

## Artificial Control of Nonlinear Second Order Systems Based on AFGSMC

*Farzin Piltan, Shahnaz Tayebi Haghighi, Marzie Zare,  
Amin Jalali, Ali Roshanzamir, A. Reza Zare and Farhad Golshan*

Industrial Electrical and Electronic Engineering, Sanatkadehe Sabze Pasargad. CO (S.S.P. Co),  
NO: 16, PO. Code 71347-66773, Fourth Floor, Dena Apr, Seven Tir Ave, Shiraz, Iran

---

**Abstract:** Design robust controller for uncertain nonlinear systems most of time can be a challenging work. One of the most active research areas in this field is control of the nonlinear second order system. The control strategies for nonlinear systems are classified in two main groups: classical and non-classical methods, where the classical methods use the conventional control theory and non-classical methods use the artificial intelligence theory. Control nonlinear systems using pure classical controllers are often having lots of problems because most of time these systems have unknown variations in the parameters and have a large uncertainty. Artificial control such as Fuzzy logic, neural network, genetic algorithm and neurofuzzy control have been applied in many applications. Therefore, stable control of nonlinear dynamic systems is challenging because of some mentioned issues. In this paper the intelligent control of nonlinear second order system such as robot manipulator using Adaptive Fuzzy Gain Scheduling Sliding Mode Controller (AFGSMC) and comparison to Adaptive Fuzzy Inference System (AFIS) and various performance indices like the RMS error, Steady state error, trajectory performance, disturbance rejection and chattering are used for test the controller performance.

**Key words:** Uncertain nonlinear systems . classical control . non-classical control . fuzzy logic . intelligent control . robot manipulator . adaptive fuzzy gain scheduling sliding mode controller . adaptive fuzzy inference system . rms error . steady state error . chattering

---

### INTRODUCTION

Control nonlinear systems using classical controllers are based on nonlinear dynamic model. These controllers are often having many problems for modelling [1]. Conventional controllers require accurate knowledge of dynamic model of nonlinear plant, but most of time these models are multi-input, multi-output, non-linear and time-varying. However, operations in unstructured environment require systems to perform more complex tasks without an adequate analytical model. In case where models are available, it is questionable whether or not uncertainty is sufficiently accounted for [2].

Nonlinear control methodologies are more general for control of nonlinear systems by used nonlinear functions. Not all conventional nonlinear control methodologies can provide good robustness for uncertainty in robotic manipulator applications. Sliding mode controller is a powerful nonlinear controller, which has been analyzed by many researchers especially in recent years. The main reason to select this controller in wide range areas are have an acceptable control performance and solve two most important

challenging topics in control which names, stability and robustness [3].

The main reasons to use fuzzy logic technology are ability to give approximate recommended solving unclear and complex problem, easy to understand and flexible then a designer is able to model controller for any nonlinear plant with a set of IF-THEN rules, or it can identify the control actions and describe them by using fuzzy rules [4]. It must be noted that application of fuzzy logic is not limited to a system that difficult for modeling, but it can be used in clear systems that have complex mathematics models because most of time it can be shortened in design, but the quality of design may not always be so high. Besides to use fuzzy logic in the main controller of a control loop, it can be used to design adaptive control, tuning parameters, working in a parallel with genetic algorithm or artificial neural network. Most of time design fuzzy logic controllers are simpler than conventional controllers [2, 5-7].

In most of nonlinear systems, controllers are still usually classical linear and nonlinear. But these systems are multi inputs multi outputs, highly nonlinear and have uncertain or varying parameters (e.g., structure and unstructured) so that is complicate the used

classical controllers. One of the most important solution is used the adaptive controller (e.g., Adaptive Fuzzy Gain scheduling sliding mode controller). Adaptive control used in systems whose dynamic parameters are varying and need to be training on line. In general states adaptive control classified in two main groups: traditional adaptive method and fuzzy adaptive method, that traditional adaptive method need to have some information about dynamic plant and some dynamic parameters must be known but fuzzy adaptive method can training the variation of parameters by expert knowledge [8-16].

This paper is organized as follows:

In section 2, main subject of sliding mode controller and formulation are presented. This section covered the following details, classical sliding mode controller for robotic manipulator, equivalent control and chatter free sliding control. In section 3, modelling of robotic manipulators is presented. Detail of fuzzy logic controllers and fuzzy rule base is presented in section 4. In section 5, design Adaptive Fuzzy Gain Scheduling Sliding Mode Controller (AFGSMC); this method is used to reduce the uncertainty and variation in dynamic parameter. In section 6, design self tuning fuzzy inference system is presented to reduce the uncertainty and variation in parameters. In section 7, the simulation result is presented and finally in section 8, the conclusion is presented.

### INTRODUCTION TO CLASSICAL SLIDING MODE CONTROL FOR ROBOTIC MANIPULATOR

This section provides a review of classical sliding mode control and the problem of formulation based on [17-20]. Consider a nonlinear single input dynamic system of the form:

$$\dot{x}^{(n)} = f(\dot{x}) + b(\dot{x})u \tag{1}$$

where  $u$  is the vector of control input,  $x^{(n)}$  is the  $n$  th derivation of  $x$ ,  $x = [x, \dot{x}, \ddot{x}, \dots, x^{(n-1)}]^T$  is the state vector,  $f(x)$  is unknown and  $b(x)$  is of known sign. The control problem is truck to the desired state,  $x_d = [x_d, \dot{x}_d, \ddot{x}_d, \dots, x_d^{(n-1)}]^T$  and have an acceptable error which is given by:

$$\tilde{x} = x - x_d = [\tilde{x}, \dots, \tilde{x}^{(n-1)}]^T \tag{2}$$

Consider a time-varying sliding surface  $s(x, t)$  is given by:

$$s(x, t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{x} = 0 \tag{3}$$

where  $\lambda$  is the positive constant. Most of researcher used an integral term to further penalize tracking error as follows:

$$s(x, t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \left(\int_0^t \tilde{x} dt\right) = 0 \tag{4}$$

The main target in this methodology is keep  $s(x, t)$  at zero when tracking is outside of  $s(x, t)$ . Therefore, one of the common strategies is to find input  $U$  outside of  $s(x, t)$ .

$$\frac{1}{2} \frac{d}{dt} s^2(x, t) \leq -\zeta |s(x, t)| \tag{5}$$

where  $\zeta$  is strictly positive constant.

$$\text{If } S(0) > 0, \frac{d}{dt} S(t) \leq -\zeta \tag{6}$$

To eliminate the derivative term, we used an integral term from  $t=0$  to  $t=t_{reach}$

$$\int_{t=0}^{t=t_{reach}} \frac{d}{dt} S(t) dt < - \int_{t=0}^{t=t_{reach}} \eta dt \rightarrow S(t_{reach}) < S(0) - \zeta(t_{reach} - 0) \tag{7}$$

where  $t_{reach}$  is the time that trajectories reach to the sliding surface so,  $S(t_{reach} = 0)$  then:

$$0 - S(0) < -\eta(t_{reach}) \rightarrow t_{reach} < \frac{S(0)}{\zeta} \tag{8}$$

and

$$\begin{aligned} \text{if } S(0) < 0 \rightarrow 0 - S(0) < -\eta(t_{reach}) \\ \rightarrow S(0) < -\zeta(t_{reach}) \rightarrow t_{reach} < \frac{|S(0)|}{\eta} \end{aligned} \tag{9}$$

equation (9) guarantees that if trajectories are outside of  $S(t)$ , they will reach the sliding surface in a finite time smaller than  $\frac{|S(0)|}{\zeta}$ .

$$\text{if } S_{t_{reach}} = S(0) \rightarrow \text{error}(x - x_d) = 0 \tag{10}$$

If  $S$  defined as below:

$$s(x, t) = \left(\frac{d}{dt} + \lambda\right) \tilde{x} = (\dot{x} - \dot{x}_d) + \lambda(x - x_d) \tag{11}$$

The derivation of  $S$ , namely,  $\dot{S}$  can be calculated by:

$$\dot{S} = (\dot{x} - \dot{x}_d) + \lambda(x - x_d) \tag{12}$$

Second order systems can be defined by the following equation,

$$\ddot{x} = f + u \rightarrow \dot{S} = f + U - \ddot{x}_d + \lambda(\dot{x} - \dot{x}_d) \quad (13)$$

where f is the dynamic uncertain, if S = 0 and  $\dot{S} = 0$ , to have the best approximation,  $\hat{U}$  defined by,

$$\hat{U} = -f + \ddot{x}_d - \lambda(\dot{x} - \dot{x}_d) \quad (14)$$

A simple solution to get the sliding condition when the dynamic parameters have uncertainty is the switching control law:

$$U_{sw} = \hat{U} - K(\dot{x}, t) \cdot \text{sgn}(s) \quad (15)$$

where  $\text{sgn}(S)$  defined by,

$$\text{sgn}(s) = \begin{cases} 1 & s > 0 \\ -1 & s < 0 \\ 0 & s = 0 \end{cases} \quad (16)$$

where the  $K(\dot{x}, t)$  is the positive constant. Now we can rewrite the equation (5) by the following equation,

$$\begin{aligned} \frac{1}{2} \frac{d}{dt} s^2(x, t) &= \dot{s} \cdot s = [f - \ddot{x}_d - K \text{sgn}(s)] \cdot s \\ &= (f - \ddot{x}_d) \cdot s - K |s| \end{aligned} \quad (17)$$

Another method is using equation (4) instead of (3) to get sliding surface

$$\begin{aligned} s(x, t) &= \left( \frac{d}{dt} + \lambda \right)^2 \left( \int_0^t \ddot{x} dt \right) \\ &= (\dot{x} - \dot{x}_d) + 2\lambda(\dot{x} - \dot{x}_d) - \lambda^2(x - x_d) \end{aligned} \quad (18)$$

with this method the approximation of U can be calculated by

$$\hat{U} = -f + \ddot{x}_d - 2\lambda(\dot{x} - \dot{x}_d) + \lambda^2(x - x_d) \quad (19)$$

Since the control input U has to be discontinuous term, the control switching could not to be perfect and this will have chattering. Chattering can caused the high frequency oscillation of the controllers output and fast breakdown of mechanical elements in actuators. Chattering is one of the most important challenging in sliding mode controllers which, many papers have been presented to solve this problems [20].

Several different methods have been proposed to reduce or eliminate the chattering, but one of the most important methods is boundary layer method. In boundary layer method the basic idea is replace the discontinuous method by saturation (linear) method with small neighbourhood of the switching surface. This replace caused to increase the error performance. Several papers have been presented about trade-off between performance and chattering [4, 21, 22]. Therefore, the boundary layer used to have a smote control law,

$$B(t) = \{x, |s(t)| < \varnothing; \varnothing > 0 \quad (20)$$

where  $\varnothing$  is the boundary layer thickness. Therefore, to have a smote control law, the saturation function  $\text{Sat}(S/\varnothing)$  added to the control law:

$$U = K(\dot{x}, t) \cdot \text{Sat}(S/\varnothing) \quad (21)$$

$$\text{sat}(S/\varnothing) = \begin{cases} 1 & (S/\varnothing > 1) \\ -1 & (S/\varnothing < -1) \\ S/\varnothing & (-1 < S/\varnothing < 1) \end{cases} \quad (22)$$

The most important goal to design sliding mode controller is to have an acceptable performance and minimum error that, tracking error is defined as:

$$e = q_d - q_a \quad (23)$$

where

$$q_d = [q_{1d}, q_{2d}, q_{3d}]^T$$

is a desired and

$$q_a = [q_{1a}, q_{2a}, q_{3a}]^T$$

is an actual output. The sliding surface with integral part and derivative part is defined as follows:

$$S = \dot{e} + \lambda_1 e + \lambda_2 \int_0^t e \cdot dt \quad (24)$$

where

$$\lambda_1 = \text{diag}[\lambda_{11}, \lambda_{12}, \lambda_{13}]$$

is chosen as the bandwidth of the robot manipulator controller. In this state when  $S=0$  then  $e \rightarrow 0$  in  $t \rightarrow \infty$  and controller is stable. The time derivative of S can be calculated by the following equation

$$\dot{S} = \ddot{e} + \lambda_1 \dot{e} + \lambda_2 e \quad (25)$$

The Lyapunov function V defines as:

$$V = \frac{1}{2} S^T M S \quad (26)$$

Where M is positive symmetric matrix and V>0 for S≠0.

Based on above discussion, the control law for a multi DOF robot manipulator can be written as:

$$\tau = \tau_{eq} + \tau_{dis} \quad (27)$$

where, the model-based component  $\tau_{eq}$  compensate for the nominal dynamics of systems. So  $\tau_{eq}$  can calculate as follows:

$$\tau_{eq} = [M^{-1}(B + C + G) + \dot{S}]M \quad (28)$$

and  $\tau_{dis}$  can calculate as follows:

$$\tau_{dis} = K \cdot \text{sgn}(S) \quad (29)$$

To reduce the chattering problem, the saturation function introduced in control law instead of sign function as follows:

$$\tau_{sat} = K \cdot \text{sat}\left(\frac{S}{\rho}\right) \quad (30)$$

$$\tau = \tau_{eq} + K \cdot \text{sat}\left(\frac{S}{\rho}\right) = \begin{cases} \tau_{eq} + K \cdot \text{sgn}(S) & .|S| \geq \rho \\ \tau_{eq} + K \cdot S/\rho & .|S| < \rho \end{cases} \quad (31)$$

### MODELING OF ROBOTIC MANIPULATOR

Manipulator is a set of rigid links that connected by joints. From the control point of view, study of robot manipulators classified in two main parts: kinematics and dynamics which, Kinematics is one of the most important subject to find the relationship between rigid bodies such as position and orientation of manipulator and end-effectors of robot arm. The dynamic modeling describes the relationship between joint motion, velocity and accelerations to force/torque or current/voltage and also it can be used to describe the particular dynamic effects to behavior of system.

The dynamic equations of robot manipulators with rigid links can be written as:

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) = \Gamma \quad (32)$$

where,  $V(q, \dot{q})$  : Centrifugal and Coriolis forces,  $G(q)$  : Gravity forces: Generalized forces

M(q): Mass Matrix-Kinetic Energy Matrix, q: Generalized Joint Coordinates. By change of the nonlinearity parameter (Centrifugal and Coriolis) that depending to the velocity in form of position, all matrices parameters depending to the manipulator position. In this state the dynamic equation is called configuration space equation and can be write by the following form [23]:

If

$$V(q, \dot{q}) = B(q) \cdot [\dot{q}, \dot{q}] + C(q)[\dot{q}]^2 \Rightarrow \tau \quad (33)$$

$$= M(q) \cdot \ddot{q} + B(q)[\dot{q}, \dot{q}] + C(q)[\dot{q}] + G(q)$$

Where

$$B(q) = n \times \frac{n(n-1)}{2}$$

Matrix of Coriolis torques, C(q) = n×n Matrix of Centrifugal torques,

$$[\dot{q}\dot{q}] = \frac{n(n-1)}{2} \times 1$$

Vector of joint velocity and  $[\dot{q}] = n \times 1$  vector given by:  $[\dot{q}_1, \dot{q}_2, \dots]$ .

To modeling the robot manipulator Khatib's method (Explicit Form) used in this paper,

The angular acceleration can be calculated by the following equation:

$$\ddot{q} = M^{-1}(q) \cdot \{\tau - [B(q) \cdot \dot{q}\dot{q} + C(q) \cdot \dot{q} + g(q)]\} \quad (34)$$

In this paper first 3 DOF PUMA robot manipulator is modelling and analysis.

### DESIGN FUZZY LOGIC CONTROLLER

After the invention of fuzzy logic theory in 1965 by Zadeh [24], this theory was used in wide range area. Fuzzy Logic Controller (FLC) is one of the most important applications of fuzzy logic theory. This controller can be used to control of nonlinear, uncertain and noisy systems. Fuzzy logic control systems, do not use complex mathematical models of plant for analysis. This method is free of some model-based techniques that used in classical controllers. It must be noted that application of fuzzy logic is not limited only to modelling of nonlinear systems but also this method can help engineers to design easier controller [2, 5-7, 22].

The fuzzy inference mechanism provides a mechanism for referring the rule base in fuzzy set.

There are two most commonly method that can be used in fuzzy logic controllers, namely, Mamdani method and Sugeno method, which Mamdani built one of the first fuzzy controller to control of system engine and Michio Sugeno suggested to use a singleton as a membership function of the rule consequent. The Mamdani fuzzy inference method has four steps, namely, fuzzification, rule evaluation, aggregation of the rule outputs and defuzzification. Sugeno method is very similar to Mamdani method but Sugeno changed the consequent rule base that he used the mathematical function of the input rule base instead of fuzzy set. The following define can be shown the Mamdani and Sugeno fuzzy rule base

Mamdani F.R<sup>1</sup>: if x is A and y as B then z is C

Sugeno F.R<sup>1</sup>: if x is A and y as B then f(x, y) is C (35)

Fuzzification is used to determine the membership degrees for antecedent part when x and y have crisp values. Rule evaluation focuses on operation in the antecedent of the fuzzy rules. This part can used AND/OR fuzzy operation in antecedent part after that the output fuzzy set can be calculated by using individual rule-base inference. There are several methodologies in aggregation of the rule outputs that can be used in fuzzy logic controllers, namely, Max-Min aggregation, Sum-Min aggregation, Max-bounded product, Max-drastic product, Max-bounded sum, Max-algebraic sum and Min-max. In this paper we used Max-min aggregation. Max-min aggregation defined as below

$$\begin{aligned} \mu_U(x_k, y_k, U) &= \mu_{U_{i-1}FR^i}(x_k, y_k, U) \\ &= \max \left\{ \min_{i-1} \left[ \mu_{A_{i-1}}(x_k, y_k), \mu_{B_{i-1}}(U) \right] \right\} \end{aligned} \quad (36)$$

where r is the number of fuzzy rules activated by x<sub>k</sub> and y<sub>k</sub> and also  $\mu_{U_{i-1}FR^i}(x_k, y_k, U)$  is a fuzzy interpretation of i-th rule. The last step in the fuzzy inference in any fuzzy set is defuzzification. This part is used to transform fuzzy set to crisp set, therefore the input for defuzzification is the aggregate output and the output is a crisp number. There are several methodologies in defuzzification of the rule outputs that can be used in fuzzy logic controllers but this paper focouses on Center of gravity method (COG), which COG method used the following equation to calculate the defuzzification

$$COG(x_k, y_k) = \frac{\sum_i U_i \sum_{j=1}^r \mu_{U_i}(x_k, y_k, U_j)}{\sum_i \sum_{j=1}^r \mu_{U_i}(x_k, y_k, U_j)} \quad (37)$$

where COG(x<sub>k</sub>, y<sub>k</sub>) illustrates the crisp value of defuzzification output, U<sub>i</sub> ∈ U is discrete element of an output of the fuzzy set, μ<sub>U</sub>(x<sub>k</sub>, y<sub>k</sub>, U<sub>i</sub>) is the fuzzy set membership function and r is the number of fuzzy rules.

### DESIGN ADAPTIVE FUZZY GAIN SCHEDULING SLIDING MODE CONTROLLER (AFGSMC)

Adaptive control used in systems whose dynamic parameters are varying and need to be training on line. In general states adaptive control classified in two main groups: traditional adaptive method and fuzzy adaptive method, that traditional adaptive method need to have some information about dynamic plant and some dynamic parameters must be known but fuzzy adaptive method can training the variation of parameters by expert knowledge. Combined adaptive method to sliding mode controllers can help to controllers to have better performance by online tuning the nonlinear and time variant parameters.

For any plants (e.g., robot manipulators) whose have variation in parameter, adaptive control can learn the dynamic parameter to design an acceptable controller. All pure classical and fuzzy controllers have common difficulty, which they need to find several scale factors. Therefore, adaptive method can adjust and tune parameters [8-16].

The addition of adaptive methodology to a sliding mode controller caused to improve the tracking performance by online tuning the parameters. The adaptive sliding mode controller is used to estimate the unknown dynamic parameters and external disturbances. Several researchers works on adaptive sliding mode control and their applications in robotic manipulator in the following references [26-42].

Design supervisory FIS for classical SMC has five steps:

- Determine inputs and outputs: This controller has one input (S) and one output (α). The input is sliding surface (S) and the output is tuning coefficient (α).
- Find membership function and linguistic variable: The linguistic variables for sliding surface (S) are; Negative Big (N.B), Negative Medium (N.M), Negative Small (N.S), Zero (Z), Positive Small (P.S), Positive Medium (P.M), Positive Big (P.B) and it is quantized in to thirteen levels represented by:-1, -0.83, -0.66, -0.5, -0.33, -0.16, 0, 0.16, 0.33, 0.5, 0.66, 0.83, 1 and the linguistic variables to find the tuning coefficient (α) are; Negative Big (N.B), Negative Medium (N.M), Negative Small (N.S), Zero (Z), Positive Small

(P.S), Positive Medium (P.M), Positive Big (P.B) and it is quantized in to thirteen levels represented by:-1, -0.83, -0.66, -0.5, -0.33, -0.16, 0, 0.16, 0.33, 0.5, 0.66, 0.83, 1.

- Choice of shape of membership function: In this part the researcher select the triangular membership function that it is shown in Fig. 1.
- Design fuzzy rule table: design the rule base of fuzzy logic controller can play important role to design best performance AFGSMC, suppose that two fuzzy rules in this controller are

F.R<sup>1</sup>: IF S is Z, THEN  $\alpha$  is Z.

F.R<sup>2</sup>: IF S is (P.B) THEN  $\alpha$  is (L.R).

The complete rule base for this controller is shown in Table 1.

Table 1: Fuzzy rule table

S	N.B	N.M	N.S	Z	P.S	P.M	P.B
$\alpha$	N.B	N.M	N.S	Z	P.S	P.M	P.B

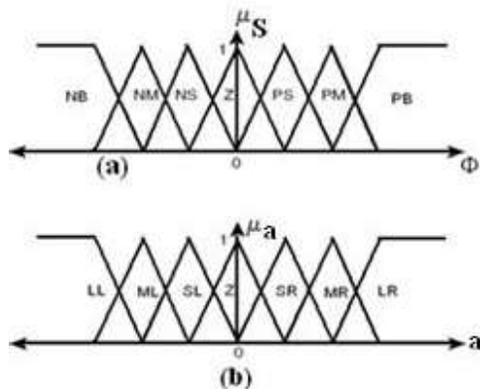


Fig. 1: Membership function: a) sliding surface b) Tuning coefficient

The control strategy that deduced by Table 1 are

- If sliding surface (S) is N.B, the control applied is N.B for moving S to S=0.
- If sliding surface (S) is Z, the control applied is Z for moving S to S=0.
- Defuzzification: The final step to design fuzzy logic controller is defuzzification, in this controller the COG method will be used.

The block diagram of AFGSMC controller is shown in Fig. 2.

### DESIGN SELF TUNING FUZZY INFERENCE SYSTEM (ST-FIS)

All conventional fuzzy logic controller have common difficulty, they need to find several parameters. Self-tuning FIS method can tune automatically the scale parameters using fuzzy rule base. Most of plants have two kinds of method for developing self-tuning controllers: the one is based on neural networks whereas the other uses fuzzy logic. To keep the structure of the controller as simple as possible and to avoid heavy computation, a fuzzy logic supervisor tuner based on fuzzy rule is selected [15]. In this method the supervisor controller tunes the output scaling factors using gain updating factors. This supervisory controller has two inputs ( $e, \dot{e}$ ) and one output ( $\alpha$ ). Two inputs are used for accessing the current conditions of the process. The inputs are error ( $e$ ) which measure the difference between desired and actual position and the rate of error ( $\dot{e}$ ) which measure the difference between desired and actual velocity and the output of this controller is  $\alpha$ , which can use to tune FIS. The linguistic variables for error( $e$ ) are; Negative Big(N.B), Negative Medium(N.M), Negative

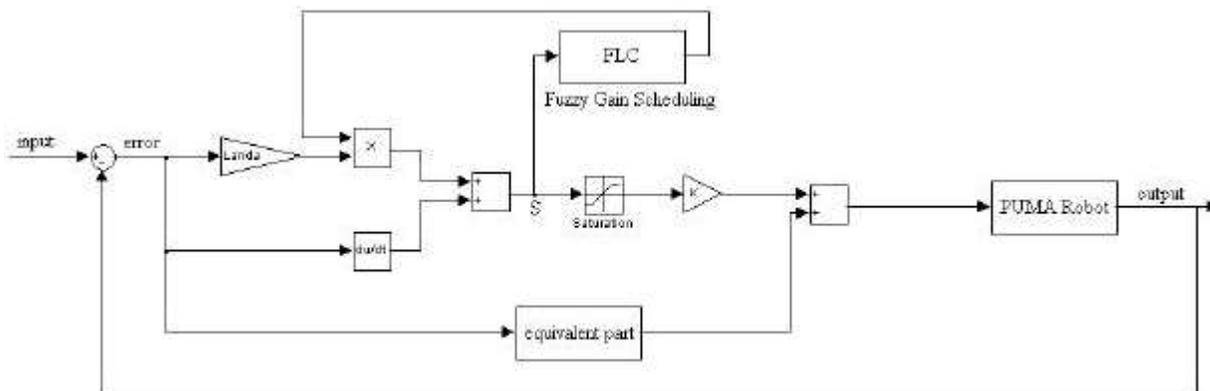


Fig. 2: Block diagram of an adaptive fuzzy gain scheduling sliding mode controller

Small(N.S), Zero(Z), Positive Small(P.S), Positive Medium(P.M), Positive Big(P.B) and it is quantized in to thirteen levels represented by: -1, -0.83, -0.66, 0.5, -0.33, -0.16, 0, 0.16, 0.33, 0.5, 0.66, 0.83, 1, the linguistic variables for change of error ( $\dot{e}$ ) are; Fast Left (FL), Medium Left (ML), Slow Left (SL), Zero (Z), Slow Right (SR), Medium Right(MR), Fast Right (FR) and it is quantized in to thirteen levels represented by: -6, -5, -0.4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6 and the linguistic variables to find the output are; Large Left (L.L), Medium Left (M.L), Small Left (S.L), Zero (Z), Small Right (S.R), Medium Right (M.R), Large Right

(L.R) and it is quantized in to thirteen levels represented by: -85, -70.8, -56.7, -42.5, -28.3, -14.2, 0, 14.2, 28.3, 42.5, 56.7, 70.8, 85. The triangular membership function, Fig. 3, used in this paper.

Design the rule base of fuzzy logic controller can play important role to design best performance fuzzy controller, suppose that the fuzzy rule in this controller is :

$$F.R^1: \text{IF } e \text{ is } Z \text{ and } \dot{e} \text{ is } Z, \text{ THEN } \alpha \text{ is } Z.$$

The complete rule base for this controller is shown in Table 2. To evaluate the conjunction of the rule antecedents, we use AND fuzzy operation intersection.

Table 3 shows the output value, that computed by COG method for AFIS.

The structure of self-tuning FIS is shown in Fig. 4. The scale factor, K1 and K2 are updated by equations (38) and (39) [43].

$$K_1^{new} = K_1^{old} \times \alpha \tag{38}$$

$$K_2^{new} = K_2^{old} \times \alpha \tag{39}$$

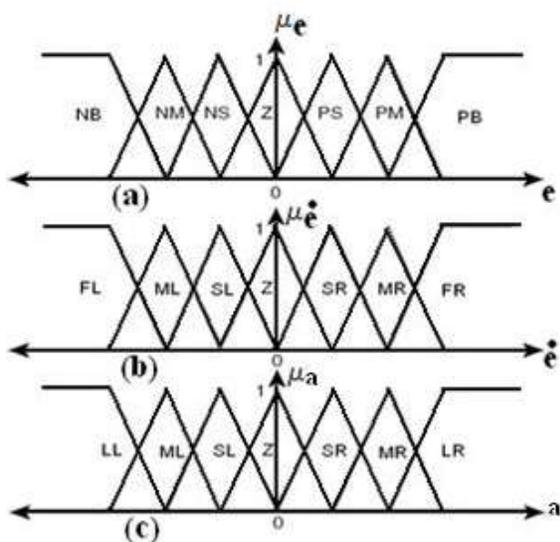


Fig. 3: Triangular membership function for error, change of error and  $\alpha$

Table 2: Fuzzy rule base table

$e/\dot{e}$	FL	ML	SL	Z	SR	MR	FR
NB	LL	LL	LL	ML	SL	SL	Z
NM	LL	ML	ML	ML	SL	Z	SR
NS	LL	ML	SL	SL	Z	SR	MR
Z	LL	ML	SL	Z	SR	MR	LR
PS	ML	SL	Z	SR	SR	MR	LR
PM	SL	Z	SR	MR	MR	MR	LR
PB	Z	SR	SR	MR	LR	LR	LR

Table 3

$\dot{e}$ (Membership Function)														
$e$	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	
-1	-85	-85	-85	-85	-85	-85	-85	-71	-40	-20	-10	-5	0	
-0.83	-85	-85	-85	-85	-70	-50	-50	-30	-30	-20	-10	0	0	
-0.66	-85	-85	-85	-70	-50	-50	-30	-30	-20	-10	0	0	20	
-0.50	-85	-85	-70	-50	-50	-30	-30	-20	-10	0	0	20	30	
-0.33	-85	-70	-50	-50	-30	-30	-20	-10	0	0	20	30	50	
-0.16	-70	-50	-50	-30	-30	-20	-10	0	0	20	30	50	50	
0.00	-70	-50	-50	-30	-30	-20	-10	0	20	30	40	50	60	
0.16	-50	-50	-30	-30	-20	-10	0	20	30	40	50	60	70	
0.33	-30	-30	-20	-20	-10	0	20	30	40	50	60	70	80	
0.50	-20	-10	0	20	30	40	50	60	70	80	80	80	85	
0.66	-10	0	20	30	40	50	60	70	80	80	80	85	85	
0.83	0	20	30	40	50	60	70	80	80	80	85	85	85	
1.00	0	20	30	40	50	60	80	85	85	85	85	85	85	

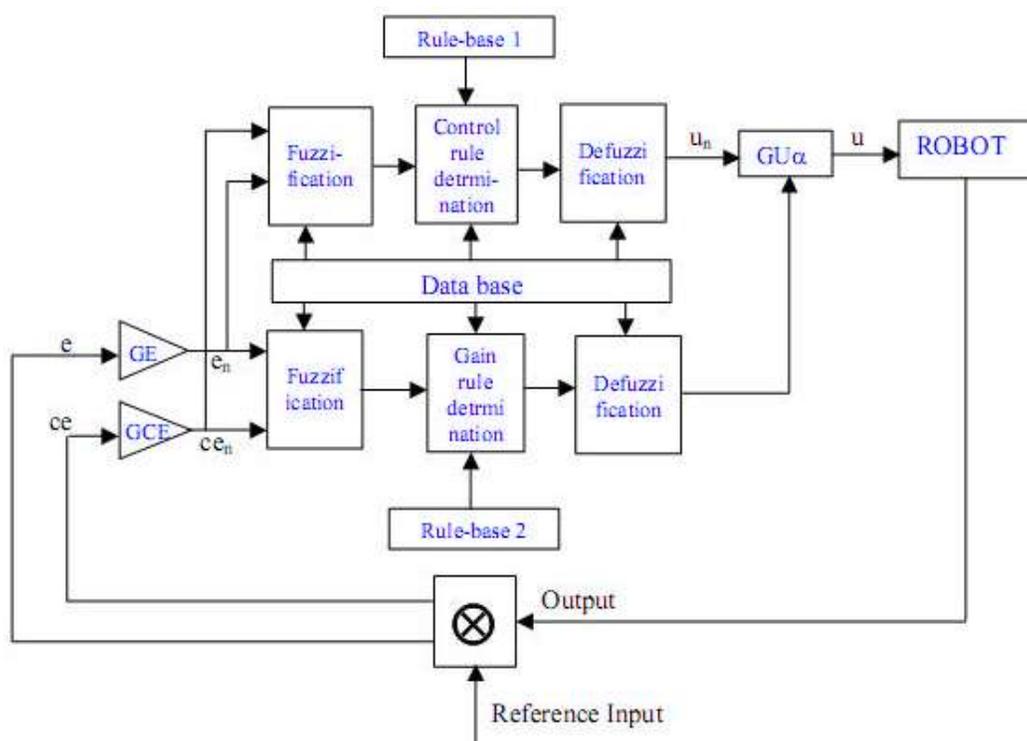


Fig. 4: Block diagram of self tuning controller (AFIS)

### SIMULATION RESULT

Adaptive Fuzzy Gain scheduling sliding mode controller (AFGSMC), Adaptive Fuzzy Inference System (AFIS) and Sliding Mode Controller (SMC) were tested for step response trajectories. In this simulation the first, second and third joints move from home to final position without and with external disturbance. The simulation was implemented in Matlab/Simulink environment. Tracking performance, error, robustness (disturbance rejection) and chattering rejection are compared.

Tracking performances: From the simulation for first, second and third trajectory without any disturbance, it was seen that AFGSMC and SMC have the same performance. This is primarily because the manipulator robot parameters do not change in simulation. The AFGSMC and SMC give significant trajectory good following when compared to FLC. Figure 5 shows tracking performance without any disturbance for AFGSMC, AFIS and SMC.

**Disturbance rejection:** An unknown output disturbance is applied in different time. Figure 6 shows disturbance rejection for AFGSMC, AFIS and SMC. However the AFGSMC gives the better performance than AFIS but AFIS also has an acceptable performance.

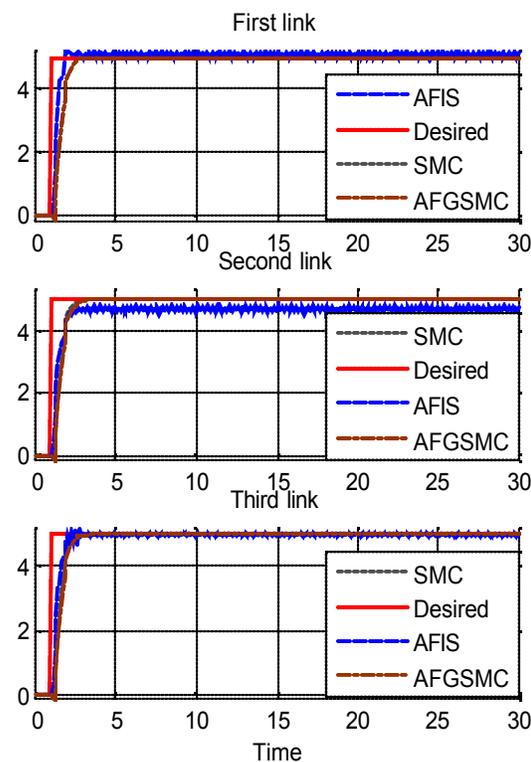


Fig. 5: Step SMC, AFIS and AFGSMC for First, second and third link trajectory without external disturbance

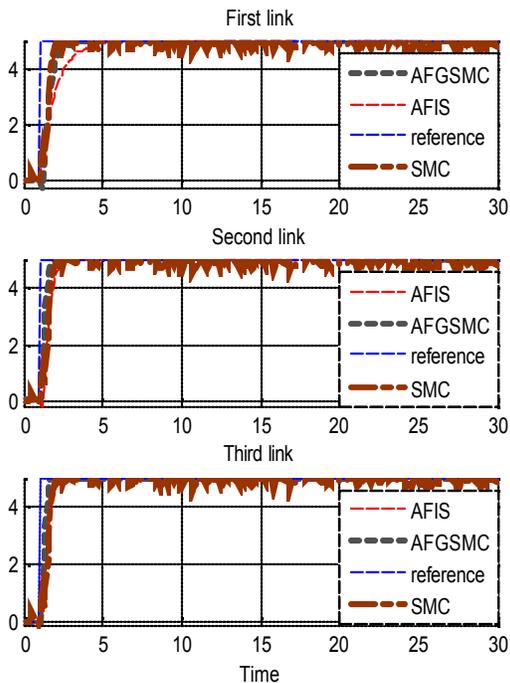


Fig. 6: Step SMC, AFIS and AFGSMC for First, second and third link trajectory with external disturbance

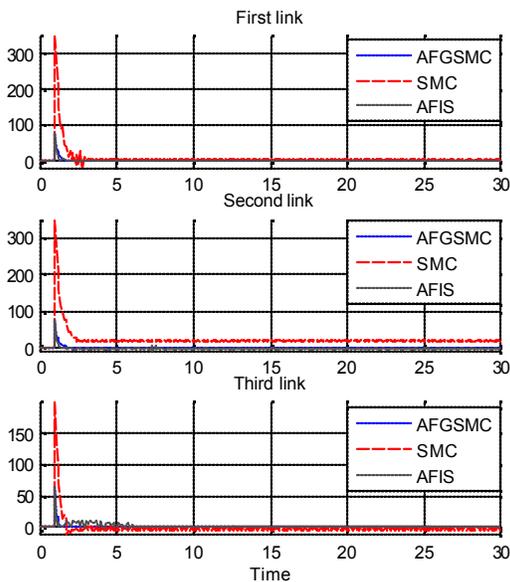


Fig. 7: Step SMC, AFIS and AFGSMC for First, second and third link errors with external disturbance

**Errors in the model:** However the AFIS gives significant error reduction when compared to pure FLC, but it is not as good as AFGSMC. The error profile for AFGSMC is smoother compared to the other

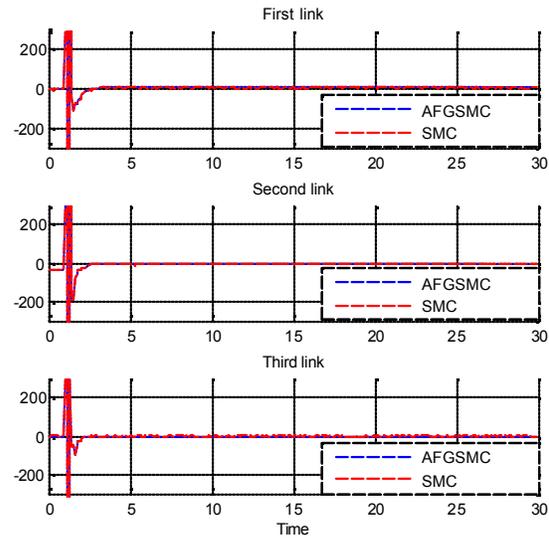


Fig. 8: Step SMC and AFGSMC for First, second and third link chattering rejection with external disturbance

controllers. Figure 7 shows a comparison of error performance for all three controllers that study in this paper.

**Chattering phenomenon:** An unknown output disturbance is applied in different time. Figure 8 shows the chattering rejection for step AFGSMC and SMC.

### CONCLUSION

This paper presents a new methodology for designing an adaptive fuzzy gain scheduling sliding mode controller for PUMA robotic manipulator. From the simulation, it was seen that AFGSMC has 7 rule base because it has one input for supervisory controller but AFIS has 49 rules for supervisory and 49 rules for main controller therefore implementing of AFIS most of time has many problems and expensive and also the AFGSMC performance is better than SMC and AFIS in most of time, Because this controller can auto tune as SMC with change the robot arm parameters, but pure SMC cannot do it.

The sliding mode controller alone displays problems in parameter variations. In the worst case, the adaptive controller has the potential to perform as well as a sliding mode controller. In AFGSMC, the fuzzy supervisory controller can changed the  $\lambda$  to achieve the best performance and in AFIS the supervisory controller can changed the gain updating factor of main FIS to have the best performance.

## ACKNOWLEDGEMENT

The authors would like to thanks to our God and our teachers and friends who gave us any help to writing this paper. Also the most thanks to our families and to our country IRAN which had bom us.

## REFERENCES

1. Desai, P., 1992. A Decentralized Adaptive Controller for Robot Manipulator Trajectory Control. S.G. Tzafestas, *Robotic Systems*, Netherlands, pp: 109-116.
2. Akbarzadeh-T.A.R., K. Kumbla, E. Tunstel and M. Jamshidi, 2000. Soft Computing for autonomous Robotic Systems. *IEEE International Conference on Systems, Man and Cybernatics*, pp: 5252-5258.
3. Thomas R. Kurfess, 2005. *Robotics and Automation Handbook*, CRC Press.
4. Wang, L.X., 1993. Stable adaptive fuzzy control of nonlinear systems. *IEEE Trans. Fuzzy Systems*, 1 (2): 146-154.
5. Zhou, J. and P. Coiffet, 1992. Fuzzy Control of Robots. *Proceedings IEEE International Conference on Fuzzy Systems*, pp: 1357-1364.
6. Banerjee, S. and Peng Yung Woo, 1993. Fuzzy logic control of robot manipulator. *Proceedings Second IEEE Conference on Control Applications*, pp: 87-88.
7. Lotfi, A. Zadeh, 1994. Fuzzy logic, Nural network and Soft computing. *Communications of the ACM*, 3 (37): 77-84.
8. Lin, C.K. and S.D. Wang, 1997. Robust self tuning rotated fuzzy basis function controller for robot arms. *IEE Proc-control Theory Appl.*, 144 (4): 293-298.
9. Elshafei, A.L. and A. Elnaggar, 1997. Adaptive versus adaptive fuzzy logic control. *IEEE, International Symposium on Intelligent Control*, pp: 49-54.
10. Kwan, E. and M. Liu, 1999. An adaptive fuzzy approach for robot manipulators tracking. *IEEE Conference on Computation Intelligent in Robotics and Automation*, pp: 53-58.
11. Liu, M., 2000. Stability analysis of decentralized adaptive fuzzy logic control for robot arm tracking, *IEEE Conference on Decision and Control*, pp: 883-888.
12. Berstecher, R.G., R. Palm and H.D. Unbehaven, 2001. An adaptive fuzzy sliding mode controller. *IEEE Trans. On Industrial Electronics*, 48 (1): 18-31.
13. Kim, V.T., 2002. Independent joint adaptive fuzzy control of robot manipulator. *IEEE Conference on Automation Congress*, pp: 645-652.
14. Wang, Y.F. and T.Y. Chai, 2004. Robust adaptive fuzzy observer design in robot arms. *IEEE Conference on Control*, pp: 857-862.
15. Sudeept Mohan and Surekha Bhanot, 2006. Comparative Study of Some Adaptive Fuzzy algorithms for Manipulator Control. *International Journal of Computation and Intelligence*, 3 (4): 303-311.
16. Sharma, R. and M. Gopal, 2008. A Markov game-adaptive fuzzy controller for robot manipulators. *IEEE Trans. On Fuzzy Systems*, 16 (1): 171-186.
17. Slotine, J.J.E., 1984. Sliding controller design for nonlinear systems. *Int. J. Control*, 40: 421-434.
18. Slotine, J.J.E. and J.A. Coetsee, 1986. Adaptive sliding controller synthesis for nonlinear systems. *Int. J. Control*, 43 (6): 1631-1651.
19. Slotine, J.J.E. and W. Li., 1987. On the adaptive control of robot manipulators. *Int. J. Robotics Research*, 6 (3): 49-59.
20. Utkin, I.V., 1977. Variable structure systems with sliding modes. *IEEE. Trans. Auto. Control*, 22 (2): 2121-222.
21. Palm, R., 1992. Sliding mode fuzzy control. *IEEE Conference*, pp: 519-526.
22. Emami, M.R., I.B. Turksen and A.A. Goldenberg, 1998. Development of a systematic methodology of fuzzy logic modelling. *IEEE Trans. On fuzzy Systems*, 6 (3): 346-361.
23. Bejczy, A., T.J. Taran, X. Yun and S. Man, 1985. Nonlinear Feedback Control of PUMA 560 Robot Arm by Computer. *IEEE Transaction Indian Electron*, 3 (24): 1680-1687.
24. Lotfi, A. Zadeh, 1997. Toward a theory of fuzzy information granulation and its centrality in human easoning and fuzzy logic. *Elsevier Science B.V Fuzzy Sets and Systems*, 1 (14): 111-127.
25. Reznik, L., 1997. *Fuzzy controllers*. B.H. Newnes.
26. Hsu, F.Y. and L.C. FU, 1995. Nonlinear control of robot manipulators using adaptive fuzzy sliding mode control. *IEEE Conference on Intelligent Robots and Systems*, pp: 156-161.
27. Yoo, B. and W. Ham, 1998. Adaptive fuzzy sliding mode control of nonlinear system. *IEEE Trans. On Fuzzy Systems*, 2 (6): 315-421.
28. Chain, C.C and C.C. Hu, 1999. Adaptive fuzzy controller for nonlinear uncertain systems, pp: 1131-1136.
29. Yoo, B.K. and W.C. Ham, 2000. Adaptive control of robot manipulator using fuzzy compensator. *IEEE Trans. On Fuzzy Systems*, 8 (2): 186-199.
30. Hwang, C.L., 2000. Takagi-sugeno-based robust and adaptive fuzzy sliding mode control systems. *IEEE Conference*, pp: 530-535.

31. Guo, Y. and P.Y. Woo, 2003. Adaptive fuzzy sliding mode control for robotic manipulators. IEEE Conference on Decision and Control, pp: 2174-2179.
32. Hwang, C.L. and S.F. Chao, 2004. A fuzzy model based variable structure control for robot arms: Theory and Experiments. IEEE Conference, pp: 5252-5258.
33. Sadati, N. and A. Talasaz, 2004. Chattering free adaptive fuzzy sliding mode control. IEEE Conference on Cybernetics and Intelligent Systems, pp: 29-34.
34. Lin, C.M. and C.F. Hsu, 2004. Adaptive fuzzy sliding mode control for induction servomotor systems. IEEE Trans. on Energy Conversion, 19 (2): 362-368.
35. Shahnazi, R., M.R. Akbarzadeh-T, 2005. Direct adaptive fuzzy PI sliding mode control for a class of uncertain nonlinear systems. IEEE Conferences on Systems Man and Cybernetics, pp: 2548-2553.
36. Shahnazi, R., H. Shanechi and N. Pariz, 2006. Position control of induction and servomotors: A novel adaptive fuzzy PI sliding mode control. IEEE Conferences on Power Engineering, pp: 1-9.
37. Shahnazi, R., H. Shanechi and N. Pariz, 2008. Position control of induction and servomotors: A novel adaptive fuzzy PI sliding mode control. IEEE Journals on Energy Conversions, 23 (1): 138-147.
38. Liu, J.K. and F.C. Sun, 2006. Global sliding mode control with adaptive fuzzy chattering free method for nonlinear system. IMAC Multi Conference on Computational Engineering in Systems Applications, pp: 541-546.
39. Medhaffar, H., N. Derbel and T. Damak, 2006. A decoupled fuzzy indirect adaptive sliding mode controller with application to robot manipulator. Int. Journal on Modeling, Identification and Control, 1 (1): 23-29.
40. Chiang, C.C. and C.H. Wu, 2007. Observer-based adaptive fuzzy sliding mode control of uncertain multi-input multi-output nonlinear systems. IEEE Fuzzy Systems Conference, pp: 1-6.
41. Weng, C.C. and W.S. Yu, 2008. Adaptive fuzzy sliding mode control for linear time-varying uncertain systems. IEEE Conference on Fuzzy Systems, pp: 1483-1490.
42. Yu, Z.X, 2009. Adaptive sliding mode-like fuzzy logic control for nonlinear systems. Journal of Communication and Computer, 6 (1): 53-60.
43. Canmarata, L. and L. Yliniemi, 1999. Development of self-tuning fuzzy logic controller for a rotary driver. University of Oulu.