

## A Fuzzy Self-Tuning Hybrid Force/Position Controller for Robotic Manipulator

حاكم باستخدام المنطق الغير محدد الموفق ذاتيا للذراع إنسان آلي

Prof. Dr. F. M. Abdel-Kader      Dr. K. A. El-Serafi      Dr. A. M. El-Saadawi  
Eng. A. S. Abdel-Rhman  
Suez Canal University, Faculty of Engineering port-Said

ملخص البحث

يوجد نوعان من الحركة التي يتم تطبيقها على ذراع الروبوت. النوع الأول الحركة الحرة ويتحرك فيها الذراع في الفراغ ويتم فيها التحكم في موضع نهاية الذراع (اليد). والنوع الآخر الحركة المقيدة ويتحرك الذراع ملامسا لمسطح البيئة المحيطة به ويتم فيها تقسيم درجات الحرية إلى مجموعتين الأولى تتحكم في الموضع والثانية تتحكم في القوة المطلوب تطبيقها عند كل نقطة. وقد تم اقتراح نظام تحكم باستخدام المنطق الغير محدد (الفازي) الموفق ذاتيا. وقد تم تطبيق هذا النظام على مشغل ذراع إنسان آلي (روبوت) ذو درجتين حرية في مستوى. وقد أظهرت النتائج المستخلصة كفاءة النظام المقترح وقدرته على تتبع مجموعة مختلفة من المسارات.

### Abstract

The major problems of hybrid force/position control arise from uncertainty of the robot manipulator and unknown parameters of the task environment. This paper proposes a self-tuning fuzzy hybrid force/ position control scheme, which can force the end-effector to tracking a desired force and position trajectories in the Cartesian space. The output of the self-tuning algorithm adjusted on-line by the Scaling Factor (SF). The selection of SF value depends on the values of error ( $e$ ) and change of error ( $\dot{e}$ ). The present work is applied to the control of a two degrees-of-freedom (DOF) planar robot manipulator. The simulation results were carried out by using matlab5.3 program.

### 1- Introduction

In many industrial and manufacturing applications, a robot manipulator is required to make contact with the environment, typical examples of these tasks are assembly of mechanical parts, deburring, painting, grinding, following contour, and assembly related tasks [1][2]. In these applications, contact forces between the robot manipulator end-effector and the environment are generated; therefore, simultaneous control of the end-effector position and the interaction force is required to successfully execute these tasks [3]. During the past years, the force control schemes are often referred to as compliant control or hybrid force/position control in literature [4][5][6][7].

Generally, the compliant control is trying to simultaneously solve the force control problem and the positions control problem in the task frame, and provide reconciliation between the respective solutions of these problems. Contrarily, the hybrid force/position control is to carefully formulate the task frame so that the associated task space can be easily decomposed into force control subspace and position control subspace, and then achieves the desired respective objectives. The control strategy is primarily based on the fact that the total degrees of freedom (denoted by  $n$ ) for the force/position problem can be partitioned into  $m$  position objectives and  $k$  force control objectives (i.e.,  $n = m + k$ ) [4]. This decoupling is accomplished by employing the reduced order transformation of [3][4] to develop  $n$  dynamic equations, which are used to develop the position and force controller separately.

When the robot manipulators are in contact with their surrounding environment the uncertainties of the parameters arise and the control system must be able to deal with these

variations in the parameters. In the presence of such uncertainties, linear feedback controllers (for example, PD or PID) cannot provide consistent performance. Therefore, in order to control the robot manipulator, it is necessary a control algorithm having simple computation and robustness to uncertainties. Fuzzy logic controllers (FLC's) have been reported to be successfully used for a number of complex and nonlinear process [8][9][10][11]. The self-tuning algorithm can tune their output automatically depending on the scale factor (SF) term, which can be selected on-line depend on the values of the error  $e$  and the change of error  $\dot{e}$  [12]. The overall control scheme drives the tracking errors to a neighborhood of zero.

2- The Dynamic Model of the Robot Manipulator

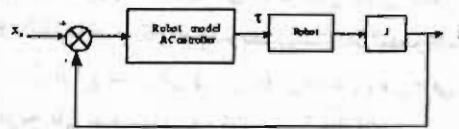


Fig. 1. A block diagram shows the robot manipulator control scheme.

Consider an  $n$  degree-of-freedom robot manipulator its dynamic model in joint space defined as

$$\tau = M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_e, \tag{1}$$

Where

$q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ : are the position, velocity, and acceleration in the joint space.

$M(q) \in \mathbb{R}^{n \times n}$ : is the inertia matrix.

$V_m(q, \dot{q})$ : is the vector representing the centrifugal and coriolis forces satisfying  $\dot{M} - 2V_m$  is a skew-symmetric matrix,

$G(q)$ : is the vector of gravitational forces,

$F(\dot{q})$ : is the vector of friction forces, and

$\tau_e$ : is an  $n \times 1$  vector in joint space coordinates which denotes the force exerted on the environment.

The robot manipulator equation is usually given by

$$\tau = M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + J^T(q)f \tag{2}$$

Where  $f \in \mathbb{R}^6$  is the output forces and torque in task space.  $f$  can be defined as

$$f = K_e(x - x_e) \tag{3}$$

Where

$K_e$ : is an  $n \times n$  diagonal, positive semi-definite, constant matrix used to denote the environmental stiffness, and

$x_e \in \mathbb{R}^n$ : is the static location of the environment at rest, and

$x \in \mathbb{R}^n$ : is the end-effector position and orientation in the task space.

If  $x$  defined by

$$x = h(q), \tag{4}$$

Where  $h(q)$  is the task space coordinates in terms of the joint space coordinates that can be found from the manipulator kinematics and the appropriate relationships between the joint and the task spaces. And  $n$  is the number of links of the robot manipulator.

The derivative of  $x$  with respect to time is defined as.

$$\dot{x} = J(q)\dot{q}, \quad (5)$$

Where  $J(q)$  is an  $6 \times n$  task space Jacobian matrix relating joint space velocity and task space velocity [13].

$$J(q) = \frac{\partial h(q)}{\partial q}, \quad (6)$$

By differentiate (5) with respect to time we obtain that

$$\ddot{x} = J(q)\ddot{q} + \dot{J}(q)\dot{q}, \quad (7)$$

Rewrite (7) to be on the form

$$\ddot{q} = J(q)^{\#}(\ddot{x} - \dot{J}(q)\dot{q}). \quad (8)$$

Where  $J(q)^{\#}$  is the pseudo inverse of the Jacobian matrix. When  $n=6$ , it is a square matrix and  $J(q)^{\#} = J(q)^{-1}$ .

Substituting (8) into (2) yields

$$\tau = M(q)J(q)^{\#}(\ddot{x} - \dot{J}(q)\dot{q}) + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + J^T(q)f. \quad (9)$$

The corresponding feedback linearization control for the dynamics given by (9) is given by

$$\tau = M(q)J(q)^{\#}(a - \dot{J}(q)\dot{q}) + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + J^T(q)f. \quad (10)$$

Where  $a$  is an  $n \times 1$  vector used to represent the linear position and force control strategies. From (9) and (10) we have

$$\ddot{x} = a. \quad (11)$$

Let  $x_T$  represent the  $m \times 1$  vector of output positions and orientation in the tangent direction ("T") and  $x_N$  represent the  $(n-m) \times 1$  vector of output positions and orientation in the normal direction ("N"), i.e.,  $x = [x_N, x_T]$ .

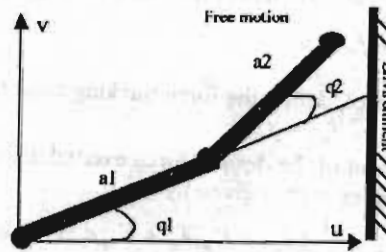


Fig. 2. Two link planar manipulator

Let  $a_T$  and  $a_N$  represent the acceleration input vector correspondingly, i.e.,  $a = [a_N, a_T]$ ; then

$$\ddot{x}_T = a_T, \quad \ddot{x}_N = a_N \quad (12)$$

Therefore, the motion of the manipulator can be controlled independently for each direction. Obviously, the control in the tangent directions will not have an effect on dynamics in the normal directions and vice versa. Fig. 2 shows the geometric of the robot and its environment. For the purpose of feedback control, define the tracking error to be

$$e = x_d - x, \quad \dot{e} = \dot{x}_d - \dot{x} \quad (13)$$

Where  $x_d$  represent the desired input trajectory in the task space. So, the corresponding linear controller is then given as

$$a = \ddot{x}_d + K_v \dot{e} + K_p e, \quad t \geq 0 \quad (14)$$

Where  $K_v$  and  $K_p$  are positive control gains.

In the free motion period only the position control is used in the tangent and normal directions and the controller in the tangent direction will be

$$a_T = \ddot{x}_{Td} + K_{Tv}\dot{e}_T + K_{Tp}e_T, \quad t < t_{sw} \tag{15}$$

Where  $K_{Tv}$  and  $K_{Tp}$  are positive control gains and  $t_{sw}$  represents the time of the detection of the impact and  $e_T = x_{Td} - x_T$

In the normal direction, the position control is used before the detection of the impact: that is,

$$a_N = \ddot{x}_{Nd} + K_{Nv}\dot{e}_N + K_{Np}e_N, \quad t < t_{sw} \tag{16}$$

Where  $e_N = x_{Nd} - x_N$ .

On the detection of the impact, the controller will switch from position control to force control in the normal directions. Therefore,  $t < t_{sw}$  is the time period of free motion mode and  $t \geq t_{sw}$  is period that the desired force trajectory needs to be tracked in the normal directions.

Assume that the environment can be modeled as a spring. Specifically, the normal force  $f_N$  exerted on the environment is given by

$$f_N = k_e(x_N - x_e) \tag{17}$$

where  $k_e$  is the environmental stiffness, and  $x_e$  is used to represent the static location of the environment in the of the normal space  $x_N$ . To formulate the force dynamics taking the second derivative of (17) with respect to time gives the expression

$$\ddot{x}_N = \frac{1}{k_e}\ddot{f}_N, \tag{18}$$

where the normal task space acceleration is written in terms of the second derivative of the normal force. Substituting (14) into (12) yields the force dynamics

$$\frac{1}{k_e}\ddot{f}_N = a_N. \tag{19}$$

For the purpose of feedback control, define the force tracking error to be

$$e_N = f_{Nd} - f_N, \tag{20}$$

where  $f_{Nd}$  represents the component of the desired force exerted normal to the environment.

The corresponding linear controller is then given by

$$a_N = \frac{1}{k_e}(\ddot{f}_{Nd} + K_{Nv}\dot{e}_N + K_{Np}e_N) \quad t \geq t_{sw} \tag{21}$$

with  $K_{Nv}$  and  $K_{Np}$  are positive control gains.

### 3- The Proposed Self-Tuning Fuzzy Controller

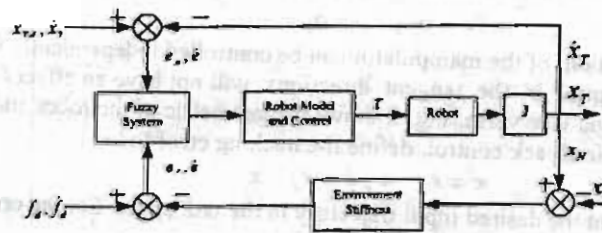


Fig. 3. Robot control with fuzzy system.

Let the controller given in (10) to be

$$r = M(q)J(q)^T(\dot{a}_T - \dot{J}(q)\dot{q}) + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + J^T(q)f. \tag{22}$$

where  $ast$  is an  $n \times 1$  vector used to represent the linear position and force control strategies where

$$ast = \begin{bmatrix} a_{NZ} \\ a_{TZ} \end{bmatrix} \tag{23}$$

where  $a_{NZ}$  is the  $m \times 1$  vector represent the linear space controller in the normal direction. And  $a_{TZ}$  is the  $(n-m) \times 1$  component that represent the linear space controller in the tangent space direction.

Let

$$a_{NZ} = \frac{1}{K_f} (\ddot{q}_{dN} + u_{NZ}) \quad u_{NZ} = G_n \cdot \alpha u_f \tag{24}$$

Where  $u_{NZ}$  is the fuzzy component that represent the PD term in  $a_{NZ}$ . This term will be computed by using the self-tuning fuzzy control system.

And

$$a_{TZ} = (\ddot{q}_{dT} + u_{TZ}) \quad u_{TZ} = G_u \cdot \alpha u_v \tag{25}$$

Where  $u_{TZ}$  is the fuzzy component that represent the PD term in  $a_{TZ}$ . This term also will be selected by using the self-tuning fuzzy control system.

The gain (output SF  $\alpha$ ) of the fuzzy controllers are adjusted on-line according to the current states of the controlled processes, thereby making them self-tuning FLC's. The objective here is to adapt the output SF for given input to achieve better control performance.

The block diagram of the proposed self-tuning FLC is shown in Fig. 4.

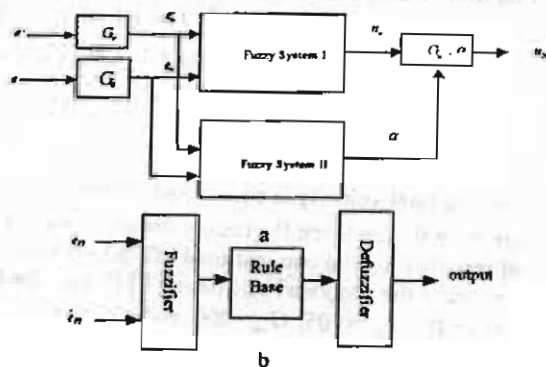


Fig. 4. a) Block diagram of the proposed self-tuning FLC b) Structure of the FLC.

$\dot{e}/e$	NB	NM	NS	Z	PS	PM	PB
PB	Z	NB	NB	NB	NB	NB	NB
PM	PB	Z	NM	NM	NM	NM	NB
PS	PB	PS	Z	NS	NS	NM	NB
Z	PB	PS	PS	Z	NS	NS	NB
NB	PB	PM	PS	PS	Z	NS	NB
NM	PB	PM	PM	PM	PM	Z	NB
NB	PB	PB	PB	PB	PB	PB	Z

Tab. 1. Fuzzy rules for computation of  $u_n$ .

$\dot{e}/e$	NB	NM	NS	Z	PS	PM	PB
PB	Z	S	SB	B	VB	VB	VB
PM	VS	S	MB	B	B	VB	VB
PS	VS	S	VS	VB	B	MB	VB
Z	S	SB	MB	Z	MB	SB	S
NS	VB	MB	B	VB	VS	S	VS
NM	VB	VB	B	B	MB	S	VS
NB	VB	VB	VB	B	SB	S	Z

Tab. 2. Fuzzy rules for computation of  $\alpha$ .

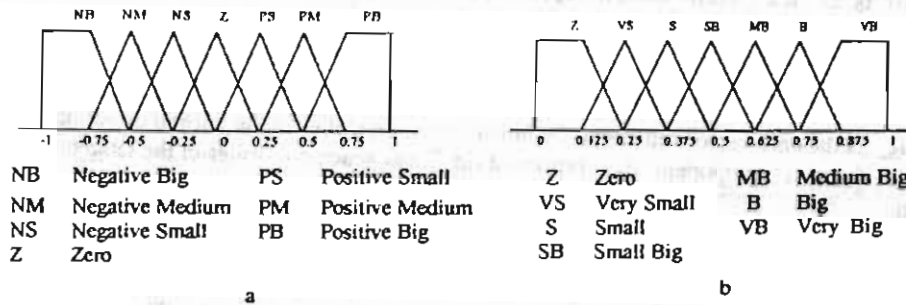


Fig. 5. Membership functions of a)  $e, \dot{e}$ , and  $u$  and b) gain updating factor  $\alpha$ .

4. The simulation result

The control algorithm was implemented on a two-link planar robot manipulator. The model is showing in *appendix A*. The results of the conventional PD hybrid force/position control are carried out with proportional constant of  $k_p = \text{diag} \{100,100\}$  and derivative constant of  $k_v = \text{diag} \{20,20\}$ . The environment is modeling as a spring with environment stiffness  $k_e=1000$  N/m.

In all cases the trajectory of the position is a 5<sup>th</sup> order polynomial represented as follows

$$y_i(t) = a_{0i} + a_{1i}t + a_{2i}t^2 + a_{3i}t^3 + a_{4i}t^4 + a_{5i}t^5, \quad i=1,2 \quad (26)$$

Where  $y_i(t)$  is the position of the robot manipulator in the Cartesian space with

$$a_{0i} = y_i(0) \quad a_{1i} = \dot{y}_i(0) \quad a_{2i} = 0 \quad a_{3i} = (10 * (y_i(f) - y_i(0)) / t_f^3) - (6 * \dot{y}_i(0) / t_f^2)$$

$$a_{4i} = (-15 * (y_i(f) - y_i(0)) / t_f^4) + (8 * \dot{y}_i(0) / t_f^3) \quad a_{5i} = (6 * (y_i(f) - y_i(0)) / t_f^5) - (3 * \dot{y}_i(0) / t_f^4)$$

where

$y_i(0)$  : is the initial position

$y_f(0)$  : is the final position

$\dot{y}_i(0)$  : is the initial velocity, and the final velocity is considered to be zero.

If we use an exponential trajectory for the force the results shows in Fig (6) and Fig (7). Fig (6) and shows the results represented by the conventional PD hybrid force/position control and Fig. (7) shows the responses of the proposed self-tuning FLC with the FLC parameters are given as  $G_{fp}=100$ ,  $G_{sp}=5$ ,  $G_{sf}=10$ ,  $G_{sf}=0.05$ ,  $G_{up}=200$ , and  $G_{uf}=500$ .

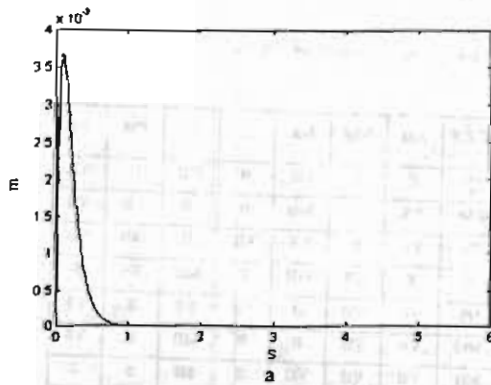


Fig. 6 The conventional PD controller response  
a) the tracking position error (continued)

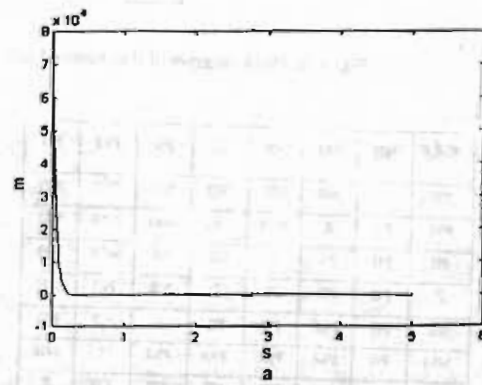


Fig. 7 The self-tuning fuzzy controller response  
a) the tracking position error (continued)

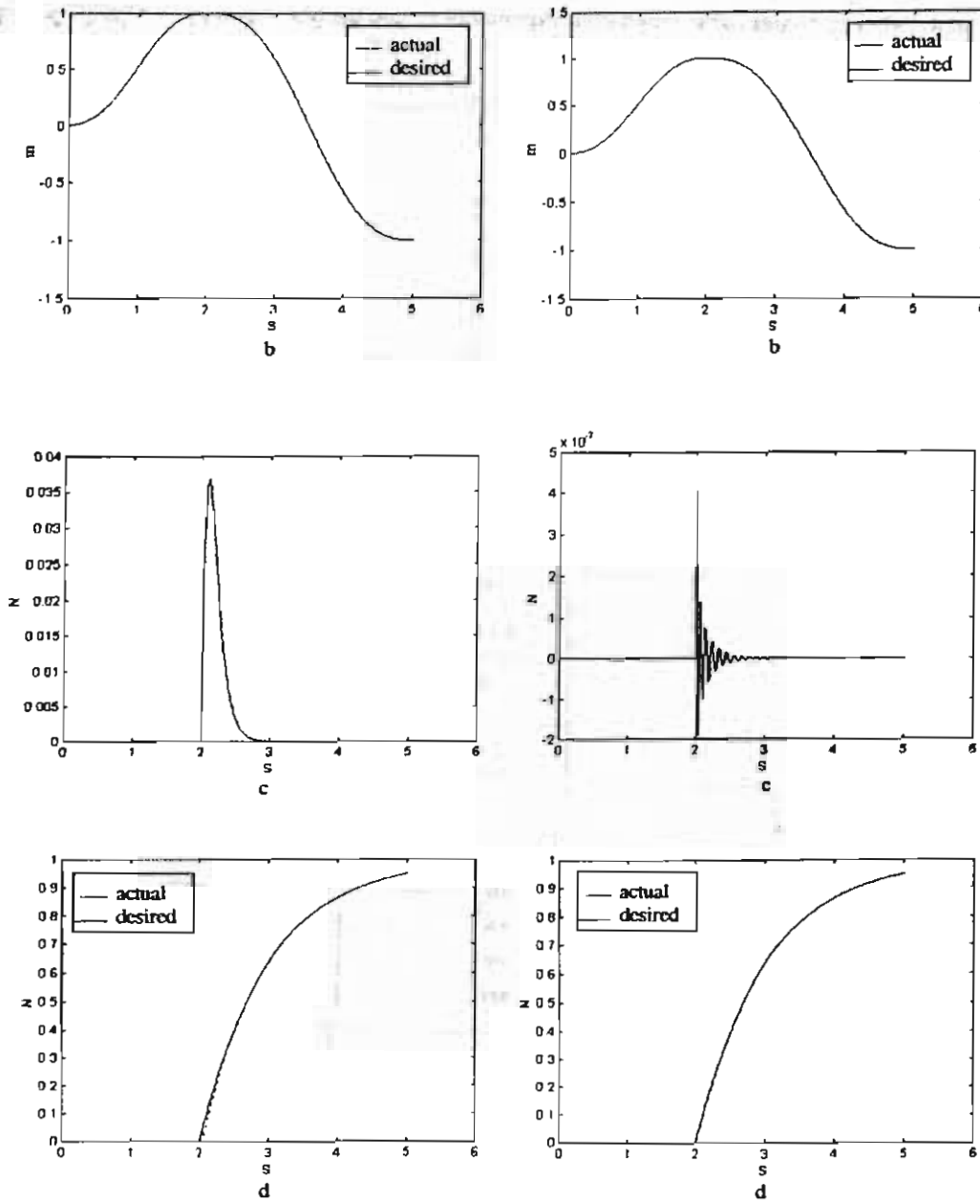


Fig. 6 The conventional PD controller response b) the desired and actual tangent position trajectory c) The tracking force error d) the desired and actual force trajectory

Fig. 7 The self-tuning fuzzy controller response b) the desired and actual tangent position trajectory c) The tracking force error d) the desired and actual force trajectory

If we use step trajectory for the force the results shows in Fig (8) and Fig (9). Fig (8) and shows the results represented by the conventional PD hybrid force/position control and Fig. (9) shows the responses of the proposed self-tuning FLC with the FLC parameters are given as  $G_{ep}=100$ ,  $G_{ep}=5$ ,  $G_{ef}=10$ ,  $G_{ef}=0.05$ ,  $G_{up}=200$ , and  $G_{uf}=500$ .

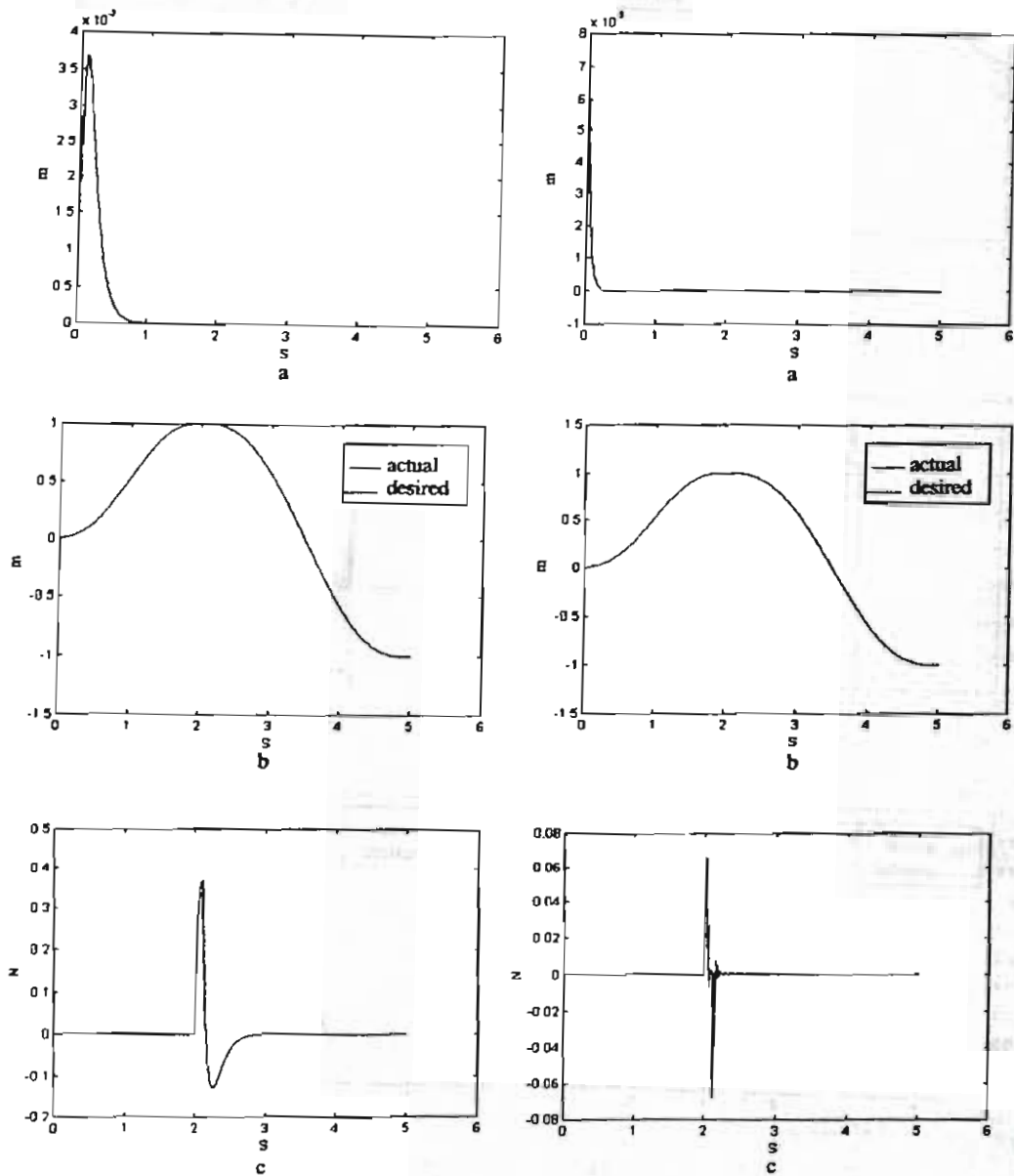


Fig. 8 The conventional PD controller response a) the tracking position error b) the desired and actual tangent position trajectory c) The tracking force error (continued)

Fig. 9 The self-tuning fuzzy controller response a) the tracking position error b) the desired and actual tangent position trajectory c) The tracking force error (continued)



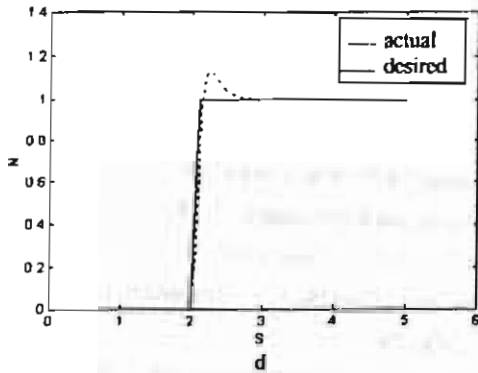


Fig. 8 The conventional PD controller response d) the desired and actual force trajectory

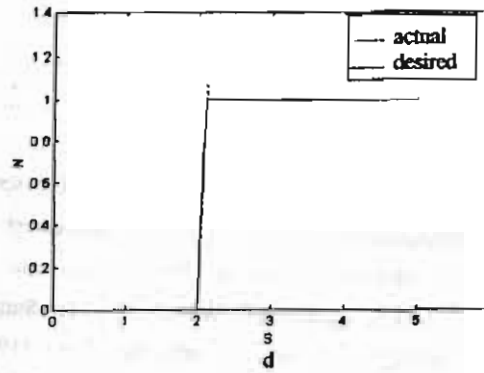


Fig. 9 The self-tuning fuzzy controller response d) the desired and actual force trajectory

**5. Conclusion**

In this paper we have proposed a self-tuning fuzzy logic controller for robot manipulator under constrained motion. The simulation is carried out on a simple two-link planner robot manipulator. It is shown that the gain output SF ( $\alpha$ ) of the proposed fuzzy controller is adjusted on-line according to the current states of the controlled process, thereby making them self-tuning fuzzy logic controller. When the simulation is carried out using the conventional PD controller, the position and force trajectories is tracking with error and the values of the error is so high. But when the proposed self-tuning fuzzy logic controller is applied, the system gives good performance and the values of the output errors are reduced.

**Appendix A**

**The robot modeling**

$m_1=10$        $m_2=5$        $L_1=1$        $L_2=1$   
 $l_{c1}=0.5$      $l_{c2}=0.5$      $I_1=10/12$      $I_2=5/12$

**The inertia matrix**

$M_{11} = m_1 l_{c1}^2 + 2m_1 l_1 l_{c2} C_2 + m_2 (l_1^2 + l_{c2}^2) + I_1 + I_2$

$M_{12} = m_2 l_1 l_{c2} C_2 + m_2 l_{c2}^2 + I_2$

$M_{21} = M_{12}$        $M_{22} = m_2 l_{c2}^2 + I_2$

**The coriolis/centripetal matrix**

$V_{m11} = -m_2 l_1 l_{c2} S_2 \dot{q}_2$

$V_{m12} = -m_2 l_1 l_{c2} S_2 (\dot{q}_1 + \dot{q}_2)$

$V_{m21} = m_2 l_1 l_{c2} S_2 \dot{q}_1$        $V_{m22} = 0$

**The gravity matrix**

$G_{11} = g(m_1 l_{c1} + m_2 l_1) C_1 + g m_2 l_{c2} C_{12}$

$G_{12} = g m_2 l_{c2} C_{12}$

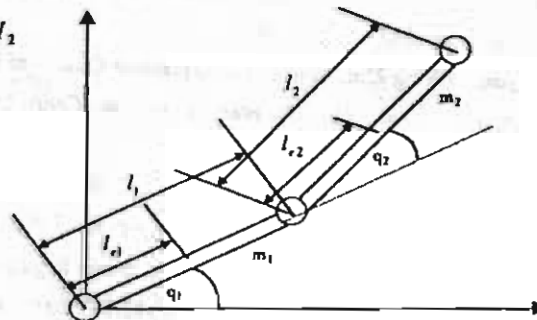


Fig. A-1. Two-link planar robot manipulator.

## References

- 1) S. Chiaverini and L. Sciavicco " The Parallel Approach to Force/Position Control of Robot Manipulator, " *IEEE Trans. on Robotics and Automation* vol. 9, no. 4, pp. 361-373, Aug. 1993.
- 2) M. S. de Queiroz, J. Hu, and D. M. Dawson, " Adaptive Position/Force Control of Robot Manipulators Without Velocity Measurements: Theory and Experimental, " *IEEE Trans. on Sys. Man. and Cybernetics- Part B: Cybernetics*, Vol. 27, No. 5, October 1997.
- 3) N. McClamroch and D. Wang, " Feedback Stabilization and Tracking of Constrained Robots, " *IEEE Trans. Automat. Contr.*, vol. 33, pp. 419-426, May 1988.
- 4) M. H. Raibert and J. J. Craig, "Hybrid Position/Force Control of Manipulators, " *ASME Journal of Dynamic Systems, Measurements, and Control*. Vol. 102 pp. 126-133, 1981.
- 5) Daniel E. Whitney, "Historical Perspective and State of the Art in Robot Force Control, " *the International Journal of Robotics Research*, vol. 6, no. 1, pp. 3-14, 1987.
- 6) Hong Zhang and Richard P. Paul, "Hybrid Control of Robot Manipulators, " *IEEE trans. on Robotics and Automation*. Vol. 5, no. 1, pp. 602-607. May 1985.
- 7) Hiroshi Ishikawa, Chihiro Sawada, Kei Kawase, and Masayuki Takata, "Stable Compliance Control and Its Implementation for a 6 D.O.F. Manipulator, " *IEEE trans. on Robotics and Automation*. Vol. 2, no. 2, pp. 98-103. April 1989.
- 8) Feng-Yih Hsu and Li-Chen Fu, "A New Design of Adaptive Fuzzy Hybrid Force/Position Controller for Robot Manipulators, " *IEEE Inter. Conf. On Robotics and Automation*, pp. 863-868, 1995.
- 9) Feng-Yih Hsu and Li-Chen Fu, "Intelligent Robot Deburring Using Adaptive Fuzzy Hybrid Force/Position Control, " *IEEE trans. on Robotics and Automation*, Vol. 16, No. 4, pp. 325-335. August 2000.
- 10) Feng-Yih Hsu and Li-Chen Fu, "A New Adaptive Fuzzy Hybrid Force/Position Control for Intelligent Robot Deburring, " *IEEE Inter. Conf. On Robotics and Automation, Detroit, Michigan*, pp. 2476-2481, 1999.
- 11) B. Qiao, J. Ying Zhu, " Fuzzy Force/Position Control of Robotic Manipulator Based on Force Feedback, " *Proc. Of 5<sup>th</sup> inter. Conf. on Control, Automation, Robotic, and Vision, ICARCV'98*, Singapore, 1998.
- 12) Rajani K. Mudi and Nikhil R. Pal, "A robust Self-Tuning Scheme for PI- and PD-Type Fuzzy Controllers, " *IEEE Trans. on Fuzzy Systems*, vol. 7, No. 1, February 1999.
- 13) T.-J. Tran, Y. Wu, N. Xi, and A. Isidori, "Force Regulation and Contact Transition Control, " *IEEE Control Systems Magazine*, pp. 32-40, February 1996.