n \tag{42}$$

or

$$\begin{aligned} \boldsymbol{\psi}_i^T \left(\boldsymbol{\Gamma}_i^{-1} \dot{\boldsymbol{\psi}}_i + s_i \boldsymbol{\eta}_i(\|\mathbf{x}\|,s_i) \right) = 0 \\ \boldsymbol{\varphi}_i^T \left(\boldsymbol{\Phi}_i^{-1} \dot{\boldsymbol{\phi}}_i + s_i \xi_i(\dot{x}_i) f_e \right) = 0 \end{aligned} \qquad \forall i = 1, \dots, n \tag{43}$$

which yields

$$\begin{aligned} \dot{\boldsymbol{\psi}}_i &= -s_i \boldsymbol{\Gamma}_i \boldsymbol{\eta}_i(\|\mathbf{x}\|,s_i) \\ \dot{\boldsymbol{\phi}}_i &= -s_i f_e \boldsymbol{\Phi}_i \xi_i(\dot{x}_i) \end{aligned} \qquad \forall i = 1, \dots, n \tag{44}$$

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

$$\begin{aligned} \dot{\mathbf{w}}_i &= s_i \boldsymbol{\Gamma}_i \boldsymbol{\eta}_i(\|\mathbf{x}\|,s_i) \\ \dot{\boldsymbol{\theta}}_i &= s_i f_e \boldsymbol{\Phi}_i \xi_i(\dot{x}_i) \end{aligned} \qquad \forall i = 1, \dots, n \tag{45}$$

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

$$\begin{aligned} \mathbf{w}_i(t_2) &= \mathbf{w}_i(t_1) + \int_{t1}^{t2} s_i \boldsymbol{\Gamma}_i \boldsymbol{\eta}_i(\|\mathbf{x}\|,s_i) dt \\ \boldsymbol{\theta}_i(t_2) &= \boldsymbol{\theta}_i(t_1) + \int_{t1}^{t2} s_i f_e \boldsymbol{\Phi}_i \xi_i(\dot{x}_i) dt \end{aligned} \qquad \forall i = 1, \dots, n \tag{46}$$

Using these equations, the parameters are updated until they reach their optimal values in order to bring the position and force errors to minima. One important point is that equations (46) are independent of the robot dynamics and the surface friction.

In the followings, it will be proved that adaptation law given in (46) can guarantee the stability of the closed loop system under certain conditions.

Theorem 1: If (46) holds $\forall t$ and if $\|\boldsymbol{\varepsilon}\|$ in (38) is bounded, then the closed loop system is stable $\forall t$.

Proof: Since (46) is directly deducted from (44), hence, by substituting (44) into (41) it yields

$$V\& = -\mathbf{s}^T\mathbf{K}\mathbf{s} + \mathbf{s}^T\boldsymbol{\varepsilon} + \sum_{i=1}^{n}\left[-s_i\boldsymbol{\psi}_i^T\boldsymbol{\Gamma}_i^{-1}\boldsymbol{\Gamma}_i\boldsymbol{\eta}_i + s_i\boldsymbol{\psi}_i^T\boldsymbol{\eta}_i\right] + \sum_{i=1}^{n}\left[-s_i f_e\boldsymbol{\varphi}_i^T\boldsymbol{\Phi}_i^{-1}\boldsymbol{\Phi}_i\xi_{\mathrm{i}} + s_i f_e\boldsymbol{\varphi}_i^T\xi_{\mathrm{i}}\right] \tag{47}$$

Now, since $\boldsymbol{\Gamma}_i$ and $\boldsymbol{\Phi}_i$ $(i = 1, \mathbf{K}, n)$ are diagonal matrices, then their inverse exists and $\boldsymbol{\Gamma}_i^{-1}\boldsymbol{\Gamma}_i = \mathbf{I}$ and $\boldsymbol{\Phi}_i^{-1}\boldsymbol{\Phi}_i = \mathbf{I}$. Therefore

$$V\& = -\mathbf{s}^T\mathbf{K}\mathbf{s} + \mathbf{s}^T\boldsymbol{\varepsilon} + \sum_{i=1}^{n} s_i\left[-\boldsymbol{\psi}_i^T\boldsymbol{\eta}_i + \boldsymbol{\psi}_i^T\boldsymbol{\eta}_i\right] + \sum_{i=1}^{n} s_i f_e\left[-\boldsymbol{\varphi}_i^T\xi_{\mathrm{i}} + \boldsymbol{\varphi}_i^T\xi_{\mathrm{i}}\right] \tag{48}$$

which yields

$$V\& = -\mathbf{s}^T\mathbf{K}\mathbf{s} + \mathbf{s}^T\boldsymbol{\varepsilon} \tag{49}$$

Since $\|\boldsymbol{\varepsilon}\|$ is bounded, then matrix $\mathbf{K}$ in (17) can be chosen as a diagonal and positive definite matrix with large elements as compared to the elements of $\boldsymbol{\varepsilon}$ such that

$$\left|\mathbf{s}^T\mathbf{K}\mathbf{s}\right| > \left|\mathbf{s}^T\boldsymbol{\varepsilon}\right| \tag{50}$$

Therefore, $V\& < 0$ and the closed loop system is stable, which concludes the proof. □

Corollary 1: The adaptation law, given in (46) has the following properties

$$\begin{aligned} \|\Delta\mathbf{w}_i(t_{k+1})\| &< \|\Delta\mathbf{w}_i(t_k)\| \\ \|\Delta\boldsymbol{\theta}_i(t_{k+1})\| &< \|\Delta\boldsymbol{\theta}_i(t_k)\| \end{aligned} \qquad \forall\begin{cases} i = 1, \mathbf{K}, n \\ t_k, k = 1, 2, \mathbf{K} \end{cases} \tag{51}$$

where

$$\begin{aligned} \|\Delta\mathbf{w}_i(t_{k+1})\| &= \|\mathbf{w}_i(t_{k+1}) - \mathbf{w}_i(t_k)\| \\ \|\Delta\boldsymbol{\theta}_i(t_{k+1})\| &= \|\boldsymbol{\theta}_i(t_{k+1}) - \boldsymbol{\theta}_i(t_k)\| \end{aligned} \qquad \forall\begin{cases} i = 1, \mathbf{K}, n \\ t_k, k = 1, 2, \mathbf{K} \end{cases} \tag{52}$$

Proof: Without loss of generality, let $t_{k-1} = 0$ and $t_k = t$. According to theorem 1, the adaptation law in (46) yields a stable closed loop system $\forall t$. Hence, according to the sliding control theory (Slotine, 1991) the sliding vector $\mathbf{s} = \&\mathbf{e} + \boldsymbol{\Lambda}\mathbf{e}$ in (14) must decrease asymptotically. That is

$$\lim_{t\to\infty} s_i(t) \to 0 \quad \forall i \tag{53}$$

Consequently, the integral terms of (46) go to zero

$$\begin{aligned} &\lim_{t\to\infty}\int_0^t s_i\boldsymbol{\Gamma}_i\boldsymbol{\eta}_i(\|\mathbf{x}\|, s_i)dt \to 0 \\ &\lim_{t\to\infty}\int_0^t s_i f_e\boldsymbol{\Phi}_i\xi_{\mathrm{i}}(x\&_i)dt \to 0 \end{aligned} \qquad \forall i \tag{54}$$

Hence, as $k \to \infty$

$$\begin{aligned} \|\mathbf{w}_i(t_{k+1})\| &\to \|\mathbf{w}_i(t_k)\| \\ \|\boldsymbol{\theta}_i(t_{k+1})\| &\to \|\boldsymbol{\theta}_i(t_k)\| \end{aligned} \qquad \forall i \tag{55}$$

Or equivalently

$$\begin{aligned} &\lim_{k\to\infty}\|\Delta\mathbf{w}_i(t_{k+1})\| \to 0 \\ &\lim_{k\to\infty}\|\Delta\boldsymbol{\theta}_i(t_{k+1})\| \to 0 \end{aligned} \qquad \forall i \tag{56}$$

which must be true only if

$$\begin{aligned} \|\Delta\mathbf{w}_i(t_{k+1})\| &< \|\Delta\mathbf{w}_i(t_k)\| \\ \|\Delta\boldsymbol{\theta}_i(t_{k+1})\| &< \|\Delta\boldsymbol{\theta}_i(t_k)\| \end{aligned} \qquad \forall\begin{cases} i = 1, \mathbf{K}, n \\ t_k, k = 1, 2, \mathbf{K} \end{cases} \tag{57}$$

This completes the proof. □

Remark: In proof of Theorem 1, it was assumed that the elements of diagonal matrix $\mathbf{K}$ are large enough such that inequality (50) holds for different values of $\boldsymbol{\varepsilon}$. This might create some difficulties in the control process, especially when the desired force is too large, which might create large overshoots. Moreover, when the states of the system are close to their desired values, but the difference between the desired force and the actual force is too high (i.e. when the end-effector is very close to the desired point but hasn't touch it yet) the elements of matrix $\mathbf{K}$ must be extremely high, which could lead to large chattering around the sliding surface. To overcome this problem, the following robust control law has been used in this paper (Sugie, 1988):

$$\mathbf{f} = \begin{bmatrix} \mathbf{w}_1^T \boldsymbol{\eta}_1(\|\mathbf{x}\|, s_1) \\ \mathbf{w}_2^T \boldsymbol{\eta}_2(\|\mathbf{x}\|, s_2) \\ \mathbf{w}_3^T \boldsymbol{\eta}_3(\|\mathbf{x}\|, s_3) \end{bmatrix} + \mathbf{f_e} + \hat{\mathbf{f}}_{\mathbf{s}}(\dot{\mathbf{x}}|\boldsymbol{\theta}) + \mathbf{K}\,\mathbf{s} + \mathbf{Q}\,\mathrm{sgn}(\mathbf{s}) \tag{58}$$

where $\mathbf{Q} \in \mathbb{R}^{n\times1} = \mathrm{diag}\{q_1, ..., q_n\}$ is a positive definite matrix. Now, the derivative of the Lyapunov function becomes

$$\dot{V} = -\mathbf{s}^T \mathbf{K}\,\mathbf{s} + \mathbf{s}^T \boldsymbol{\varepsilon} - \mathbf{Q}\,\mathrm{sgn}(\mathbf{s}) \tag{59}$$

Let define the elements of matrix $\mathbf{Q}$ to be

$$q_i \geq |e_i|, \qquad i = 1, ..., n \tag{60}$$

Then, $\dot{V} < 0$, and $\dot{V} = 0$ only if $\mathbf{s} = 0$, which means that using control law (58), the closed-loop system is asymptotically stable. Moreover, this control law can avoid large chattering around the sliding surface.

6- Simulations

The membership functions for the input and the output variables of subcontrollers are shown in Figs. 7 and 8, respectively. The adaptive parameters for the controller (i.e. the elements of vector $\mathbf{w}$) are the center of the membership functions shown in Fig. 8 (i.e. $\bar{y}^{j_1 j_2}$s in (25)). Also, 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 error.
The membership functions for the input and the output variables of the subcompensators for surface friction are shown in Figs. 9 and 10, respectively. The adaptive parameters of the compensator (i.e. the elements of vector $\boldsymbol{\theta}$) are the center of the membership functions shown in Fig. 10 (i.e. $\bar{y}^{j}$s in (27)). Also, the fuzzy rules for subcompensators have been given in Table 2. As this table shows, the compensator has only 7 rules with only 7 adaptive parameters, which must be updated adaptively for a better estimation of the desired surface friction.

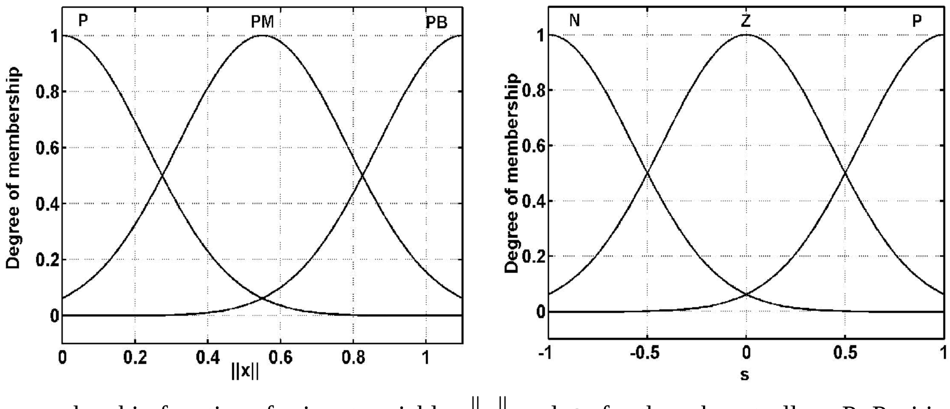


Fig. 7- The membership functions for input variables $\|\mathbf{x}\|$ and s_i for the subcontrollers. P=Positive, PM=Positive Medium, PB=Positive Big, N=Negative, Z=Zero, P=Positive.

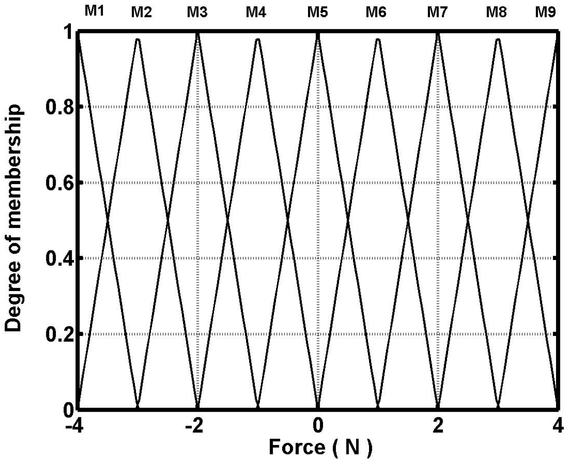


Fig. 8- The membership functions for the output variable in subcontrollers

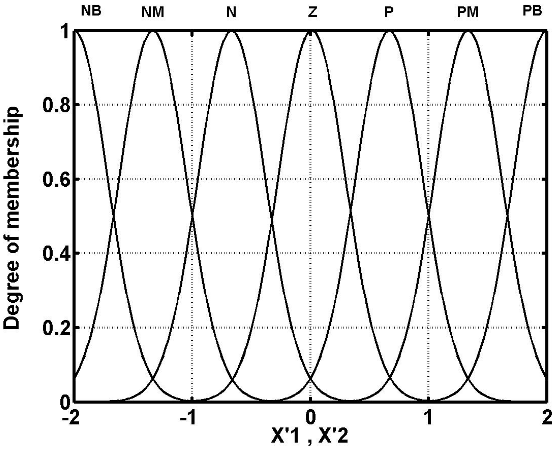


Fig. 9- The membership functions for the input variables $\tilde{x}_1$ and $\tilde{x}_2$ of the surface-friction compensators. NB=Negative Big, NM=Negative Medium, N=Negative, Z=Zero, P=Positive, PM=Positive Medium, PB=Positive Big.

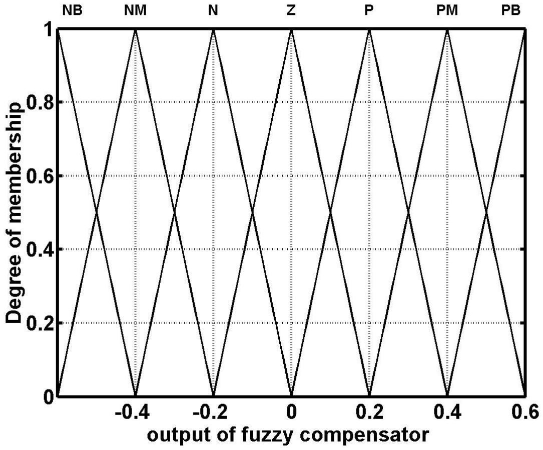


Fig. 10- The membership functions for the output variable of the surface-friction compensator

Table 1: Fuzzy rules for the subcontrollers

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

Table 2: Fuzzy rules for the subcompensators

1- IF $\dot{x}_i$ is NB, THEN y_i is NB

2- IF $\dot{x}_i$ is NM, THEN y_i is NM

3- IF $\dot{x}_i$ is N, THEN y_i is N

4- IF $\dot{x}_i$ is Z, THEN y_i is Z

5- IF $\dot{x}_i$ is P, THEN y_i is P

6- IF $\dot{x}_i$ is PM, THEN y_i is PM

7- IF $\dot{x}_i$ is PB, THEN y_i is PB

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 the following constants:

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

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

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

$$\mathbf{K}\text{=diag}\{30 \;\; 30 \;\; 30\}, \quad \mathbf{\Lambda}\text{=diag}\{0.4 \;\; 0.4 \;\; 0.4\}$$

$$\mathbf{\Gamma}_i\text{=diag}\{50 \;\; 50 \;\; 50\} \quad \forall i, \quad \mathbf{\Phi}_i\text{=diag}\{50 \;\; 50 \;\; 50\} \quad \forall i, \quad \mathbf{Q}\text{=diag}\{10 \;\; 10 \;\; 10\}$$

The desired path is a circle in $x_1 - x_2$ plane with a radius of one meter and the center of the origin. The end-effector must exert 10 N forces on the surface while following the desired path. The initial value of the adaptive parameters of the controllers has been chosen equally as follows (although it is worth to mention that the performance of the developed method in this paper is independent of the initial value of the adaptive parameters):

$$\mathbf{w}_1(0) = \mathbf{w}_2(0) = \mathbf{w}_3(0) = \begin{bmatrix} -4 & -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 \end{bmatrix}^T$$

In order to show the adaptation ability of the proposed controller and the compensator, simulations have been performed for different values of stiffness coefficients of the surface, surface frictions, joints frictions and arms masses of the robot, and measurement noises. Figs. 11 to 18 show the simulation results. The changes in these parameters are 100% of their initial values occurring at $t = 5$ s. Fig. 18 shows the results for adding 10% measurement noise in position and force sensors.

As the simulation results show, the proposed controller is perfectly capable of controlling the position and the applied force of the end-effector to the desired values even with large amount of changes in system parameters and the environment.

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 [11]. The simulation results have been shown in Figs. 19 to 25. As these figures show, when there are no changes in the system parameters or in the environment, the controller can perform an excellent 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. 21). Moreover, due to uncompensated surface-friction force, any variation in surface friction may even result in separation of the end-effector from the surface (Fig. 23). In addition, Fig. 25 shows that computed torque method is very susceptible to measurements noise.

Also, Fig. 26 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 knowledge we have from the system. Also, as the parameters of the system change at *t*=5 s, the weights quickly adapt themselves to the changes, maintaining small amount of tracking errors both for the position control as well as the force. Fig. 27 indicates that the derivative of the Lyapunov function, defined in Eq. (33), remains negative for the case of 100% increase in the arms masses of the robot, meaning that the system is stable at all time.

One important point in the simulation results is that although the fuzzy systems are universal approximators and are always associated with some errors, but, due to the adaptation property of the controller as well as the compensator, the approximation error, inherent in the fuzzy system, is cancelled out by continuously changing the adaptive parameters.

Although the simulation results show very good and robust performance for the proposed method, there are some limitations that should be addressed: 1) the singular points in the robot work space, which hasn't been considered in the simulations, should be taken into account, 2) the contact surface was assumed to be flat and in the x1-x2 plane, which may not be the general case in practice. Evaluation of the proposed controller when encountering an uneven surface, in which the surface friction force must be decomposed into three components along three axes, will be addressed in the future works, 3) it was assumed that the speed of the end effector can be measured with some sensors. In practice, these sensors are usually associated with noise, 4) even though, measurement noise was considered in simulations, estimating the speed of the end effector, instead of measuring it, could benefit the practical applications of the proposed method.

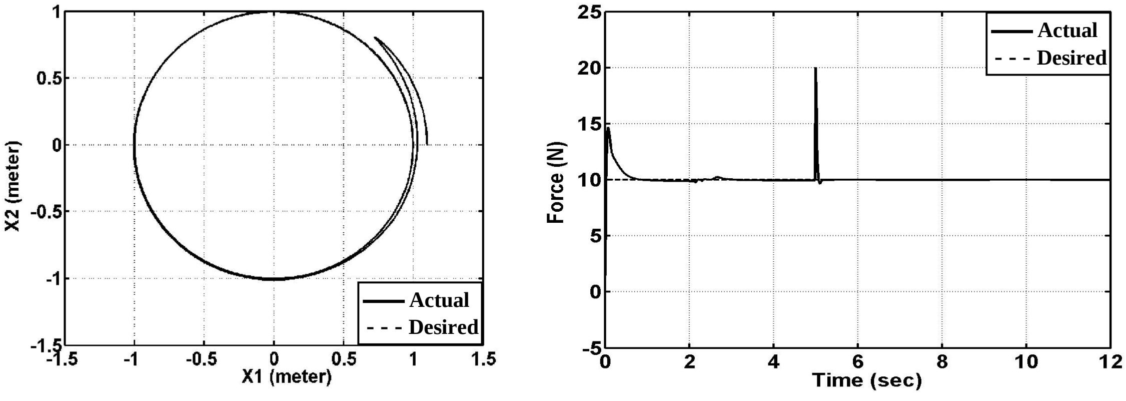


Fig. 11- Position and force control with 100% increase in surface-stiffness coefficient

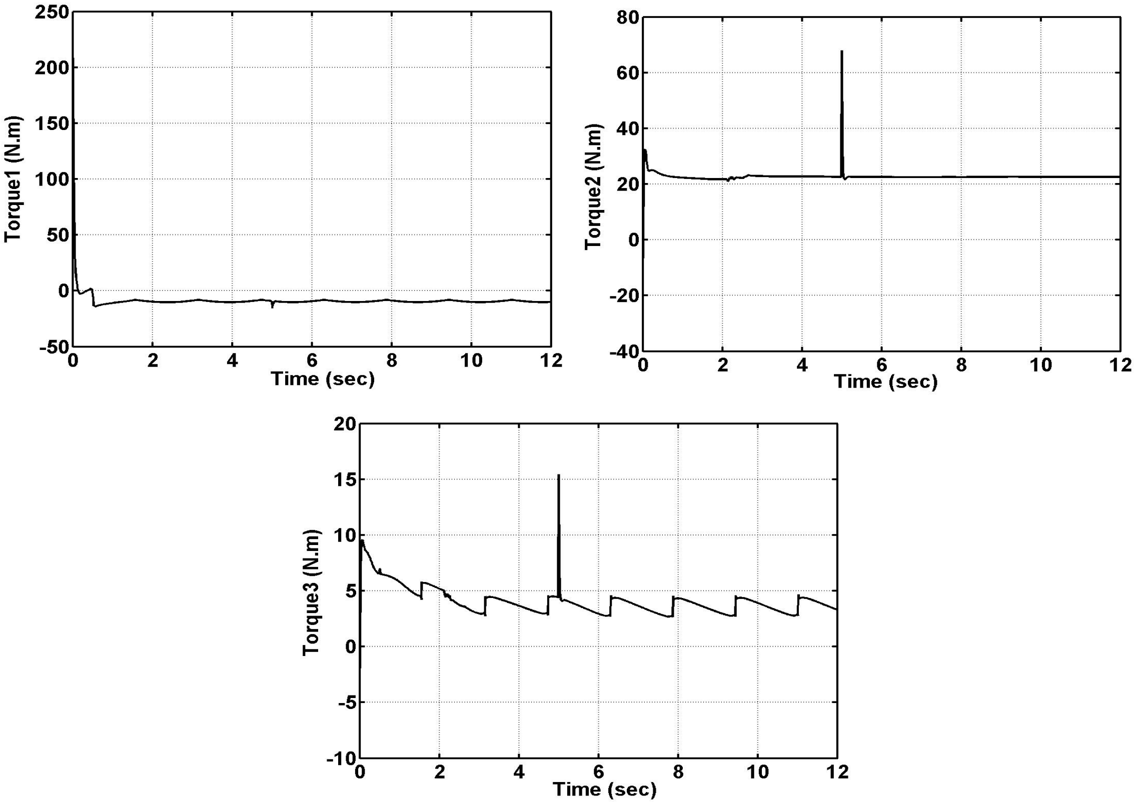


Fig. 12- The joint torques for the case of Fig. 8

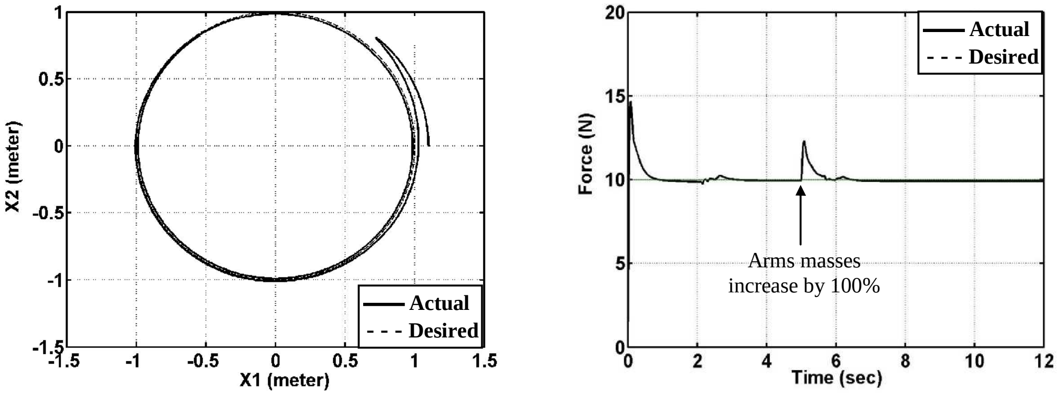


Fig. 13- Position and force control with 100% increase in arms masses

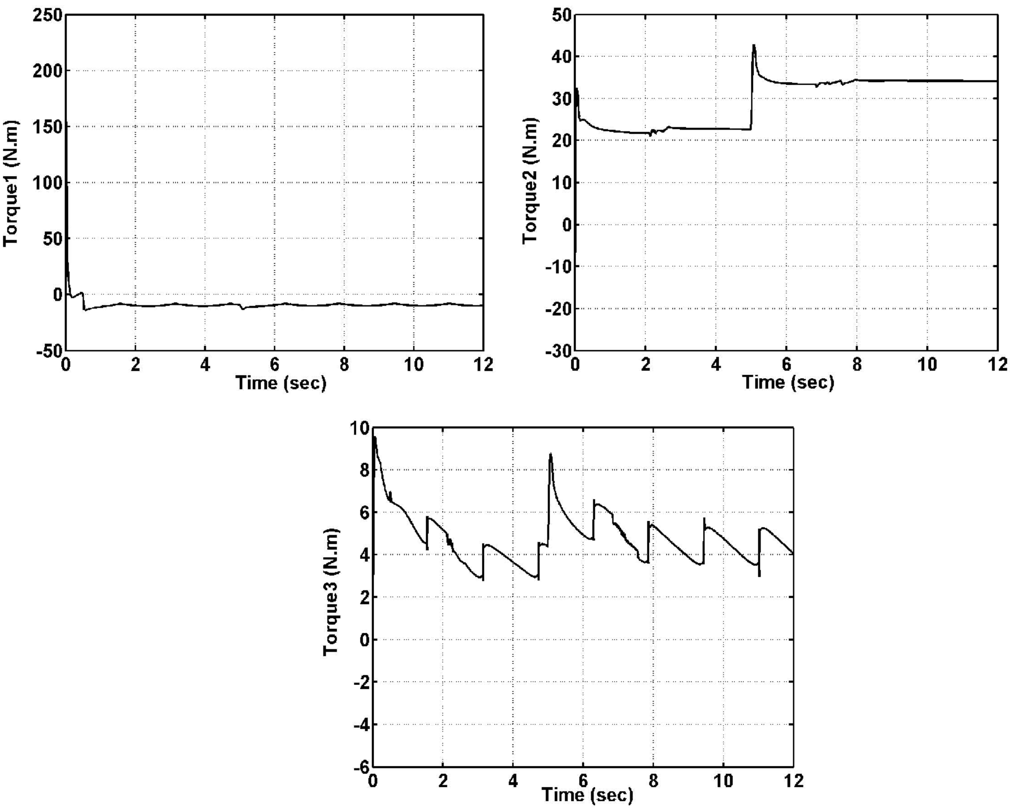
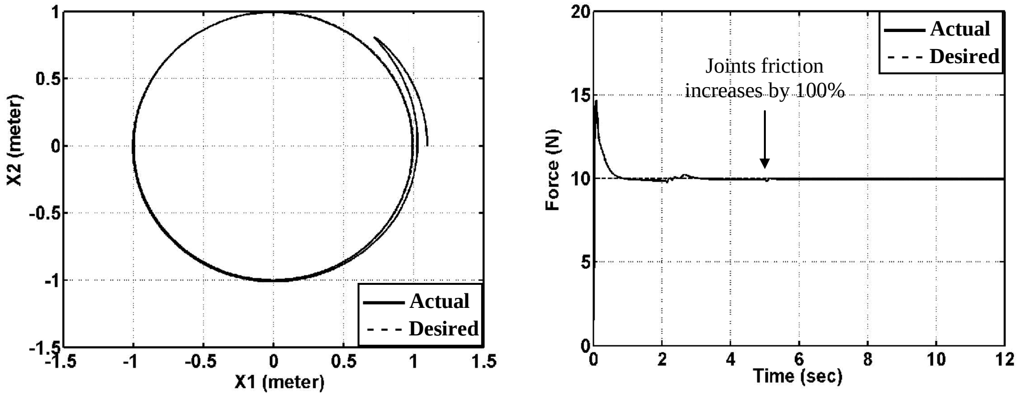
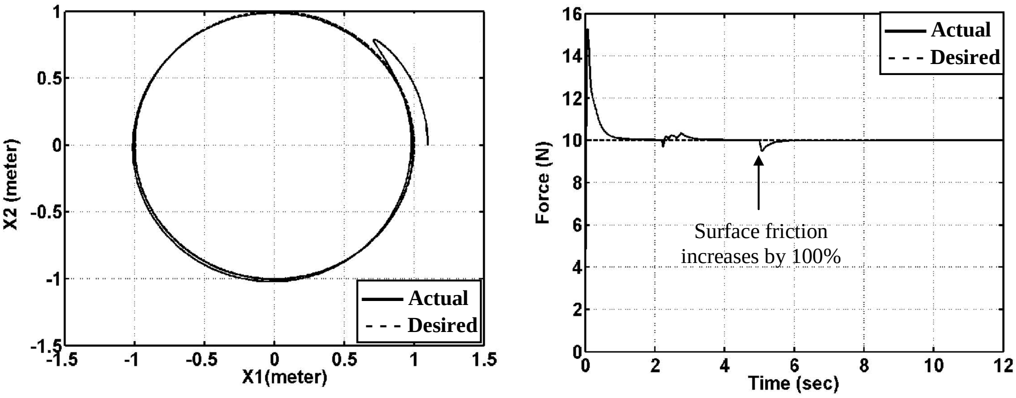


Fig. 14- The joint torques for the case of Fig. 10



Joints friction increases by 100%

Fig. 15- Position and force control with 100% increase in joint friction



Surface friction increases by 100%

Fig. 16- Position and force control with 100% increase in surface friction

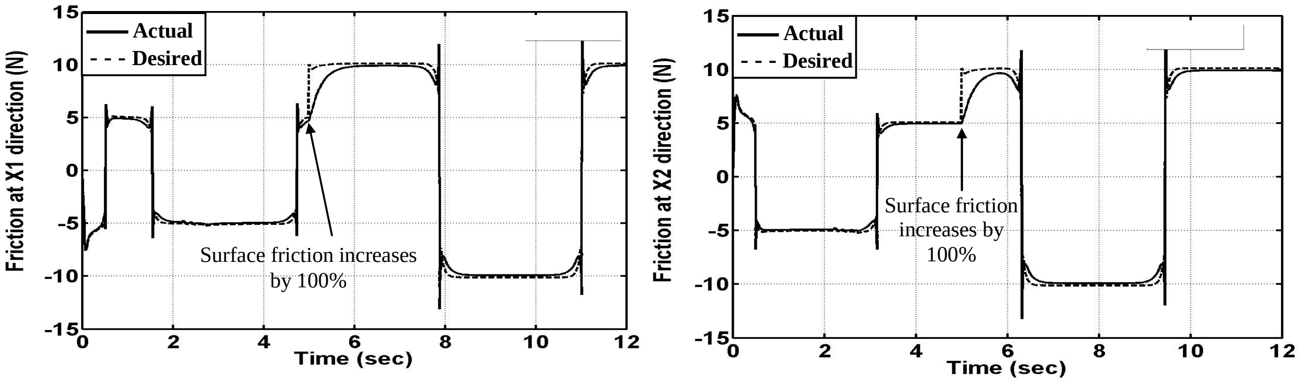


Fig. 17- Friction forces in x_1 and x_2 directions (actual and estimated)

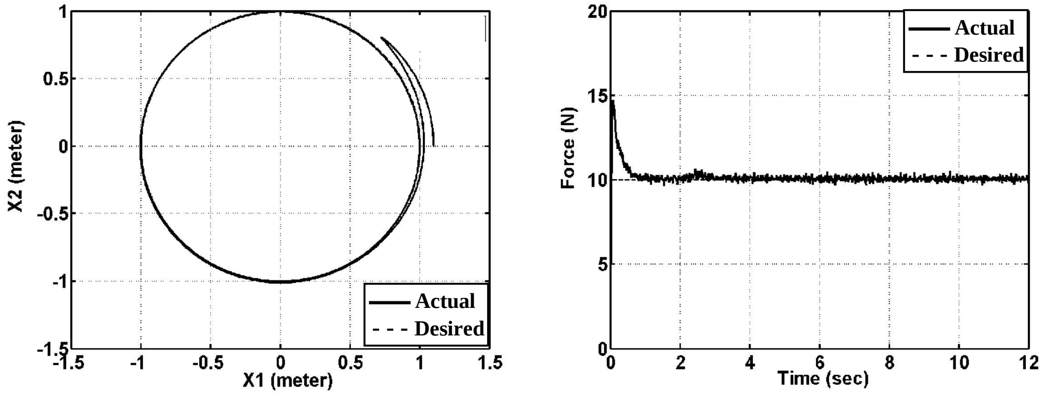


Fig. 18- Position and force control with 10% measurement noise in position and force sensors

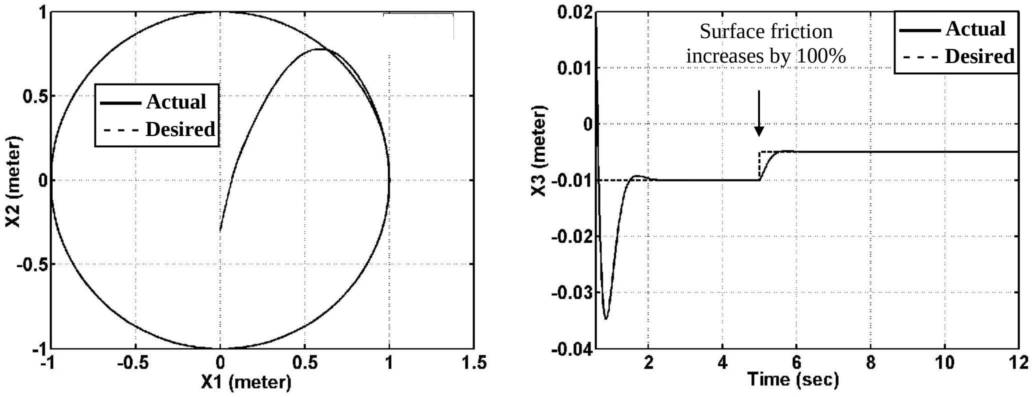


Fig. 19- Position and force control with 100% increase in surface stiffness coefficient (computed torque method)

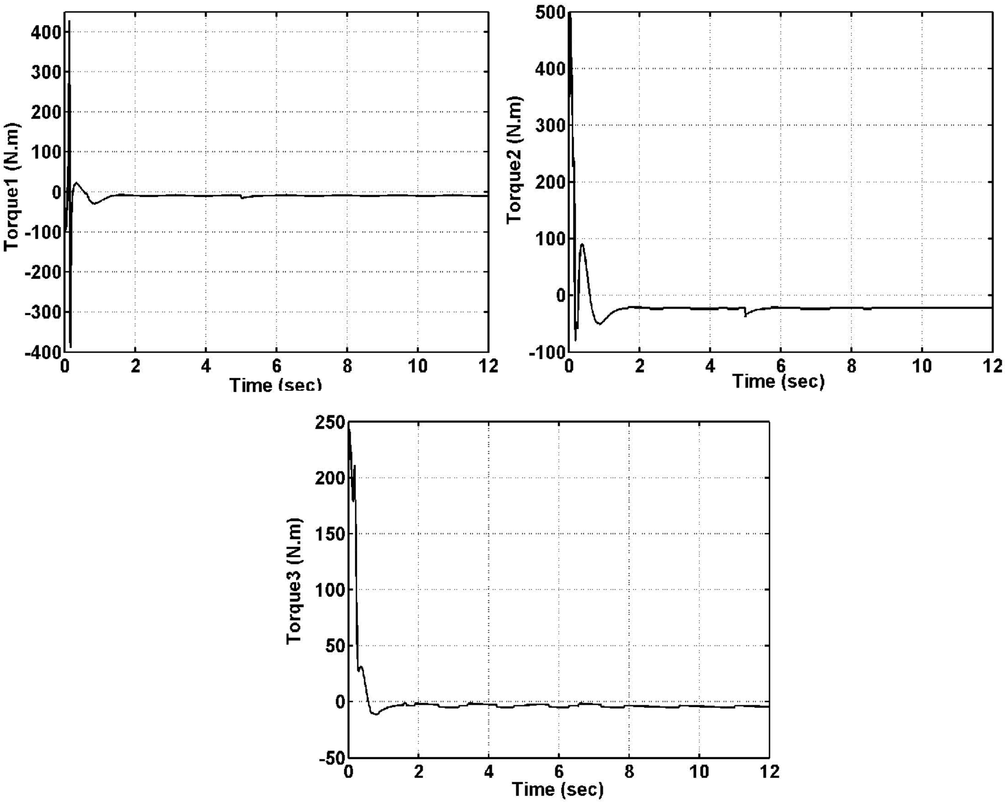


Fig. 20- The joint torques for the case of Fig. 16

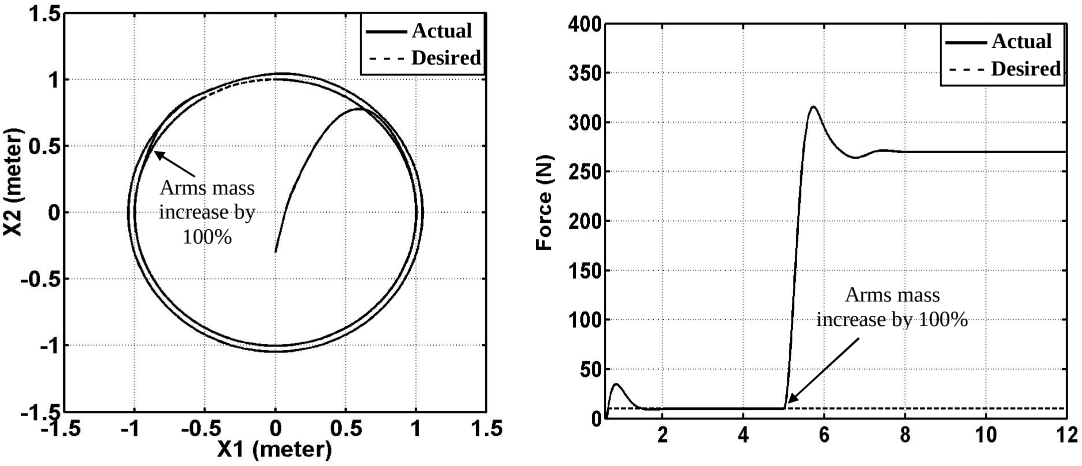


Fig. 21- Position and force control with 100% increase in arms mass (computed torque method)

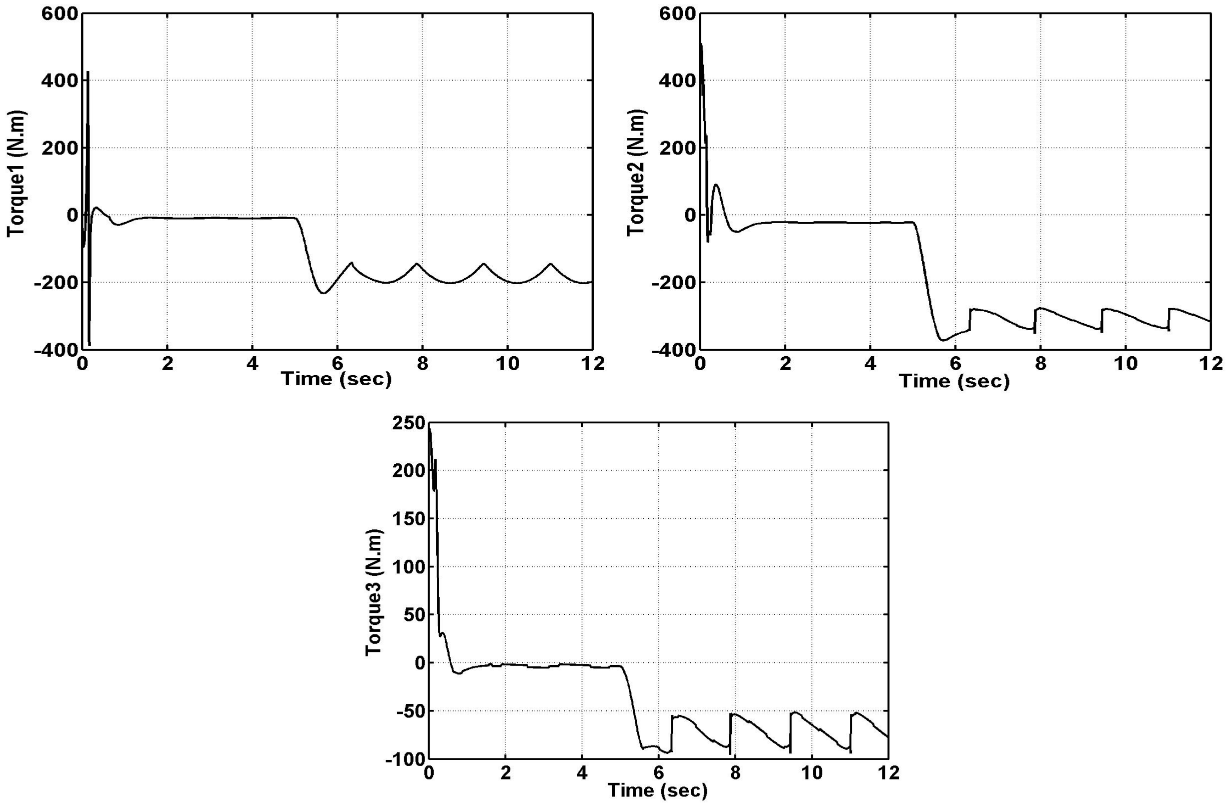


Fig. 22- The joint torques for the case of Fig. 18

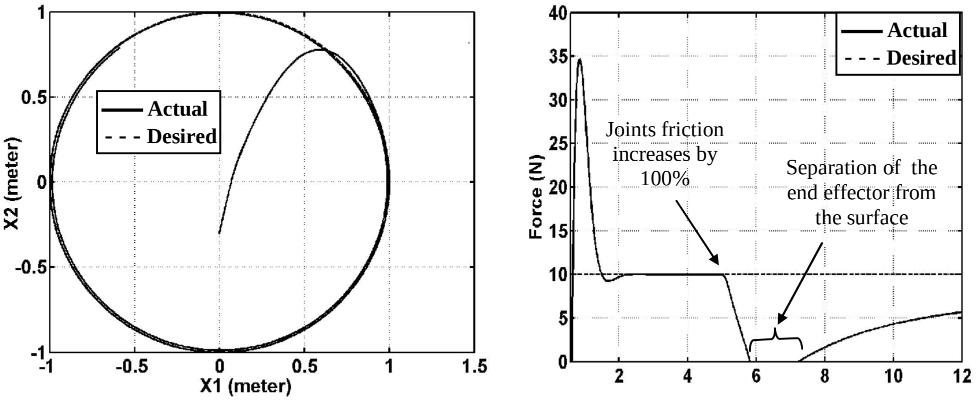


Fig. 23- Position and force control with 100% increase in surface friction (computed torque method)

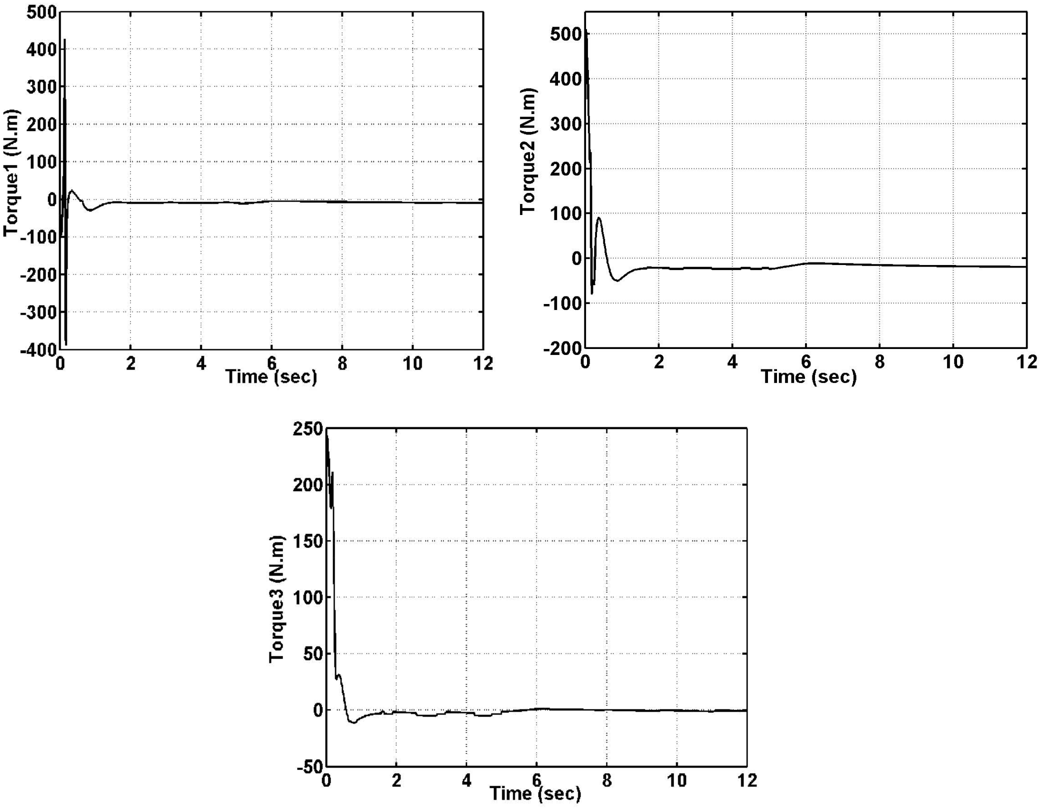


Fig. 24- The joint torques for the case of Fig. 20

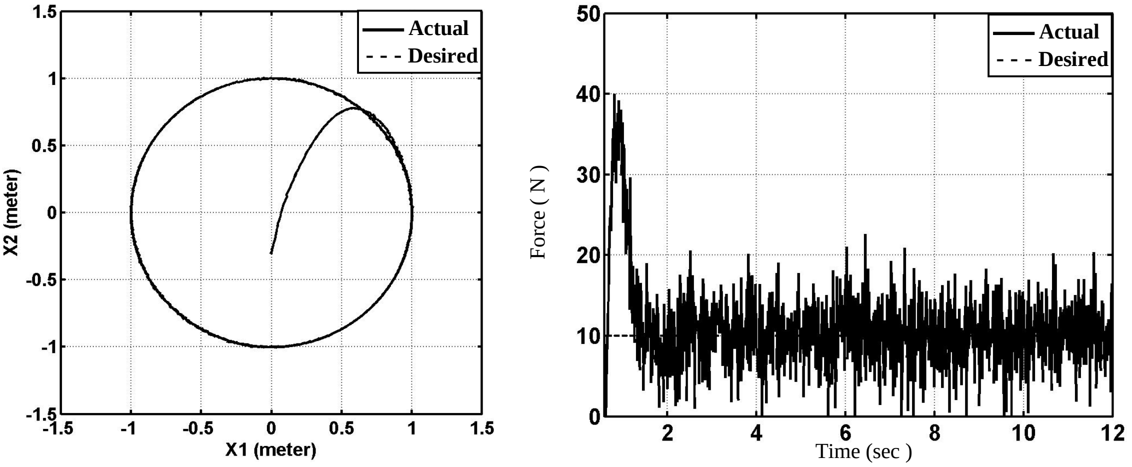


Fig. 25- Position and force control with 10% measurement noise in position and force sensors (computed torque method)

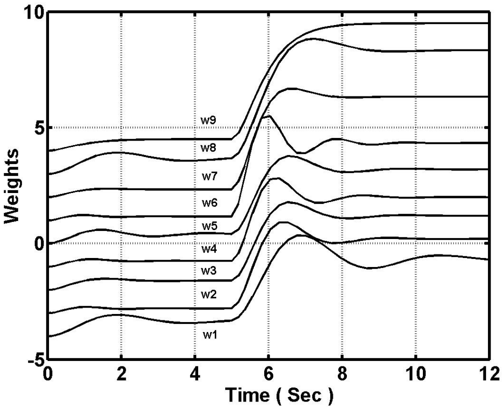


Fig. 26- Adaptive weight changes of one of the controllers during operation of the robot, when 100% increase in arms masses occurs at *t*=5 s.

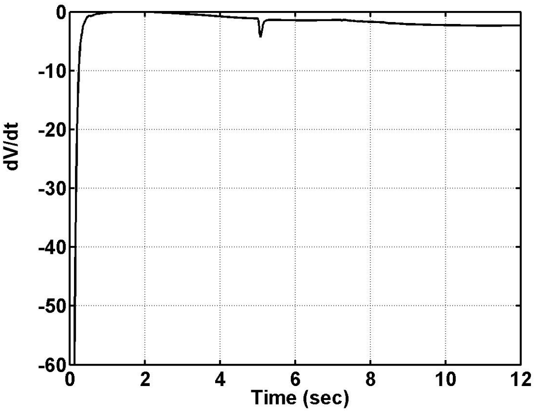


Fig. 27- Derivative of the Lyapunov function, defined in Eq. (33), during operation of the robot, when 100% increase in arms masses occurs at *t*=5 s.

7- Outline of the Future Work

The direction of the future work for this paper can be outlined as follows:

1- Performing a robustness analysis on the designed controller as well as the surface friction compensator.

2- Considering singular points in the work space, when the robot is performing position/force control.

3- Evaluation of the proposed controller when encountering an uneven surface, in which the surface friction force must be decomposed into three components along three axes.

4- Estimating the speed of the end effector, instead of measuring it, that usually is associated with noises created by the speed sensor.

8- 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. In addition, an on-line estimator was designed for adaptive computation of the stiffness coefficient of the surface. 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 is 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 a very fast and appropriate method for different applications of robot manipulators. Moreover, it was proved that the adaptive parameters are convergent under the proposed method. Simulation results showed good performance of the proposed controller even for some large changes in the robot dynamics as well as the surface parameters as compared with computed torque method.

Although the paper addresses the position/force control of robot manipulators with three DoF, but the proposed method can be easily extended to other manipulators with different DoF. Moreover, due to the adaptive neuro-fuzzy modeling, 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.

9- References

[1] M. H. Raibert and J. J. Craig, “Hybrid position/force control of manipulators,” *ASME J. Dyn. Meas. Contr.*, 1981.

[2] N. Hogan, “Impedance control: an approach to manipulator, part i, ii, iii,” *ASME J. Dyn. Meas. Contr.*, vol. 3, pp. 1-24, 1985.

[3] F. Y. Hsu and L. C. Fu, “A new design of adaptive fuzzy hybrid force/position controller for robot manipulators,” *IEEE Int. Conf. Robot. Automat.*, pp. 863-868, 1995.

[4] F. Y. Hsu and L. C. Fu, “Adaptive fuzzy hybrid force/position control for robot manipulators following contours of an uncertain object,” *IEEE. Int. Conf. Robot. Automat.*, pp. 2232-2237, Apr. 1996.

[5] F. Y. Hsu and L. C. Fu, “Intelligent robot deburring using adaptive fuzzy hybrid position/force control,” *IEEE Trans. Robot. Automat.*, vol.16, no.4, pp. 325-335, Aug. 2000.

[6] K. Kiguchi and T. Fukuda, “Fuzzy neural friction compensation method of robot manipulation during position/force control,” *IEEE Int. Conf. Robot. Automat.*, pp. 372-377, Apr. 1996.

[7] K. Kiguchi and T. Fukuda, “Intelligent position/force controller for industrial robot manipulators -application of fuzzy neural networks,” *IEEE Trans. Ind. Electrons.*, vol. 44, no. 6, pp. 753-761, Dec. 1997.

[8] K. Kiguchi, K. Watanabe, K. Izumi and T. Fukuda, “Two-stage adaptation of a position/force robot controller – application of soft computing techniques,” *IEEE Int. Conf. Knowledge-Based Intelligent Information Eng. Syst.*, pp. 141-144, Aug. 1999.

[9] K. Kiguchi and T. Fukuda, “Position/Force control of robot manipulators for geometrically unknown objects using fuzzy neural networks,” *IEEE Trans. Ind. Electrons.*, vol. 47, no. 3, pp. 641-649, June. 2000.

[10] D. Xiao, B. K. Ghosh, N. Xi and T. J. Tran, “Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment,” *IEEE Trans. Contr. Syst. Technol.*, vol. 8, no. 4, pp. 635-645, July. 2000.

[11] F. L. Lewis, C. T. Abdallah, and D. N. Dawson, “*Control of Robot Manipulators*,” Macmillan Publishing Co., 1993.

[12] J. J. Craig, “*Introduction to Robotic: Mechanics and Control*,” Addison-Wesley Publishing Co., 1989.

[13] S.T. Lin and A. K. Huang, “Hierarchical fuzzy force control for industrial robots,” *IEEE Trans. Ind. Electrons.*, vol. 45, no. 4, pp. 646-653, Aug. 1998.

[14] S. Jung and T. C. Hisa, “On neural network application to robust impedance control of robot manipulators,” *IEEE Int. Conf. Robot. Automat.*, pp. 869-874, 1995.

[15] S. Jung and T.C. Hsia, “Neural network impedance force control of robot manipulators,” *IEEE Trans. Ind. Electrons.*, vol. 45, no. 3, pp. 451-461, Jun. 1998.

[16] B. Bona, and M. Indri, “Friction compensation and robustness issues in force/position controlled manipulators,” *IEE Proc. Cont. Theory Applicat.*, vol. 142, no. 6, pp. 569-574, Nov. 1995.

[17] P. Tomei, “Robust adaptive friction compensation for tracking control of robot manipulators,” *IEEE Trans. Automat. Contr.*, vol. 45, no. 6, pp. 2164-2169, 2000

[18] S. N. Huang, K. K. Tan, and T. H. Lee, “Adaptive friction compensation using neural network approximations,” *IEEE Trans. Syst., Man, Cybern. C*, vol. 30, no. 4, pp. 551-557, Nov. 2000.

[19] R. R. Selmic and F. L. Lewis, “Neural-network approximation of piecewise continuous functions: application to friction compensation,” *IEEE Trans. Neural Networks*, vol. 13, no. 3, pp. 745-751, May. 2002.

[20] R. G. Berstecher, R. Palm, and H. D. Unbehauen, “An adaptive fuzzy sliding-mode controller,” *IEEE Trans. Ind. Electrons.*, vol. 48, no. 1, pp. 18-31, Feb. 2001.

[21] R. Amirat, M. Indri, T. Stomboli and B. Bona, “Experiments on robust friction compensation, the inverted pendulum case,” *Proc. American Cont. Conf.*, June. pp. 3303-3305, 1995.

[22] E. C. Park, H. Lim, and C. H. Choi, “Position control of x–y table at velocity reversal using presliding friction characteristics,” *IEEE Trans. Contr. Syst. Technol.*, vol. 11, no.1, pp. 24-31, Jan. 2003.

[23] T. Sugie, “Robust controller design for robot manipulators,” *Trans. ASME Dyn. Meas. and Contr.*, vol. 110, no.1, pp. 94-96, Mar. 1988.

[24] S. Jung, T. C. Hsia, and R. G. Bonitz, “Force tracking impedance control for robot manipulators with an unknown environment: theory, simulation, and experiment,” *The International Journal of Robotics Research*, vol. 20, no.9, pp. 765-774, Sep. 2001.