2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Fuzzy Neural Network-based Adaptive Impedance Force Control Design of Robot Manipulator under Unknown Environment

Wei-Chen Wang and Ching-Hung Lee *Senior Member, IEEE*

***Abstract*—In this paper, an adaptive impedance force control scheme for an *n*-link robot manipulator under unknown environment is proposed. The system dynamics of the robot manipulator is assumed that system model is not exactly known or has system uncertainty. Therefore, the traditional adaptive impedance force controller is not valid. Herein, the fuzzy neural networks are adopted to estimate the system model terms of robot and the force tracking control is developed by the proposed adaptive scheme. The proposed scheme is established by gradient descent approach. Using the Lyapunov stability theory, the update laws of fuzzy neural networks can be derived and the stability of the closed-loop system is guaranteed. Finally, simulation results of a two-link robot manipulator with environment constraint are introduced to illustrate the performance and effectiveness of our approach.**

I. Introduction

When a robot manipulator performs a task in contact with environment, the end-effector and the environment yields force between each others. The force impedance control is one of the complicated control problems. To treats the control problem, the position and force control are required. For moving payloads or painting objects tasks, these tasks are only to follow the desired trajectories. However, for during grinding or deburring tasks, interacting forces should be developed between the robot manipulator and the working environment. Consequently, the position and interacting force of end-effector must be controlled. In addition, the controller must be robust to deal with the unknown environment stiffness and position.

Within the impedance force control, several approaches constrained motion controls have been suggested, such as impedance control and the hybrid force control algorithms [1, 2]. The impedance control of robot manipulators is to adjust the end-effector position and sense the contact force in response such that a second order mass-spring-damper system is satisfied the target impedance behavior. Several robust control schemes and adaptive control strategies have been proposed [3-8]. In literature [9-12], adaptive impedance control schemes for an *n*-link robot manipulator without using regressor and approximation unknown system parameter based on the function approximation technique are introduced. Therefore, the lack of force tracking capability of the impedance control method has been attention by many researchers to solve the direct force tracking problem. The modification of impedance functions have been proposed to solve the force tracking problem on unknown environment [13].

For impedance controller design, the robust problem is important for robots with the uncertainties and under unknown environment. Recently, many researchers proposed several approaches to deal with the controller design for the systems with uncertainties and disturbance. The intelligent controllers using neural network (NNs) have been proposed, which can regulate impedance properties through the learning of NNs in considering of robot uncertainties. In [15, 16], the NNs play the rule of compensator to treat the system uncertainties in force control problem. In addition, the on-line learning method is proposed to regulate all impedance parameters as well as desired trajectory at the same time [17]. Similar to the NNs, the fuzzy logic control was developed by human experience to implement nonlinear system algorithms [18-20]. Besides, the fuzzy neural network (FNNs) can be introduced to deal with uncertainties and disturbances [21]. In this paper, a FNN-based adaptive impedance force control is proposed for robot manipulator.

Herein, an adaptive impedance force control scheme for an *n*-link robot manipulator under unknown environment is proposed. The dynamic model of the robot manipulator is assumed to be unknown or unavailable, since the system model uncertainties are assumed to be time dependent and their variation bounds ate not practically available. Therefore, we cannot solve the impedance force control by using the traditional adaptive approach. Thus, the estimators of FNNs are adopted to estimate the system model's matrices and the force tracking control is developed. Based on the Lyapunov stability theory, the update laws of fuzzy neural networks can be derived and the stability of the closed-loop system is guaranteed. Finally, simulation results of a 2-DOF robot manipulator with environment constraint are introduced to illustrate the performance and effectiveness of our approach.

The rest of this paper is as follows. Section II introduces the problem formulation and the used fuzzy neural network systems. The proposed FNN-based adaptive impedance force control scheme is introduced in Section III. In Section IV, simulation results of a two-link robot manipulator are presented to show the effectiveness of our approach. Finally, conclusion is given.

Ching-Hung Lee and Wei-Chen Wang are with the Department of Mechanical Engineering, National Chung Hsing University, Taichung, Taiwan, R.O.C. (Tel: +886-4-22840433, ext: 417; fax: +886-422877170; e-mail: chleenchu@dragon.nchu.edu.tw).

This work was supported in part by the National Science Council, Taiwan, R.O.C., under contracts NSC-100-2221- E-005-093-MY2.

II. Problem Formulation

A. System Description

The dynamic model of an n-link rigid robot manipulator in joint space coordinate can be given by

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}}+\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\mathbf{g}(\mathbf{q})=\boldsymbol{\tau}-\mathbf{J}^T\mathbf{F}_{ext} \tag{1}$$

where $\mathbf{q}\in\Re^n$ denotes the vector of generalized displacement in robot coordinates, $\boldsymbol{\tau}\in\Re^n$ is the vector of joint torque, $\mathbf{D}(\mathbf{q})\in\Re^{n\times n}$ is the inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}\in\Re^n$ is the vector of centrifugal and Coriolis forces, $\mathbf{g}(\mathbf{q})\in\Re^n$ is the gravity vector. $\mathbf{J}(\mathbf{q})$ is the $n\times n$ Jacobian matrix that must be square and invertible, and $\mathbf{F}_{ext}\in\Re^n$ is the external force at the end-effector. The corresponding Cartesian space representation is

$$\mathbf{D}_x(\mathbf{x})\ddot{\mathbf{x}}+\mathbf{C}_x(\mathbf{x},\dot{\mathbf{x}})\dot{\mathbf{x}}+\mathbf{g}_x(\mathbf{x})=\mathbf{J}^{-T}\boldsymbol{\tau}-\mathbf{F}_{ext} \tag{2}$$

where

$$\mathbf{D}_x(\mathbf{x})=\mathbf{J}^{-T}\mathbf{D}(\mathbf{q})\mathbf{J}^{-1} \tag{3}$$

$$\mathbf{C}_x(\mathbf{x},\dot{\mathbf{x}})=\mathbf{J}^{-T}(\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})-\mathbf{D}(\mathbf{q})\mathbf{J}^{-1}\dot{\mathbf{J}})\mathbf{J}^{-1} \tag{4}$$

$$\mathbf{g}_\mathbf{x}(\mathbf{x})=\mathbf{J}^{-\mathbf{T}}\mathbf{g}(\mathbf{q}) \tag{5}$$

From the results of literature [9-11], the system uncertainties are assumed to be time dependent and their variation bounds ate not practically available, thus estimation of the robot model are needed for the controller design. In this paper, this problem will be solved by the fuzzy neural network.

In addition, the control goal is to design an adaptive control scheme to generate the proper control signals such that the end-effector of the robot manipulator, with unknown parameters and under unknown environment, follows the design trajectories of position and force for free space and contact space, respectively.

B. Fuzzy Neural Network

Herein, we introduce the fuzzy neural network (FNN) systems [18-20]. The schematic diagram of the used FNN is shown in Fig. 1. There are four layers. Layer 1 accepts input variables and its nodes represent fuzzy input linguistic variables. The nodes in this layer only transmit input variables to the next layer directly, i.e., $O_i^{(1)}(k)=x_i(k)$. Layer 2 is used to calculate the corresponding Gaussian membership value, i.e.,

$$O_{ij}^{(2)}(k)=\exp\left[-\frac{\left(O_i^{(1)}(k)-m_{ij}\right)^2}{\left(\sigma_{ij}\right)^2}\right] \tag{6}$$

where m_{ij} and σ_{ij} are the center and the width of the Gaussian function. Nodes in layer 2 represent the term-nodes of the respective linguistic variables. In layer 3, each node represents the fuzzy rule and the t-norm product operation is adopted. Links before layer 3 represent the preconditions of the rules, and the links after layer 3 represent the consequences of the rule node, i.e.,

$$O_j^{(3)}(k)=\prod_i O_{ij}^{(2)}(k). \tag{7}$$

Layer 4 is the output layer. Each node is for actual output of this FNN system, which is connected by weighting value w_j, the pth output is represented as

$$y_p=O^{(4)}(k)=\mathbf{w}^T\boldsymbol{\psi}=\frac{\sum_{j=1}^{R}w_jO_j^{(3)}(k)}{\sum_{j=1}^{R}O_j^{(3)}(k)} \tag{8}$$

where $\mathbf{w}=[w_1\ w_2,\ \ldots,\ w_R]^T$ is the weighting vector and $\boldsymbol{\psi}=\left[\boldsymbol{\psi}^1\ \boldsymbol{\psi}^2,\ldots,\boldsymbol{\psi}^R\right]$, $\boldsymbol{\psi}^j=O_j^{(3)}(k)\Big/\sum_j^R O_j^{(3)}(k)$ represents the normalized value and R is the chosen rule number.

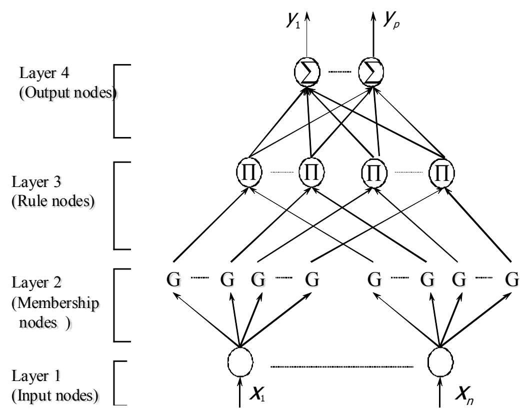

Figure 1: Schematic diagram of fuzzy neural network [18-20] (the "G" denotes the Gaussian membership function).

III. FNN-based Adaptive Impedance Force Control

As the results of traditional impedance control [22], the corresponding system parameters are assumed to be exactly known, and then the closed-loop system satisfies the target impedance relationship, i.e.,

$$\mathbf{M}_m(\ddot{\mathbf{x}}-\ddot{\mathbf{x}}_d)+\mathbf{B}_m(\dot{\mathbf{x}}-\dot{\mathbf{x}}_d)+\mathbf{K}_m(\mathbf{x}-\mathbf{x}_d)=-\mathbf{F}_{ext} \tag{9}$$

where $\mathbf{M}_m$, $\mathbf{B}_m$, and $\mathbf{K}_m$ are diagonal $n\times n$ matrices, $\mathbf{x}_d\in\Re^n$ is the desired end-point trajectory. This means that the system trajectory follows the desired trajectory for free space phase ($\mathbf{F}_{ext}$=0). In practical cases, the system parameters are usually not exactly known and the traditional impedance controller cannot be derived, therefore we will consider

$$\mathbf{M}_m(\ddot{\mathbf{x}}_m-\ddot{\mathbf{x}}_d)+\mathbf{B}_m(\dot{\mathbf{x}}_m-\dot{\mathbf{x}}_d)+\mathbf{K}_m(\mathbf{x}_m-\mathbf{x}_d)=-\mathbf{F}_{ext} \quad (10)$$

where $\mathbf{x}_m \in \Re^n$ is the state vector of (10).

Our control goal is to design an adaptive impedance controller to derive $\mathbf{x}$ approaches to $\mathbf{x}_m$ asymptotically. Thus, the new target impedance (10) converges to traditional one (9). In addition, we should pay additional effort to design the adaptive force controller such that $\mathbf{F}_{ext}$ is equal to the desired force $\mathbf{F}_d$ for contact space. From the results of [9-12], the environment stiffness is difficulty of known then we cannot design the reference trajectory $\mathbf{x}_r$ to achieve the desired force. Herein, we will solve the problem by FNN-based adaptive impedance force control. Herein, we first introduce the adaptive force control scheme.

Adaptive Force Control Scheme

As above description, our goal is to achieve $\mathbf{F}_{ext}$=$\mathbf{F}_d$. In contact cases, the end-effector of robot manipulator satisfies $\mathbf{x}_d$=$\mathbf{x}_e$ and the relationship between force and environment. We know the inertia and damping parameters only influence the transient response of the end-effector [24]

$$-\mathbf{F}_{ext} \cong \mathbf{K}_e(\mathbf{x}_m-\mathbf{x}_e)\,. \quad (11)$$

Thus, we have the desired trajectory force $\mathbf{F}_d$ as

$$-\mathbf{F}_d \cong \mathbf{K}_e(\mathbf{x}_m^*-\mathbf{x}_e)\,. \quad (12)$$

where $\mathbf{x}_m^*$ is desired trajectory such that $\mathbf{F}_{ext}$=$\mathbf{F}_d$. According equations (12) and (13), we have

$$\mathbf{F}_d-\mathbf{F}_{ext} \cong \mathbf{K}_e(\mathbf{x}_m^*-\mathbf{x}_m)\,. \quad (13)$$

However, the stiffness parameter $\mathbf{K}_e$ is usually unknown, therefore $\mathbf{x}_m^*$ cannot be obtained by equation (12) directly. Therefore, the estimation of the stiffness should be designed, thus (13) is modified as

$$\mathbf{F}_d-\mathbf{F}_{ext} \cong \hat{\mathbf{K}}_e(\mathbf{x}_m^*-\mathbf{x}_m)\,, \quad (14)$$

where $\hat{\mathbf{K}}_e$ is the estimated stiffness parameter. To achieve the adaptation of $\hat{\mathbf{K}}_e$, we first define the force error and the corresponding objective function to be minimized as

$$\mathbf{v}=\mathbf{F}_d-\mathbf{F}_{ext} \quad (15)$$

and

$$\Im=\frac{1}{2}\mathbf{v}^T\mathbf{v}\,. \quad (16)$$

For simplicity, we consider that force is applied to only one direction. Then, let $\hat{K}_e$ be an element of $\hat{\mathbf{K}}_e$ and $v=f_d-f_{ext}$, the corresponding gradient of $\Im$ is

$$\frac{\partial \Im}{\partial \hat{K}_e}=\left[\frac{\partial v}{\partial \hat{K}_e}\right]^T v\,. \quad (17)$$

Thus, the following update law is chosen

$$\hat{K}_e(k+1)=\hat{K}_e(k)+\Delta\hat{K}_e(k)=\hat{K}_e(k)+(-\eta\frac{\partial \Im}{\partial \hat{K}_e}) \quad (18)$$

where η is the learning rate. For the training process, the learning rate plays an important role for this case. A small value of η leads a slower convergence and a large value may have unstable result. Hence, the selection of η is important but it is not easy to solve. Then, the Lyapunov approach is adopted to guarantee the convergence of $\hat{K}_e$. Rewrite equation (16) as

$$V(k)=\frac{1}{2}v^2(k)=\frac{1}{2}(f_d(k)-f_{ext}(k))^2=\frac{1}{2}e_f^2(k) \quad (19)$$

where $e_f(k)=f_d(k)-f_{ext}(k)$ and k is time-instant for parameter update. Let $\Delta f(k)=f_d(k)-f_{ext}(k)$ and then we have

$$\begin{aligned}\Delta V(k)&=V(k+1)-V(k)\\&=\frac{1}{2}[e_f(k+1)+e_f(k)][e_f(k+1)-e_f(k)]\\&=\frac{1}{2}[e_f(k)+\Delta f(k)]\cdot\Delta f(k)\\&=\frac{\Delta f^2(k)}{2}+e_f(k)\cdot\Delta f(k).\end{aligned}$$

(20)

By the gradient result, $\Delta f(k)\approx\frac{\partial e_f}{\partial \hat{K}_e}\Delta\hat{K}_e$ and $\Delta\hat{K}_e=-\eta(\frac{\partial V}{\partial \hat{K}_e})$, we obtain

$$\Delta f(k)=\frac{\partial e_f}{\partial \hat{K}_e}\left(-\eta\frac{\partial V}{\partial \hat{K}_e}\right)=-\eta v\left(\frac{\partial e_f}{\partial \hat{K}_e}\right)^2$$

(21)

and

$$\Delta V(k)=\frac{\eta^2 v^2}{2}\left(\frac{\partial e_f}{\partial \hat{K}_e}\right)^4-\eta v^2\left(\frac{\partial e_f}{\partial \hat{K}_e}\right)^2 = -\eta v^2\left(\frac{\partial e_f}{\partial \hat{K}_e}\right)^2\cdot\left[1-\frac{1}{2}\eta\left(\frac{\partial e_f}{\partial \hat{K}_e}\right)^2\right]. \tag{22}$$

As above, we can obtain the stability condition for selection of learning rate η

$$0<\frac{\eta}{2}\left(\frac{\partial e_f}{\partial \hat{K}_e}\right)^2<1 \tag{23}$$

this implies $\Delta V(k)<0,\ \forall k$. Then, the stiffness parameter can be obtained. To accomplish force control, we should design the suitable $\mathbf{x}_m$ for contact space. From (14), we have the variation of desired position

$$\Delta \mathbf{x}_m=\frac{\Delta \mathbf{F}}{\hat{K}_e} \tag{24}$$

where $\Delta \mathbf{x}_m = x_m^* - x_m$ and $\Delta \mathbf{F}=\mathbf{F}_d-\mathbf{F}_{ext}$, then we design the desired trajectory to achieve force control as

$$\mathbf{x}_m^*(k+1)=\mathbf{x}_m(k)+\Delta\mathbf{x}_m(k)\ . \tag{25}$$

This guarantees the convergence of force tracking error even the stiffness gain K_e is unknown, i.e., $\mathbf{F}_{ext}\to\mathbf{F}_d$. Figure 2 summaries the proposed adaptive force control scheme.

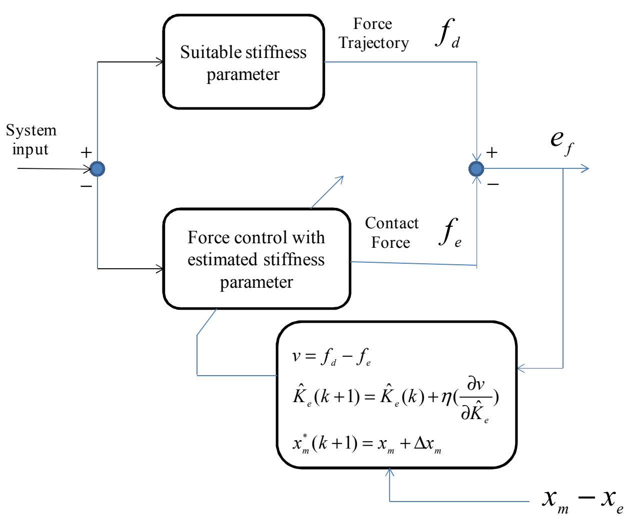

Figure 2: The block diagram of force tracking scheme.

After the desired trajectory $\mathbf{x}_m^*$ being desired, the reminding work is to consider the tracking control of robot manipulator. The system parameter **D**, **C**, and **g** are not exactly known in real world applications. We define the state error $\mathbf{e}=\mathbf{x}-\mathbf{x}_m$ and error vector $\mathbf{s}=\dot{\mathbf{e}}+\Lambda\mathbf{e}$, $\Lambda=diag(\lambda_1,\lambda_2,...,\lambda_n)$ with $\lambda_i>0$ for i=1, …, n. Rewrite the robot dynamics model (2) as

$$\hat{\mathbf{D}}_x\dot{\mathbf{s}}+\hat{\mathbf{C}}_\mathbf{x}\mathbf{s}+\hat{\mathbf{g}}_\mathbf{x}+\hat{\mathbf{D}}_\mathbf{x}\ddot{\mathbf{x}}_\mathbf{m}-\hat{\mathbf{D}}_\mathbf{x}\Lambda\dot{\mathbf{e}}+\hat{\mathbf{C}}_\mathbf{x}\dot{\mathbf{x}}_\mathbf{m}-\hat{\mathbf{C}}_\mathbf{x}\Lambda\mathbf{e} = \mathbf{J}^{-T}\boldsymbol{\tau}-\mathbf{F}_{ext}\ . \tag{26}$$

And the designed controller is

$$\boldsymbol{\tau}=\mathbf{J}^T(\mathbf{F}_{ext}+\hat{\mathbf{g}}_\mathbf{x}+\hat{\mathbf{D}}_\mathbf{x}\ddot{\mathbf{x}}_\mathbf{m}-\hat{\mathbf{D}}_\mathbf{x}\Lambda\dot{\mathbf{e}}+\hat{\mathbf{C}}_\mathbf{x}\dot{\mathbf{x}}_\mathbf{m}- \hat{\mathbf{C}}_\mathbf{x}\Lambda\mathbf{e}-\mathbf{K}_\mathbf{d}(\dot{\mathbf{e}}+\Lambda\mathbf{e})) \tag{27}$$

where $\hat{\mathbf{D}}_\mathbf{x}$, $\hat{\mathbf{C}}_\mathbf{x}$, and $\hat{\mathbf{g}}_x$ are the estimations of $\mathbf{D}_\mathbf{x}$, $\mathbf{C}_\mathbf{x}$, and $\mathbf{g}_\mathbf{x}$ using FNNs, respectively. They can be represented by according equation (8)

$$\hat{\mathbf{D}}_\mathbf{x}=\hat{\mathbf{w}}_D^T\boldsymbol{\psi}_D \tag{28a}$$

$$\hat{\mathbf{C}}_\mathbf{x}=\hat{\mathbf{w}}_C^T\boldsymbol{\psi}_C \tag{28b}$$

$$\hat{\mathbf{g}}_\mathbf{x}=\hat{\mathbf{w}}_g^T\boldsymbol{\psi}_g\ . \tag{28c}$$

According to the universal approximation of the FNNs, the unknown functions are represented as follows with approximation error ε, i.e.,

$$\mathbf{D}_\mathbf{x}=\mathbf{w}_D^{*T}\boldsymbol{\psi}_D+\varepsilon_{\mathbf{D}_X} \tag{28d}$$

$$\mathbf{C}_\mathbf{x}=\mathbf{w}_C^{*T}\boldsymbol{\psi}_C+\varepsilon_{\mathbf{C}_X} \tag{28e}$$

$$\mathbf{g}_\mathbf{x}=\mathbf{w}_g^{*T}\boldsymbol{\psi}_g+\varepsilon_{\mathbf{g}_X} \tag{28f}$$

where $\mathbf{w}_D^*\in\Re^{n^2R}$, $\mathbf{w}_C^*\in\Re^{n^2R}$, and $\mathbf{w}_g^*\in\Re^{nR}$ are weighting matrices and $\boldsymbol{\psi}_D\in\Re^{n^2R}$, $\boldsymbol{\psi}_C\in\Re^{n^2R}$, $\boldsymbol{\psi}_G\in\Re^{nR}$ are the firing strength of the corresponding fuzzy rules. R represents the number of fuzzy rule of the FNN. Substituting (27) to equation (2)

$$\mathbf{D}_x\dot{\mathbf{s}}+\mathbf{C}_x\mathbf{s}+\mathbf{K}_\mathbf{d}\mathbf{s}=\left(\hat{\mathbf{D}}_\mathbf{x}-\mathbf{D}_\mathbf{x}\right)\left(\ddot{\mathbf{x}}_\mathbf{m}-\Lambda\dot{\mathbf{e}}\right) +\left(\hat{\mathbf{C}}_\mathbf{x}-\mathbf{C}_\mathbf{x}\right)\left(\dot{\mathbf{x}}_\mathbf{m}-\Lambda\mathbf{e}\right)+\left(\hat{\mathbf{g}}_x-\mathbf{g}_x\right). \tag{29}$$

If we can design the proper update laws of FNNs such that $\hat{\mathbf{D}}_\mathbf{x}\to\mathbf{D}_\mathbf{x}$, $\hat{\mathbf{C}}_x\to\mathbf{C}_x$, and $\hat{\mathbf{g}}_\mathbf{x}\to\mathbf{g}_\mathbf{x}$. Hence the closed-loop system becomes stable and the tracking error will approach to zero by the following discussion. Rewrite (30) as

$$\mathbf{D}_x\dot{\mathbf{s}}+\mathbf{C}_x\mathbf{s}+\mathbf{K}_\mathbf{d}\mathbf{s}=0 \tag{30}$$

where $\mathbf{K}_\mathbf{d}$ is positive definite. We select the Lyapunov candidate function as

$$V_1=\frac{1}{2}\mathbf{s}^T\mathbf{D}_x\mathbf{s} \tag{31}$$

such that

$$\dot{V}_1 = -\mathbf{s}^T \mathbf{K}_d \mathbf{s} + \frac{1}{2}\mathbf{s}^T \left(\dot{\mathbf{D}}_x - 2\mathbf{C}_x\right)\mathbf{s}. \tag{32}$$

Since $\dot{\mathbf{D}}_x - 2\mathbf{C}_x$ is skew-symmetric, we have

$$\dot{V}_1 = -\mathbf{s}^T \mathbf{K}_d \mathbf{s} \le 0 . \tag{33}$$

It is simple to prove that $\mathbf{s} \in L_\infty \cap L_2$, and $\dot{\mathbf{s}} \in L_\infty$. Thus, $\mathbf{s}$ converges to zero when t approaches to infinity, this implies the tracking error $\mathbf{e}$ will approaches to zero as $t \to \infty$. This completes the proof of tracking control problem. Therefore, the remained work is the stability analysis and the convergence of the FNNs.
For the system with uncertainties, equation (29) can be rewritten as

$$\begin{aligned}&\mathbf{D}_x \dot{\mathbf{s}} + \mathbf{C}_\mathbf{x} \mathbf{s} + \mathbf{K}_\mathbf{d} \mathbf{s} = \tilde{\mathbf{w}}_D^T \boldsymbol{\psi}_D \left(\ddot{\mathbf{x}}_\mathbf{m} - \boldsymbol{\Lambda} \dot{\mathbf{e}}\right) + \\ &\tilde{\mathbf{w}}_C^T \boldsymbol{\psi}_C \left(\dot{\mathbf{x}}_\mathbf{m} - \boldsymbol{\Lambda} \mathbf{e}\right) + \tilde{\mathbf{w}}_g^T \boldsymbol{\psi}_g + \boldsymbol{\varepsilon}_1\end{aligned} \tag{34}$$

where $\tilde{\mathbf{w}}_{(.)} = \hat{\mathbf{w}}_{(.)} - \mathbf{w}^*_{(.)}$ and $\boldsymbol{\varepsilon}_1 = \boldsymbol{\varepsilon}_1(\boldsymbol{\varepsilon}_{\mathbf{D}_x}, \boldsymbol{\varepsilon}_\mathbf{C}, \boldsymbol{\varepsilon}_\mathbf{g}, \mathbf{s}, \ddot{\mathbf{x}}_i)$ is due to the approximation of $\mathbf{D_x}$, $\mathbf{C_x}$, and $\mathbf{g_x}$. Let us consider the Lyapunov candidate function

$$\begin{aligned}&V(\mathbf{s}, \tilde{\mathbf{w}}_\mathbf{D}, \tilde{\mathbf{w}}_\mathbf{C}, \tilde{\mathbf{w}}_\mathbf{g}) = \frac{1}{2}\mathbf{s}^T \mathbf{D}_x \mathbf{s} + \\ &\frac{1}{2} Tr(\tilde{\mathbf{w}}_\mathbf{D}^T \mathbf{Q}_\mathbf{D} \tilde{\mathbf{w}}_\mathbf{D} + \tilde{\mathbf{w}}_\mathbf{C}^T \mathbf{Q}_\mathbf{C} \tilde{\mathbf{w}}_\mathbf{C} + \tilde{\mathbf{w}}_\mathbf{g}^T \mathbf{Q}_\mathbf{g} \tilde{\mathbf{w}}_\mathbf{g})\end{aligned} \tag{35}$$

where $\mathbf{Q}_\mathbf{D} \in \Re^{n^2R \times n^2R}$, $\mathbf{Q}_\mathbf{C} \in \Re^{n^2R \times n^2R}$, $\mathbf{Q}_\mathbf{g} \in \Re^{nR \times nR}$ are positive definite matrices and $Tr(.)$ is the trace operation. Hence

$$\begin{aligned}\dot{V} = &-\mathbf{s}^T \mathbf{K}_d \mathbf{s} + \mathbf{s}^T \boldsymbol{\varepsilon}_1 \\ &- Tr[\tilde{\mathbf{w}}_D^T (\boldsymbol{\psi}_D \left(\ddot{\mathbf{x}}_\mathbf{m} - \boldsymbol{\Lambda}\dot{\mathbf{e}}\right)\mathbf{s}^T + \mathbf{Q}_\mathbf{D} \dot{\hat{\mathbf{w}}}_\mathbf{D}) \\ &+ \tilde{\mathbf{w}}_C^T (\boldsymbol{\psi}_C \left(\dot{\mathbf{x}}_\mathbf{m} - \boldsymbol{\Lambda}\mathbf{e}\right)\mathbf{s}^T + \mathbf{Q}_\mathbf{C} \dot{\hat{\mathbf{w}}}_\mathbf{C}) \\ &+ \tilde{\mathbf{w}}_g^T (\boldsymbol{\psi}_g \mathbf{s}^T + \mathbf{Q}_\mathbf{g} \dot{\hat{\mathbf{w}}}_\mathbf{g})]\end{aligned}$$
(36)

Since $\mathbf{s}^T[\dot{\mathbf{D}}_x(\mathbf{x}) - 2\mathbf{C}_x(\mathbf{x}, \dot{\mathbf{x}})]\mathbf{s} = 0$ for all $\mathbf{s} \in \Re^n$ and we choosing the update laws to be [25]

$$\dot{\hat{\mathbf{w}}}_\mathbf{D} = -\mathbf{Q}_\mathbf{D}^{-1}(\boldsymbol{\psi}_\mathbf{D}(\ddot{\mathbf{x}}_m - \boldsymbol{\Lambda}\dot{\mathbf{e}})\mathbf{s}^T + \vartheta_{\mathbf{D}x} \hat{\mathbf{w}}_{\mathbf{D}x}) \tag{37a}$$

$$\dot{\hat{\mathbf{w}}}_\mathbf{C} = -\mathbf{Q}_\mathbf{C}^{-1}(\boldsymbol{\psi}_\mathbf{C}(\dot{\mathbf{x}}_m - \boldsymbol{\Lambda}\mathbf{e})\mathbf{s}^T + \vartheta_{\mathbf{C}x} \hat{\mathbf{w}}_{\mathbf{C}x}) \tag{37b}$$

$$\dot{\hat{\mathbf{w}}}_\mathbf{g} = -\mathbf{Q}_\mathbf{g}^{-1}(\boldsymbol{\psi}_\mathbf{g}\mathbf{s}^T + \vartheta_{\mathbf{g}x} \hat{\mathbf{w}}_{\mathbf{g}x})$$
(37c)

where $\vartheta_{(.)} > 0$ and is used to smooth the controller [24]. Hence, equation (36) becomes

$$\begin{aligned}\dot{V} = &-\mathbf{s}^T \mathbf{K}_d \mathbf{s} + \mathbf{s}^T \boldsymbol{\varepsilon}_1 + \vartheta_{\mathbf{D}_x} Tr(\tilde{\mathbf{w}}_{D_x}^T \hat{\mathbf{w}}_{\mathbf{D}_x}) \\ &+ \vartheta_{\mathbf{C}_x} Tr(\tilde{\mathbf{w}}_{C_x}^T \hat{\mathbf{w}}_{\mathbf{C}_x}) + \vartheta_{\mathbf{g}_x} Tr(\tilde{\mathbf{w}}_{g_x}^T \hat{\mathbf{w}}_{\mathbf{g}_x})\end{aligned} . \tag{38}$$

It can further be derived to guarantee the stability of the closed-loop system. Herein, we omitted the detailed proof due to the limitation of the writing space.

IV. Simulation Results

Herein, the proposed control approach is applied on the impedance force tracking control of a robot manipulator system with two rigid links and two rigid revolute joints shown in Fig. 2. For i=1, 2, m_i, l_i, l_{ci}, and I_i are mass, length, gravity center length and inertia of link i, respectively. The actual values of quantities refer [12] as m_1=2kg, m_2=1kg, l_1=l_2=0.75m, l_{c1}=l_{c2}=0.375m, I_1=0.09375kg-m^2, I_2=0.046975 kg-m^2. The inertia matrix $\mathbf{D(q)}$ is defined as

$$\mathbf{D(q)} = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}$$

where $d_{11} = m_1 l_{c1}^2 + m_2 l_1^2 + I_1 + m_2 l_{c2}^2 + I_2 + 2m_2 l_1 l_{c2} \cos q_2$, $d_{12} = d_{21} = m_2 l_{c2}^2 + I_2 + m_2 l_1 l_{c2} \cos q_2$, and $d_{22} = m_2 l_{c2}^2 + I_2$. $\mathbf{C(q, \dot{q})}$ and $\mathbf{g(q)}$ are defined as

$$\mathbf{C(q, \dot{q})} = \begin{bmatrix} -m_2 l_1 l_{c2} \dot{q}_2 \sin q_2 & -m_2 l_1 l_{c2} (\dot{q}_1 + \dot{q}_2) \sin q_2 \\ m_2 l_1 l_{c2} \dot{q}_1 \sin q_2 & 0 \end{bmatrix},$$

$$\mathbf{g(q)} = \begin{bmatrix} \left(m_1 l_{c2} + m_2 l_1\right) g \cos q_1 + m_2 l_{c2} g \cos(q_1 + q_2) \\ m_2 l_{c2} g \cos(q_1 + q_2) \end{bmatrix},$$

The Jacobian matrix is defined as

$$\mathbf{J(q)} = \begin{bmatrix} -m_2 l_1 l_{c2} \dot{q}_2 \sin q_2 & -m_2 l_1 l_{c2} (\dot{q}_1 + \dot{q}_2) \sin q_2 \\ m_2 l_1 l_{c2} \dot{q}_1 \sin q_2 & 0 \end{bmatrix},$$

The initial condition for the end-point and reference state are $\mathbf{x}(0) = [0.75\ 0.75\ 0\ 0]^T$ and $\mathbf{x}_m(0) = [0.8\ 0.8\ 0\ 0]^T$. The desired trajectory $\mathbf{x}_d$ is a 0.2 m-radius circle centered at (0.8m, 1.0m) in 10 seconds. As above discussion, system parameter matrices $\mathbf{D}$, $\mathbf{C}$, and $\mathbf{g}$ are assumed to be not exactly known and are approximated by the FNNs. Initial weighting vectors of FNN are chosen as to be zero to avoid initial large control effort and the fuzzy rule number are chosen as seven for $\hat{\mathbf{D}}$, $\hat{\mathbf{C}}$, and $\hat{\mathbf{g}}$. The update laws gain matrices in (24) are chosen with

$$\mathbf{Q}_\mathbf{D}^{-1} = \mathbf{Q}_\mathbf{C}^{-1} = diag(q_1, ..., q_7), \quad q_i = 1, \ i = 1, ..., 7 .$$

$$\mathbf{Q}_{\mathbf{g}}^{-1} = diag(q_1,...,q_7), \quad q_i = 100, \quad i = 1,...,7 .$$

In this simulation, we assume the approximation error can be neglected such that $\boldsymbol{\varepsilon}_1 \approx 0$, and hence the ϑ-modification parameter are chosen $\vartheta_{(\cdot)} = 0$. The constraint surface is a flat wall with a triangle crack and the environment can be modeled as linear spring $f_{ext} = k_w(x - x_w)$. f_{ext} is the force acting on the surface, k_w =5000 N/m is the environment stiffness, x is the coordinate of the end-point in the x-direction, and $x_w = 0.95m$ is the position of the surface. Therefore, the external force vector in equation (2) becomes $\mathbf{F}_{ext} = [f_{ext} \;\; 0]^T$. The controller gains are selected as

$$K_d = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, \quad \boldsymbol{\Lambda} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}.$$

Matrices in the target impedance are designed to be [9]

$$\mathbf{M}_m = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} \text{(kg)}, \quad \mathbf{B}_m = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} \text{(kg.sec/m)},$$
$$\mathbf{K}_m = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \text{(kg/m)}.$$

Several illustrated examples are introduced to demonstrate the performance and effectiveness of the proposed adaptive control approach.

Simulation results and comparisons are shown Figures 3-6. Figure 3 shows the trajectory tracking performance in the Cartesian space (solid line: robot position; dashed-line: environment position; dash-dotted: desired position). We can find that the end point of robot contact the constraint surface with a triangle crack and track the reference trajectory in free space. For the force control, herein, we test the different learning rates η = 10000, 30000 and 60000 to have result in Fig. 4. Obviously, the larger value of η performs quickly. As our experience, the learning rate is chosen as η = 15000. Figure 5 shows the estimate stiffness $\hat{\mathbf{K}}_e$ can be stabilized by our proposed adaptive algorithm and the estimated $\hat{x}_m^*$ is changed value at 1.35s, 2.1s, 2.51s, 2.9s, since the end point touches the wall and triangle crack, respectively. From the observation of Fig. 5, we can find that the estimated stiffness parameter is convergent and the estimated environment position is contained to achieve the adaptive force control. Figure 6 shows the control effort at t = 1~10s.

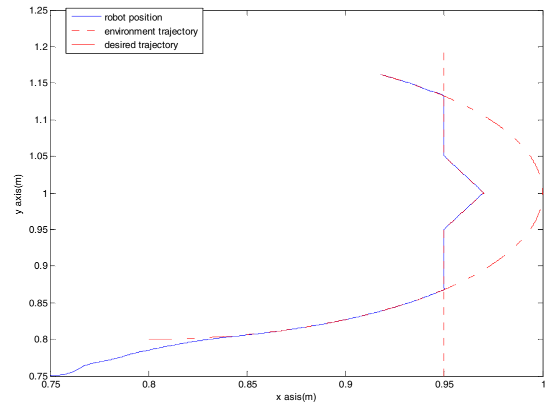

Figure 3: Position tracking performance result at t=0~4s.

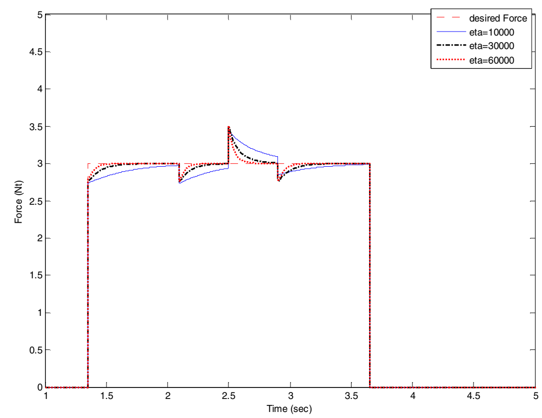

Figure 4: Force tracking performance with different learning rates.

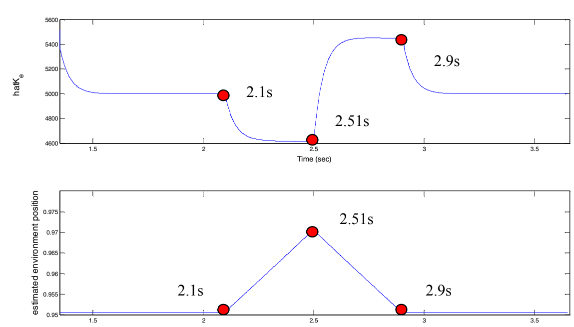

Figure 5: Estimated Stiffness and estimated environment position.

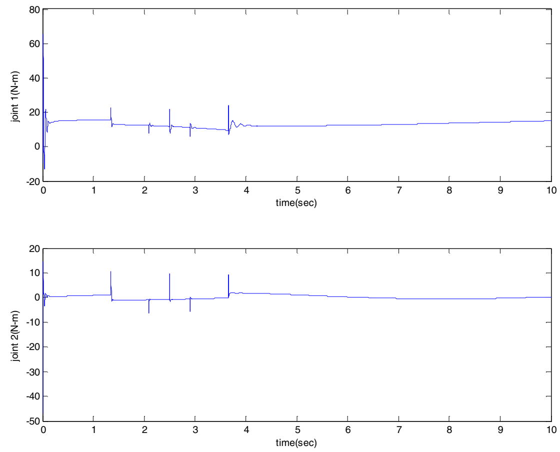

Figure 6: control effort.

We can observe that the proposed adaptive impedance force control scheme performs well for unknown environment case and variant desired force case.

V. CONCLUSION

In this paper, we have proposed the adaptive control scheme to treat the impedance force tracking control for an *n*-link robot manipulator under unknown environment. The unknown system parameter matrices are estimated by the fuzzy neural networks and the adaptive force control laws are established by gradient method. Based on the Lyapunov stability theory, the update laws of fuzzy neural networks can be derived and the stability of the closed-loop system is guaranteed. Finally, simulation results of a 2-DOF robot manipulator with environment constraint are introduced to illustrate the performance and effectiveness of our approach.

ACKNOWLEDGMENT

This work was supported in part by the National Science Council, Taiwan, R.O.C., under contracts NSC-100-2221-E-005-093-MY2.

REFERENCES

[1] N. Hogan, "Impedance control: an approach to manipulation: Part 1- Theory, Part 2- Implementation, Part 3- An approach to manipulation," ASME *Journal of Dynamic System, Measurement, and Control*, vol. 107, pp. 1-24, 1985.

[2] M. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," ASME *Journal of Dynamic System, Measurement, and Control*, vol. 102, pp. 126-133, 1981.

[3] A. Abdallah, D. Dawson, P. Dorato, and M. Jamishidi, "Survey of robust control for rigid robots," *IEEE Control System Magazine,* vol. 11, no. 2, pp. 24-30, 1991.

[4] D. Cai and Y. Dai, "A globally convergent robust controller for robot manipulator," *Proc. of IEEE Int. Conf. on Control Applications,* pp. 328-332, 2001.

[5] E. S. Shin and K. W. Spong, "Robust output feedback control of robot manipulator using high-gain observer," *Proc. of IEEE International Conf. on Control Applications*, pp.881-886, 1999.

[6] R. Ortega and M. W. Spong, "Adaptive motion feedback control of rigid robot: a tutorial," *Proc. 27th Conf. on Decision and Control*, pp. 1575-1584, 1988.

[7] D. Sun and J. K. Mills, "Performance improvement of industrial robot trajectory tracking using adaptive-learning scheme," ASME *Journal of Dynamic System, Measurement, and Control*, vol. 121**,** no. 2, 285-292, 1999.

[8] N. Sedegh and R. Horowitz, "Stability and robustness analysis of a class of adaptive controller for robotic manipulators," *Int. J. of Robotic Research*, vol. 9, no. 3, 74-92, 1990.

[9] M. C. Chien and A. C. Huang, "Adaptive impedance control of robot manipulator based on function approximation technique," *Robotica*, vol. 22*,* pp. 395-403, 2004.

[10] A. C. Huang and Y. S. Kuo, "Sliding control of nonlinear systems containing time varying uncertainties with unknown bound," *Int. J. of Control,* vol. 74, no 3, pp. 252-264, 2001.

[11] A. C. Huang and Y. C Chen, "Adaptive sliding control for single link flexible-joint robot with mismatched uncertainties," *IEEE Trans. on Control System Technology,* vol. 12, no. 5, pp. 770-775, 2004.

[12] S. C. Wu, *Adaptive Controller Design for Robot Manipulators Based on Function Approximation Technique*, Master Thesis, National Taiwan University of Science and Technology, Taiwan, July, 2002.

[13] S. Jung, T. C. Hsia and R. G. Bonitz, "Force tracking impedance control of robot manipulators under unknown environment," *IEEE Trans. on Control System Technology*, vol.12, no.2, pp. 474-483, 2004.

[14] S. Jung and T. C. Hsia, "Neural network impedance force control of robot manipulator," *IEEE Trans. on Industrial Electronics*, vol. 45, no. 3, pp. 451-461, 1998.

[15] S. Jung, S. B. Yim, and T. C. Hsia, "Experimental studies of neural impedance force control of robot manipulators," *Proc. IEEE Conf. on Robotics and Automations,* pp. 3453-3458, 2001, Seoul, Korea.

[16] S. Jung and T. C. Hsia, "Reference compensation technique of neural force tracking impedance control for robot manipulators," *World Congress on Intelligent Control and Automation*, Jinan, China, July 6-9, 2010.

[17] Y. Tanaka and T. Tsuji, "On-line Learning of Robot Arm Impedance Using Neural Network," *Proc. of IEEE Int. Conf. on Robotic and Biomimetics*, Shenyang, China, August 22-26, 2004.

[18] C. H. Lee and C. C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural network," *IEEE Trans. on Fuzzy Systems,* vol. 8, no. 4, pp. 349-366, 2000.

[19] C. H. Lee and B. R. Chung, "FNN-based disturbance observer controller design for nonlinear uncertain systems," *WSEAS Trans. on Systems and Control*, vol. 2, no. 3, pp. 255-260, 2007.

[20] C. H. Lee and B. R. Chung, "Adaptive backstepping controller design for nonlinear uncertain systems using fuzzy neural systems," *Int. J. of Systems Science*, vol. 43, no. 10, pp. 1855-1869, 2012.

[21] T. J. Ren, "A robust impedance control using recurrent fuzzy neural networks," *The 3nd Int. Conf. Innovative Computing Information and Control (ICICIC'08)*, 2008.

[22] H. Asada and J. J. E. Slotine, *Robot-Analysis and Control*, J. Wiley, N. Y., 1986.

[23] J. J. E. Slotine and W. Li, *Applied Nonlinear Control,* Englewood Cliffs, NJ: Prentice-Hall, 1991.

[24] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot Manipulator Control Theory and Practice,* Second Edition, Revised and Expanded, Marcel Dekker, 2004.

[25] A. C. Huang and M. C. Chien, *Adaptive Control of Robot Manipulators,* World Scientific Publishing Co. Pte. Ltd, 2010.