A neuro fuzzy controller for planar robot manipulators

FRANCESCO M. RAIMONDI, MAURIZIO MELLUSO, VINCENZO BONAFEDE
Dipartimento di Ingegneria dell'Automazione e dei Sistemi
University of Palermo
Viale delle Scienze – Post code 90128
PALERMO (ITALY)

Abstract: - A neuro fuzzy control algorithm for planar robotic manipulators is developed in this paper. This control algorithm is able to handle uncertainties, disturbances or unmodeled dynamics. Asymptotic stability of the equilibrium state of the control system is shown by use of direct Lyapunov method and Barbalat's Lemma. A series of simulations confirms the algorithm goodness.

Key-Words: - fuzzy controller, neuro fuzzy, Lyapunov stability, tracking control, robot manipulators

1 Introduction

A main concern of robotics applications is to find an effective controller to achieve accurate tracking of desired motions. Several existing control methods can be used in the case that the manipulator parameters are known in advance. However, in the presence of parameter uncertainties, the many of these control techniques fail to track the desired motion satisfactorily.

Hybrid control techniques that combine fuzzy logic and conventional control to design fuzzy logic controller can offer better control performance of complex systems. The rationate behind this control approach is to mix, in a balanced form, model-based control techniques with fuzzy logic schemes. Applications of this philosophy to control the motion in joint space of robot manipulators has grown in recent years. The classical fuzzy controllers in literature [1] [2] are based on fuzzy tuning algorithms to select the Proportional and Derivative (PD) gains of model-based controllers according to the actual position error. In contrast with the use of fuzzy logic to help conventional control schemes by means of fuzzy tuning or fuzzy feedforward signals, one important application of fuzzy logic is as direct fuzzy controllers where the control actions are directly computed by the fuzzy controller.

In this paper a motion tracking controller for robot manipulators based on a combination of a model based technique and a fuzzy scheme is proposed. This paper is organized as follows.

In Section 2 there is a description of the robot dynamics and fuzzy motion tracking control. The complete control structure is composed by a fuzzy controller plus a full non linear robot dynamic compensation – linearizing feedback [3] – in such a way that this structure leads to a very simple closed-loop system, which is represented by a non autonomous nonlinear differential equation. In this paper the rigorous proof – via Lyapunov theory and Barbalat's Lemma – that the closed-loop system is asymptotically stable is proposed. A neural approach to the problem is also proposed by application of Adaptive-Network based Fuzzy Inference System (ANFIS) algorithm [4]. This method allows to obtain adaptive neural networks which are functionally equivalent to Fuzzy Inference Systems (FIS).

In Section 3 the practical feasibility of the proposed controller is shown by means of experiments on a two degrees of freedom planar robotic manipulator. All the simulations have been developed using MATLAB 6.1 and MATLAB FUZZY TOOLBOX.

2 Robot Dynamics and Fuzzy Motion Tracking Control

The dynamics of a planar n-link robot can be written as [3]

$$\boldsymbol{\tau} = \mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{f}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) \quad (1)$$

where $\mathbf{q}$ is the nx1 generalized vector of joint positions, $\dot{\mathbf{q}}$ is the nx1 vector of joint velocities, $\boldsymbol{\tau}$ is the nx1 vector of applied torque inputs, $\mathbf{D(q)}$ is the nxn symmetric positive defined inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}$ is the nx1 vector of centripetal and Coriolis torques, $\mathbf{f}(\dot{\mathbf{q}})$ is the nx1 vector of the viscous friction term, and $\mathbf{g(q)}$ is the nx1 vector of gravitational torques.

The motion control problem of manipulators in joint space can be stated in the following terms. Assume that joint position $\mathbf{q}$ and joint velocity $\dot{\mathbf{q}}$ are available for measurement. Let the nx1 desired joint position vector $\mathbf{q_d}$ be a twice differentiable vector function. A motion controller is defined as a controller to determine the actuator torques $\boldsymbol{\tau}$ in such a way that the following control aim be achieved:

$$\lim_{t\to\infty}\mathbf{q}(t)=\mathbf{q_d}(t)$$

One conventional solution to this problem is provided by the linearizing feedback given by [3]

$$\boldsymbol{\tau}=\mathbf{D}(\mathbf{q})\left[\ddot{\mathbf{q}}_{\mathbf{d}}+\mathbf{K_d}\dot{\mathbf{e}}+\mathbf{K_p}\mathbf{e}\right]+\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\mathbf{f}(\dot{\mathbf{q}})+\mathbf{g}(\mathbf{q}) \quad (2)$$

(2)

where the joint position and velocity error vectors are denoted respectively by the nx1 vectors $\mathbf{e}=\mathbf{q_d}-\mathbf{q}$ and $\dot{\mathbf{e}}=\dot{\mathbf{q}}_{\mathbf{d}}-\dot{\mathbf{q}}$, where $\dot{\mathbf{q}}_{\mathbf{d}}$ being the nx1 vector of desired velocity, and $\mathbf{K_d}$ and $\mathbf{K_p}$ are symmetric positive definited matrices. Supposing $\mathbf{f}(\dot{\mathbf{q}})$ is unknown, the proposed Fuzzy Logic Controller (FLC) is given by the control law:

$$\boldsymbol{\tau}=\mathbf{D}(\mathbf{q})\left[\ddot{\mathbf{q}}_d+\boldsymbol{\Phi}(\mathbf{e},\dot{\mathbf{e}})\right]+\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\mathbf{g}(\mathbf{q}) \quad (3)$$

where $\boldsymbol{\Phi}(\mathbf{e},\dot{\mathbf{e}})$ is a nx1 vector whose entries $\phi_i(e_i,\dot{e}_i)$ (i=1,2,…,n) are the real input-output mapping of the FLC, that is:

$$\Phi(\mathbf{e},\dot{\mathbf{e}})=\begin{bmatrix}\phi_1(e_1,\dot{e}_1)\\ \phi_2(e_2,\dot{e}_2)\\ \vdots\\ \phi_n(e_n,\dot{e}_n)\end{bmatrix} \quad (4)$$

The control law (3) keeps the non linear compensation terms of (2), but the undesiderable constant terms $\mathbf{K_d}$ and $\mathbf{K_p}$ are obviated. The elements of (4) fulfill some key properties [5]:

a) $\boldsymbol{\Phi}(\mathbf{e},\dot{\mathbf{e}})$ is globally continous and bounded
b) $\phi_i(0,0)=0$
c) $\phi_i(e_i,\dot{e}_i)=-\phi_i(-e_i,-\dot{e}_i)$
d) for every $(e_i,\dot{e}_i)$:
$$0\le e_i\left[\phi_i(e_i,\dot{e}_i)-\phi_i(0,\dot{e}_i)\right]$$
$$0\le \dot{e}_i\left[\phi_i(e_i,\dot{e}_i)-\phi_i(e_i,0)\right] \quad (5)$$
e) $\phi_i(e_i,0)=0\Rightarrow e_i=0$

for i=1,2,…,n.

2.1 Stability Analysis

In this subsection the proof of the closed loop fuzzy system (see fig. 1) stability is developed.

Theorem: Let the non autonomous, nonlinear, closed-loop fuzzy system (see Fig.1) obtained by combining (1) and (3):

$$\frac{d}{dt}\begin{bmatrix}\mathbf{e}\\ \dot{\mathbf{e}}\end{bmatrix}=\begin{bmatrix}\dot{\mathbf{e}}\\ -\boldsymbol{\Phi}(\mathbf{e},\dot{\mathbf{e}})\end{bmatrix}. \quad (6)$$

The properties (5) of $\phi_i(e_i,\dot{e}_i)$ are verified. Then, by choice of the following Lyapunov function:

$$V(\mathbf{e},\dot{\mathbf{e}})=\frac{1}{2}\dot{\mathbf{e}}^T\dot{\mathbf{e}}+\sum_{1=1}^{n}\int_0^{e_i}\phi_i(\xi_i,0)d\xi_i \quad (7)$$

the equilibrium state of the closed loop model (6) is globally asimptotically stable.

Proof. The first term of $V(\mathbf{e},\dot{\mathbf{e}})$ is a positive definite function with respect to $\dot{\mathbf{e}}$.

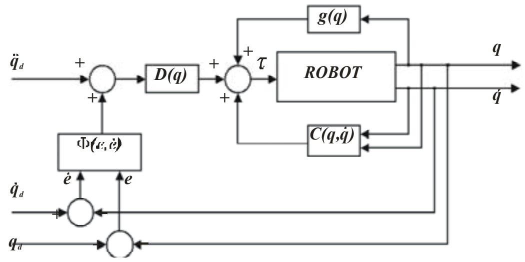

Fig.1 – Closed-loop system.

For the second term of (7), notice that, from properties (5b), (5d) and (5e) it results that

$$0<e_i\phi_i(e_i,0) \qquad \forall e_i\neq 0$$

This means that the following conditions are verified

$\phi_i(e_i,0)>0$ if $e_i>0$ [$\phi_i(e_i,0)<0$ if $e_i<0$]

from property (5c):

$$\int_0^{e_i}\phi_i(\xi_i,0)d\xi_i>0 \qquad \forall e_i\neq 0 \quad (8)$$

$$\int_0^{e_i}\phi_i(\xi_i,0)d\xi_i\to\infty \quad \text{for } e_i\to\infty$$

From (8) we can say that $V(\mathbf{e},\dot{\mathbf{e}})$ is globally positive defined and radially unbounded function; therefore (7) qualifies as a Lyapunov function.

The time derivative of the Lyapunov function is:

$$\dot{V}(\mathbf{e},\dot{\mathbf{e}})=\dot{\mathbf{e}}^T\ddot{\mathbf{e}}+\sum_{i=1}^{n}\frac{\partial}{\partial e_i}\left[\int_0^{e_i}\phi_i(\xi_i,0)d\xi_i\right]\dot{e}_i$$
$$=\dot{\mathbf{e}}^T\ddot{\mathbf{e}}+\dot{\mathbf{e}}^T\boldsymbol{\Phi}(\mathbf{e},\mathbf{0}) \quad (9)$$

By using (6), the time derivative of the Lyapunov function yields

$$\dot{V}(\mathbf{e},\dot{\mathbf{e}})=-\dot{\mathbf{e}}^T\left[\boldsymbol{\Phi}(\mathbf{e},\dot{\mathbf{e}})-\boldsymbol{\Phi}(\mathbf{e},\mathbf{0})\right] \quad (10)$$

Since $\boldsymbol{\Phi}(\mathbf{e},\dot{\mathbf{e}})$ is a decoupled nonlinearity of the form (4), we can use property (5d) to conclude that $\dot{V}(\mathbf{e},\dot{\mathbf{e}})$ is a globally negative semidefinite function. Thus, by invoking the Lyapunov's direct method [6] it is possible to conclude stability of the closed-loop system. In order to prove the closed-loop system global asymptotic stability, because the system is non autonomous, we use Barbalat's Lemma [6]. In order to use the Barbalat's Lemma we have to verify that $\ddot{V}(\mathbf{e},\dot{\mathbf{e}})$ is a bounded function.

The time derivative of (10) is:

$$\ddot{V}(\mathbf{e},\dot{\mathbf{e}})=-\ddot{\mathbf{e}}^T\left[\boldsymbol{\Phi}(\mathbf{e},\dot{\mathbf{e}})-\boldsymbol{\Phi}(\mathbf{e},\mathbf{0})\right]-\dot{\mathbf{e}}^T\left[\dot{\boldsymbol{\Phi}}(\mathbf{e},\dot{\mathbf{e}})-\dot{\boldsymbol{\Phi}}(\mathbf{e},\mathbf{0})\right] \quad (11)$$

By using (6), function (11) yields:

$$\ddot{V}(\mathbf{e},\dot{\mathbf{e}})=\boldsymbol{\Phi}^T(\mathbf{e},\dot{\mathbf{e}})\left[\boldsymbol{\Phi}(\mathbf{e},\dot{\mathbf{e}})-\boldsymbol{\Phi}(\mathbf{e},\mathbf{0})\right]-\dot{\mathbf{e}}^T\left[\dot{\boldsymbol{\Phi}}(\mathbf{e},\dot{\mathbf{e}})-\dot{\boldsymbol{\Phi}}(\mathbf{e},\mathbf{0})\right] \quad (12)$$

The first term of (12) is bounded because of $\boldsymbol{\Phi}(\mathbf{e},\dot{\mathbf{e}})$ is bounded. For the second term of (12) notice that it consists of two terms: velocity error vector $\dot{\mathbf{e}}$ and $\left[\dot{\mathbf{\Phi}}(\mathbf{e},\dot{\mathbf{e}})-\dot{\mathbf{\Phi}}(\mathbf{e},\mathbf{0})\right]$.

Because of (7) is a positive definite function and it is not increasing function of time, the arguments $(\mathbf{e},\dot{\mathbf{e}})$ are bounded. Thus the velocity error vector $\dot{\mathbf{e}}$ is bounded. $\dot{\mathbf{\Phi}}(\mathbf{e},\dot{\mathbf{e}})$ and $\dot{\mathbf{\Phi}}(\mathbf{e},\mathbf{0})$ are time derivatives of $\mathbf{\Phi}(\mathbf{e},\dot{\mathbf{e}})$ that is continous and bounded, so they are continous and bounded. We can conclude that (12) consists of two bounded terms, so it is bounded.

From Barbalat's Lemma results:

$$\lim_{t\to\infty}\dot{V}(t)=0 \qquad (13)$$

By using (9) we can see that (13) is verified in two cases:

1) $\dot{\mathbf{e}}=\mathbf{0}$

This condition implies that:

$$\mathbf{\Phi}(\mathbf{e},\mathbf{0})-\mathbf{\Phi}(\mathbf{e},\mathbf{0})=\mathbf{0} \quad \forall\mathbf{e}$$

By using (6) results

$$\ddot{\mathbf{e}}=\frac{d}{dt}\dot{\mathbf{e}}=-\mathbf{\Phi}(\mathbf{e},\dot{\mathbf{e}})$$

because $\dot{\mathbf{e}}=\mathbf{0}\Rightarrow\ddot{\mathbf{e}}=\mathbf{0}$. Due to verify this condition it is necessary that:

$$\mathbf{\Phi}(\mathbf{e},\mathbf{0})=\mathbf{0}$$

By using property (5e) the expression is verified only if **e=0**.

2) $\mathbf{\Phi}(\mathbf{e},\dot{\mathbf{e}})-\mathbf{\Phi}(\mathbf{e},\mathbf{0})=\mathbf{0}$

that is equal to:

$$\mathbf{\Phi}(\mathbf{e},\dot{\mathbf{e}})=\mathbf{\Phi}(\mathbf{e},\mathbf{0})$$

This condition is verified for $\dot{\mathbf{e}}=\mathbf{0}$. Also in this case, likewise case 1, we obtain **e=0**.

Therefore, invoking the Barbalat's Lemma, we conclude that the origin of state space $\mathbf{e}=\dot{\mathbf{e}}=\mathbf{0}$ is a globally asymptotitically stable equilibrium of the closed-loop system (6) (Q.E.D.).

2.2 Design of the Fuzzy Logic Controller

The synthetized FLC uses a set of four fuzzy IF-THEN rules (see Tab 1):

IF *Error* is *small* AND *DotError* is *small* THEN *Out* is *negBig*
IF *Error* is *small* AND *DotError* is *big* THEN *Out* is *negSmall*
IF *Error* is *big* AND *DotError* is *small* THEN *Out* is *posSmall*
IF *Error* is *big* AND *DotError* is *big* THEN *Out* is *posBig*

Tab. 1 – IF-THEN rules.

In this case we have two inputs and one output. The input and output membership functions are symmetrical with respect to zero (see Fig. 2). Fig. 2c shows that the system chosen is Sugeno-type system.

This kind of system generates the input-output mapping shown in Fig.3. Visual examination of this surface is useful to verify the properties (5).

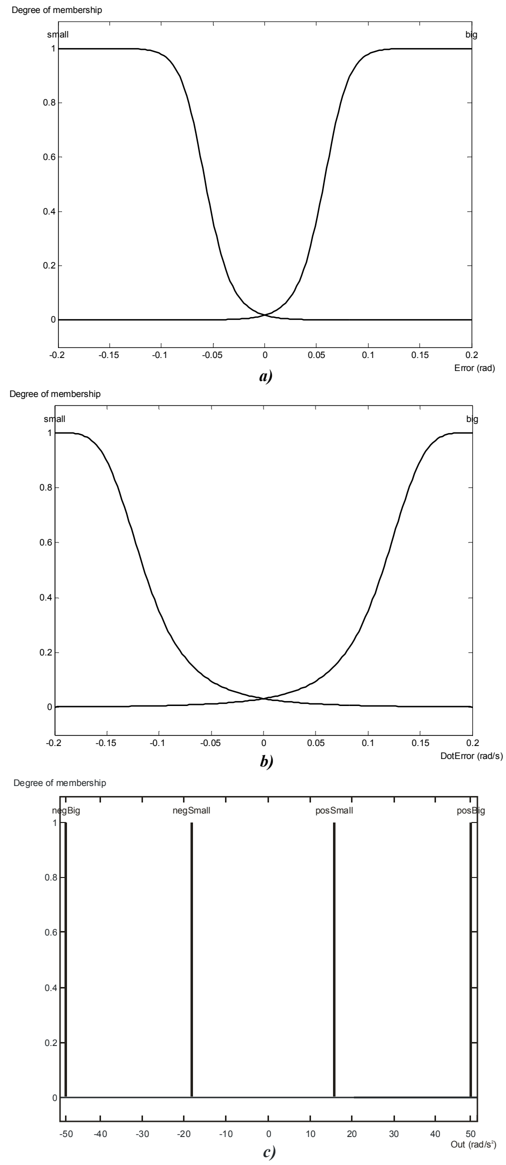

Fig.2 – Sugeno membership functions: a) e_i, b) de_i/dt c) output.

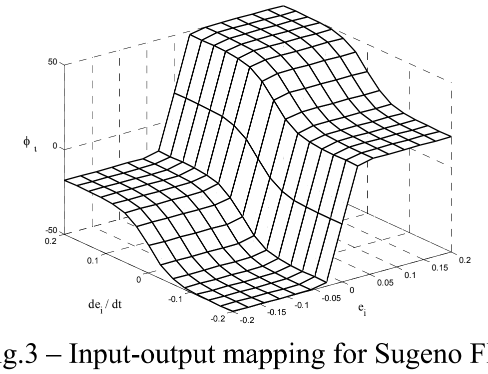

Fig.3 – Input-output mapping for Sugeno FIS.

2.3 Application of ANFIS Method

The ANFIS method generates a FIS using a neural network [4]. The ANFIS architecture is like a neural network that is functionally equivalent to a FIS. In this way, it is possibile to obtain fuzzy membership functions and inference rules starting from a neural network. The link between the ANFIS architecture and the physical system is given by data sets obtained by measurements on physical system. The data sets are used for the network learning algorithm. The hybrid learning algorithm used in this paper mixed least squares and backpropagation methods.

In order to obtain the membership functions the MATLAB ANFIS Editor GUI [7] have been used (see Fig.4). This kind of system generates the input-output mapping shown in Fig.5. Visual examination of this surface is useful to verify the properties (5).

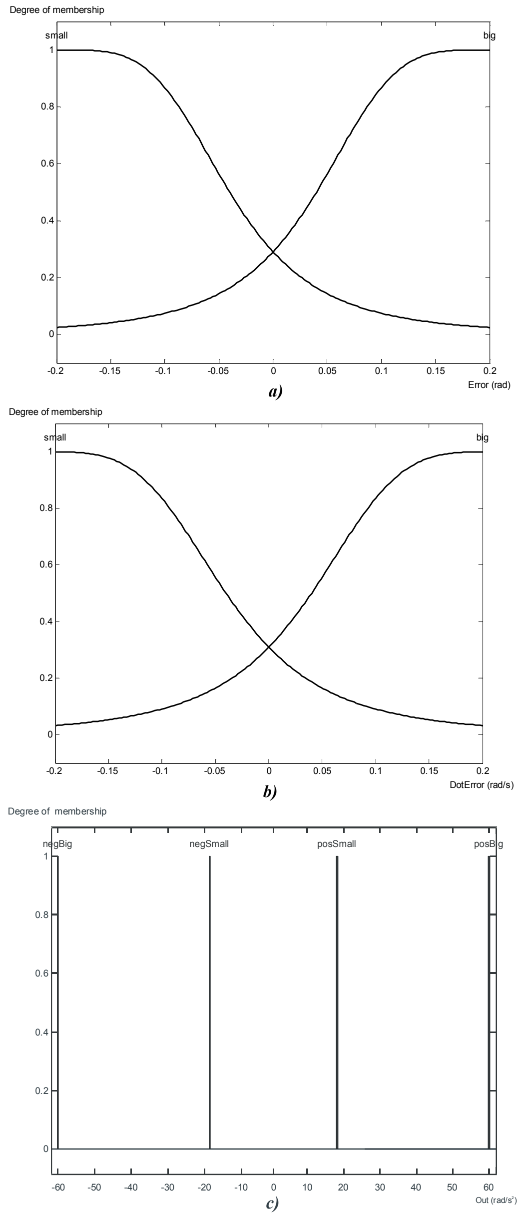

Fig.4 – ANFIS membership functions: a) e_i b) de_i/dt c) output

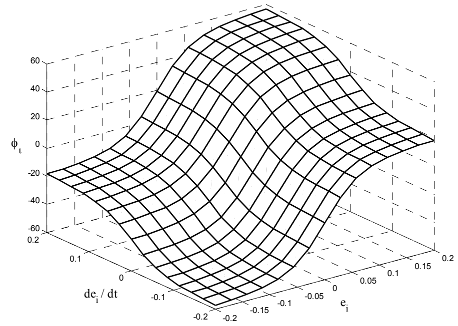

Fig.5 – Input-output mapping for ANFIS.

3 Experimental Results

For the experimental simulations a two link and two degree of freedom planar manipulator is used (see Fig.6).

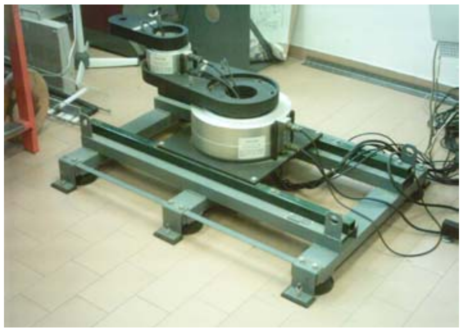

Fig.6 – Robotic Manipulator

The Inertia and Coriolis matrices of this manipulator are:

$$\mathbf{D}(\mathbf{q}) = \begin{bmatrix} p_1 + 2p_3\cos(q_2) & p_2 + p_3\cos(q_2) \\ p_2 + p_3\cos(q_2) & p_2 \end{bmatrix}$$

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -\dot{q}_2 p_3 \sin(q_2) & -(\dot{q}_2 + \dot{q}_1)p_3 \sin(q_2) \\ \dot{q}_1 p_3 \sin(q_2) & 0 \end{bmatrix}$$

where:

$$p_1 = 3.3165\left[Kgf \cdot m/s^2\right]$$
$$p_2 = 0.1168\left[Kgf \cdot m/s^2\right]$$
$$p_3 = 0.1630\left[Kgf \cdot m/s^2\right]$$

$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ (planar manipulator)

$\mathbf{f}(\dot{\mathbf{q}})$ unmodeled dynamics

The two links of manipulator move in the horizontal plane. During the simulation experiments we suppose the friction term $\mathbf{f}(\dot{\mathbf{q}})$ is unknown.

The motors are operated in torque mode, so they act as torque source and accept an analog voltage as reference torque signal.

According to the actuators manufacturer, the motors of manipulator are able to supply torques within the following bounds:

$$
\begin{aligned}
|\tau_1| &\le \tau_1^{\max} = 250[Nm] \\
|\tau_2| &\le \tau_2^{\max} = 40[Nm]
\end{aligned}
\qquad (14)
$$

Link 1 and link 2 are the bigger and the smaller arm respectively.

For the linearizing feedback controller we use the parameter matrices

$$
\mathbf{K_p} = \mathbf{K_d} = \begin{bmatrix} \mathbf{25} & \mathbf{0} \\ \mathbf{0} & \mathbf{2{,}5} \end{bmatrix}
$$

The referred signals for all the simulations are shown in Fig.7

$$
\begin{aligned}
q_d &= \sin(t) \\
\dot{q}_d &= \cos(t) \\
\ddot{q}_d &= -\sin(t)
\end{aligned}
$$

Two performance indexes were considered to verify the goodness of the controllers: ISE (Integral Square Error) and IAE (Integral Absolute Error)

$$
\mathrm{ISE} = \int_0^t \left(e_1^2 + e_2^2\right)dt \quad \mathrm{IAE} = \int_0^t \left(|e_1| + |e_2|\right)dt \qquad (15)
$$

The membership functions of ANFIS model were obtained by using the ANFIS Editor GUI of MATLAB FUZZY TOOLBOX [7].

We used two 50-points data sets: training and checking set. The first one is used to training the neural network, while the second is used for model validation.

The simulations show that the linearizing feedback controller is not able to minimize the tracking error particularly for link 2. Fig. 8 shows that position error on link 2 oscillates between –50 degrees and +50 degrees, while the position error on link 1 has acceptable values (between –0.2 degrees and +0.2 degrees).

Fig. 9 shows the position error for Sugeno FIS. The position error on link 1 oscillates between -0,025 and +0,025 degrees, while the position error on link 2 oscillates between –0,59 and +0,59 degrees.

Fig. 10 shows the position error for ANFIS. The position error on link 1 oscillates between –0,08 and +0,08 degrees, while the position error on link 2 oscillates between –1,87 and +1,87 degrees.

The applied torques to link 1 and link 2 are shown in Figs. 11-12. Fig. 11 shows that the applied torque to link 1 maximum value is 165 Nm (for Sugeno FIS), while the minimum value is –47 Nm (for Sugeno FIS). Fig. 12 shows that the applied torque to link 2 maximum value is 17 Nm (for Sugeno FIS), while the minimum value is –2 Nm (for Sugeno FIS). The obtained values remain inside the prescribed allowable maximum torque for each actuator (15). Tab. 2 summarizes the performance indexes. From Tab. 2 we can conclude that the FLC with Sugeno FIS presents the smallest tracking error.

In this paper, ANFIS controller is not the best. Data sets used for network learning algorithm include the friction and aleatories noises. Moreover, we have to consider the sampling error on manipulator data (see Fig.13).

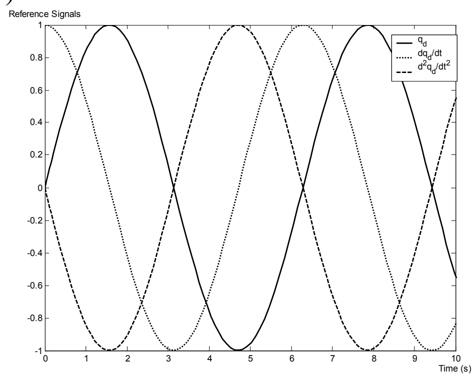

Fig.7 – Reference signals.

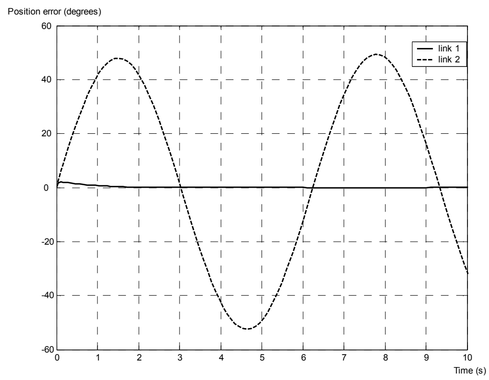

Fig.8 – Position error for linearizing feedback.

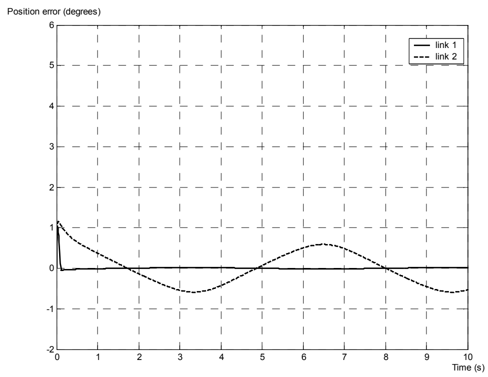

Fig.9 – Position error for Sugeno FIS.

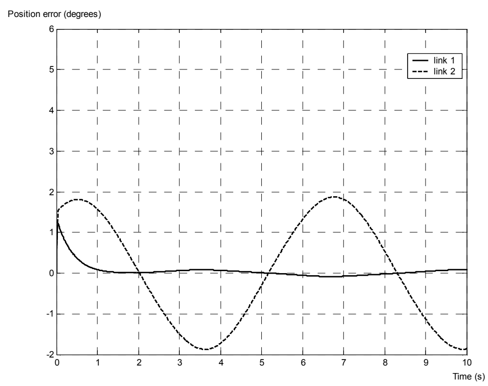

Fig.10 – Position error for ANFIS.

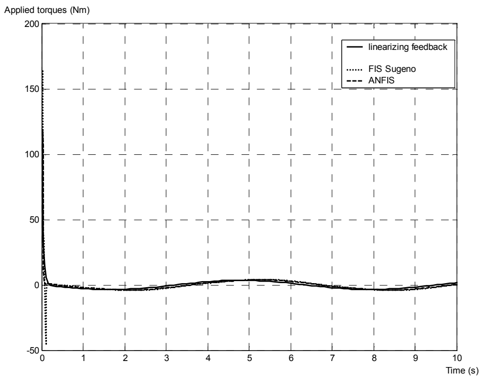

Fig.11 – Applied torques to link 1.

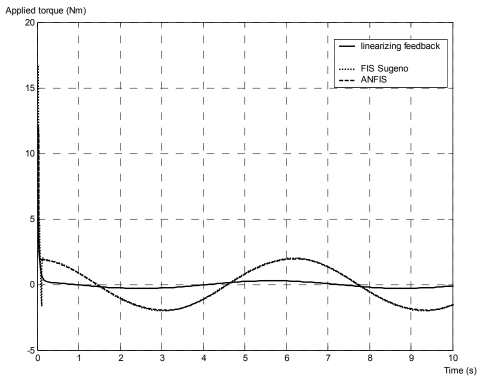

Fig.12 – Applied torques to link 2.

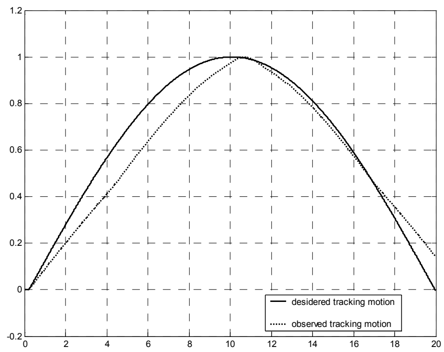

Fig. 13 – IMI manipulator tracking motion.

	ISE	IAE
linearizing feedback	11.74	17
Sugeno FIS	0.0015497	0.19679
ANFIS	0.015941	0.65150

Tab. 2 – Performance indexes.

4 Conclusion

The simulations show the goodness of the synthetized FLC. In order to minimize tracking error it is necessary to construct $\mathbf{\Phi}(\mathbf{e},\dot{\mathbf{e}})$ in such a way that properties (5) are verifyed.

Furthermore, a FIS using the ANFIS method have been used. In this way data sets obtained by measurement on the physical system under consideration have been used.

In this paper, ANFIS controller is not the best. Data sets used for network learning algorithm include the friction and aleatories noises. Moreover, we have to consider the sampling error on manipulator data.

Notwithstanding the ANFIS controller has position error smaller than linearizing feedback controller. Better results could be obtained by crossing a lot of data sets from the system under consideration. Otherwise we could use another network learning algorithm. The hybrid learning algorithm used in this paper mixed least squares and backpropagation methods. Once we obtain the neural network using ANFIS method, we can use any learning algorithm.

In conclusion the synthetized FLC is able to solve the tracking control problem for planar robotic manipulator. It is also able to minimize the effects of unmodeled uncertainties like friction, external disturbances or unknown loads.

References:

[1] M.A. Llama, R. Kelly, V. Santibanez, "Stable computed-torque control of robot manipulators: a fuzzy self-tuning procedure", *IEEE Trans. Syst., Man and Cybernetics B*, Vol.30, No.1 Feb. 2000, pp. 143-150.

[2] Y.Ch. Hsu, G. Chen, E. Sanchez, "A Fuzzy PD controller for multi-link robot control : stability analisys", *Proc. IEEE Int. Conf. Robotics and Automation*, Albuquerque, NM, 1997, pp. 1412-1417.

[3] L. Sciavicco, B. Siciliano, *"Modelling and Control of Robot Manipulators"*, 2nd Edition, Verlag Advanced Textbooks in Control and Signal Processing Series, UK, 2000.

[4] J.-S. R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System", *IEEE Trans. on Syst., Man and Cybernetics*, Vol.23, No.3, May/June 1993, pp.665-685.

[5] G. Calcev, "Some Remarks on the Stability of Mamdani Fuzzy Control Systems", *IEEE Transactions on Fuzzy Systems*, Vol.6, No.3, August 1998, pp. 436-442.

[6] J. E. Slotine e W. Li, *"Applied Nonlinear Control"*. Englewood Cliffs (USA): Prentice Hall, 1991.

[7] *"Fuzzy Logic Toolbox User's Guide"*, The Mathworks Inc., July 2002.