

Design New Rule-based Effect Fuzzy Controller

Mina Mirzaie¹, Farzin Piltan¹, Nasri Sulaiman^{1,2} and Shahnaz Tayebi Haghighi¹

¹*Intelligent System and Robotic Lab, Iranian Institute of Advance Science and Technology (IRAN SSP), Shiraz/Iran*

²*Department of Electrical Engineering, Faculty of Engineering, University Putra Malaysia, Malaysia
piltan_f@iranssp.com, WWW.IRANSSP.COM/english*

Abstract

Rule base is one of the important key factors in design of fuzzy controller. The performance of the system is affected if nonlinear parameters are present in the system. This paper presents a method to select rule base size depending on the nonlinear parameters present in the system in parallel with robust sliding mode controller. The significance of the proposed method is tested on highly nonlinear second order system (robot manipulator) using modified rule bases. A criterion is developed for selection of rule base depending on amount of type of system: nonlinearity, multi input- multi output, time variant. The graphical and analytical results are given to show the importance of the proposed method.

Keywords: *Fuzzy logic control, rule base, second order system, sliding mode controller*

1. Introduction and Background

Robot manipulator is a N Degrees Of Freedom (DOF) serial or parallel links. In serial links robot manipulator the axes of first three joints are known as major axes, these axes show the position of end-effector and design a position controller is based on first three axes. One of the significant challenges in control algorithms is a linear behavior controller design for nonlinear systems (*e.g.*, robot manipulator). Some of robot manipulators which work in industrial processes are controlled by linear PD, proportional-integral-derivative (PID) controllers, but the design of linear controller for robot manipulators is extremely difficult because they are hardly nonlinear and uncertain [1-2, 6]. To reduce the above challenges, the nonlinear robust controller is used to compensate the linear control of robot manipulator.

Controller is a device which can sense information from linear or nonlinear system (*e.g.*, robot manipulator) to improve the systems performance [3]. The main targets in designing control systems are stability, good disturbance rejection, and small tracking error[4-5]. Several industrial robot manipulators are controlled by linear methodologies (*e.g.*, Proportional-Derivative (PD) controller, Proportional- Integral (PI) controller or Proportional- Integral-Derivative (PID) controller), but when robot manipulator works with various payloads and have uncertainty in dynamic models this technique has limitations. From the control point of view, uncertainty is divided into two main groups: uncertainty in unstructured inputs (*e.g.*, noise, disturbance) and uncertainty in structure dynamics (*e.g.*, payload, parameter variations). In some applications robot manipulators are used in an unknown and unstructured environment, therefore strong mathematical tools used in new control methodologies to design fuzzy PD controller based on sliding mode compensation to have an acceptable performance (*e.g.*, minimum error, good trajectory, disturbance rejection) [4-5].

Fuzzy-logic aims to provide an approximate but effective means of describing the behavior of systems that are not easy to describe precisely, and which are complex or ill-defined [6-7]. It is based on the assumption that, in contrast to Boolean logic, a statement can be partially true (or false). For example, the expression (I live near SSP.Co) where the fuzzy value (near) applied to the fuzzy variable (distance), in addition to being imprecise, is subject to interpretation. The essence of fuzzy control is to build a model of human expert who is capable of controlling the plant without thinking in terms of its mathematical model. As opposed to conventional control approaches where the focus is on constructing a controller described by differential equations, in fuzzy control the focus is on gaining an intuitive understanding (heuristic data) of how to best control the process [8-10], and then load this data into the control system. The experience has shown that fuzzy controllers are preferable choice for dynamic systems. PI type FLC are the most commonly used for process control applications. The performance of the fuzzy controller can be improved in number of ways. This can be done by considering the design parameters associated with the fuzzy controller. The parameters are classified as structural and tuning parameters. The tuning parameters are mainly scaling factors and parameters related to membership functions. Various methods have been proposed by researchers for the improvement of FLC using tuning parameters. The structural parameters are fuzzy inference, fuzzy membership functions and fuzzy rules, fuzzification and defuzzification methods [9].

Sliding mode control (SMC) is obtained by means of injecting a nonlinear discontinuous term. This discontinuous term is the one which enables the system to reject disturbances and also some classes of mismatches between the actual system and the model used for design [9-11]. These standard SMCs are robust with respect to internal and external perturbations, but they are restricted to the case in which the output relative degree is one. Besides, the high frequency switching that produces the sliding mode may cause chattering effect. The tracking error of SMC converges to zero if its gain is bigger than the upper bound of the unknown nonlinear function. Boundary layer SMC can assure no chattering happens when tracking error is less than; but the tracking error converges to ε ; it is not asymptotically stable [13-14]. A new generation of SMC using second-order sliding-mode has been recently developed by [10] and [11]. This higher order SMC preserves the features of the first order SMC and improves it in eliminating the chattering and fast convergence.

This paper focuses the use of type of system for selection of the rule base of FLC in fuzzy sliding mode controller. There are different aspects that have to be considered for selection of the rule base. The main idea in the new approach is rule base should be selected according the factor obtained from the systems nonlinear or controller's parameter. The rule base selection is given in tabulated form using the parameters of second order system models. The impact of the proposed method is shown by simulation on robot manipulator.

This paper is organized as follows; second part focuses on the modeling dynamic formulation based on Lagrange methodology. Third part is focused on the methodology which can be used to reduce the error, increase the performance quality and increase the robustness and stability based on fuzzy logic methodology and sliding mode controller to have a robust control with minimum rule base. Simulation result and discussion is illustrated in forth part which based on trajectory following and disturbance rejection. The last part focuses on the conclusion and compare between this method and the other ones.

2. Theory

Robot Manipulator's Dynamic: Dynamic modeling of robot manipulators is used to describe the behavior of robot manipulator such as linear or nonlinear dynamic behavior, design of model based controller such as pure sliding mode controller and pure computed torque controller which design these controller are based on nonlinear dynamic equations, and for simulation. 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 (e.g., inertia, coriolios, centrifugal, and the other parameters) to behavior of system[1]. The Unimation PUMA 560 serially links robot manipulator was used as a basis, because this robot manipulator is widely used in industry and academic. It has a nonlinear and uncertain dynamic parameters serial link 6 degrees of freedom (DOF) robot manipulator. The equation of an n -DOF robot manipulator governed by the following equation:

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau \quad (1)$$

Where τ is actuation torque, $M(q)$ is a symmetric and positive definite inertia matrix, $N(q, \dot{q})$ is the vector of nonlinearity term. This robot manipulator dynamic equation can also be written in a following form:

$$\tau = M(q)\ddot{q} + B(q)[\dot{q} \dot{q}] + C(q)[\dot{q}]^2 + G(q) \quad (2)$$

Where $B(q)$ is the matrix of coriolios torques, $C(q)$ is the matrix of centrifugal torques, and $G(q)$ is the vector of gravity force. The dynamic terms in equation (2) are only manipulator position. This is a decoupled system with simple second order linear differential dynamics. In other words, the component \ddot{q} influences, with a double integrator relationship, only the joint variable q_i , independently of the motion of the other joints. Therefore, the angular acceleration is found as to be:

$$\ddot{q} = M^{-1}(q). \{\tau - N(q, \dot{q})\} \quad (3)$$

This technique is very attractive from a control point of view.

3. Methodology

The model-free control strategy is based on the assumption that the joints of the manipulators are all independent and the system can be decoupled into a group of single-axis control systems. Therefore, the kinematic control method always results in a group of individual controllers, each for an active joint of the manipulator. With the independent joint assumption, no a priori knowledge of robot manipulator dynamics is needed in the kinematic controller design, so the complex computation of its dynamics can be avoided and the controller design can be greatly simplified. This is suitable for real-time control applications when powerful processors, which can execute complex algorithms rapidly, are not accessible. However, since joints coupling is neglected, control performance degrades as operating speed increases and a manipulator controlled in this way is only appropriate for relatively slow motion. The fast motion requirement results in even higher dynamic coupling between the various robot joints, which cannot be compensated for by a standard robot controller such as PD, and hence model-based control becomes the alternative. Based on above discussion;

$$e_1(t) = \theta_{desired}(t) - \theta_{actual}(t) \quad (4)$$

$$\tau_\alpha = K_{p_\alpha} e_1 + K_{v_\alpha} \dot{e}_1 \quad (5)$$

Consider a nonlinear single input dynamic system is defined by:

$$\mathbf{x}^{(n)} = \mathbf{f}(\tilde{\mathbf{x}}) + \mathbf{b}(\tilde{\mathbf{x}})\mathbf{u} \quad (6)$$

Where \mathbf{u} is the vector of control input, $\mathbf{x}^{(n)}$ is the n^{th} derivation of \mathbf{x} , $\mathbf{x} = [\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}, \dots, \mathbf{x}^{(n-1)}]^T$ is the state vector, $\mathbf{f}(\mathbf{x})$ is unknown or uncertainty, and $\mathbf{b}(\mathbf{x})$ is of known *sign* function. The main goal to design this controller is train to the desired state; $\mathbf{x}_d = [\mathbf{x}_d, \dot{\mathbf{x}}_d, \ddot{\mathbf{x}}_d, \dots, \mathbf{x}_d^{(n-1)}]^T$, and trucking error vector is defined by:

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d = [\tilde{x}, \dots, \tilde{x}^{(n-1)}]^T \quad (7)$$

A time-varying sliding surface $\mathbf{s}(\mathbf{x}, \mathbf{t})$ in the state space \mathbf{R}^n is given by:

$$\mathbf{s}(\mathbf{x}, \mathbf{t}) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{\mathbf{x}} = \mathbf{0} \quad (8)$$

where λ is the positive constant. To further penalize tracking error, integral part can be used in sliding surface part as follows:

$$\mathbf{s}(\mathbf{x}, \mathbf{t}) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \left(\int_0^t \tilde{\mathbf{x}} dt\right) = \mathbf{0} \quad (9)$$

The main target in this methodology is kept the sliding surface slope $\mathbf{s}(\mathbf{x}, \mathbf{t})$ near to the zero. Therefore, one of the common strategies is to find input \mathbf{U} outside of $\mathbf{s}(\mathbf{x}, \mathbf{t})$.

$$\frac{1}{2} \frac{d}{dt} \mathbf{s}^2(\mathbf{x}, \mathbf{t}) \leq -\zeta |\mathbf{s}(\mathbf{x}, \mathbf{t})| \quad (10)$$

where ζ is positive constant.

$$\text{If } \mathbf{S}(0) > 0 \rightarrow \frac{d}{dt} \mathbf{S}(t) \leq -\zeta \quad (11)$$

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

$$\int_{t=0}^{t=t_{reach}} \frac{d}{dt} \mathbf{S}(t) \leq - \int_{t=0}^{t=t_{reach}} \eta \rightarrow \mathbf{S}(t_{reach}) - \mathbf{S}(0) \leq -\zeta(t_{reach} - 0) \quad (12)$$

Where t_{reach} is the time that trajectories reach to the sliding surface so, suppose $\mathbf{S}(t_{reach} = 0)$ defined as;

$$0 - \mathbf{S}(0) \leq -\eta(t_{reach}) \rightarrow t_{reach} \leq \frac{\mathbf{S}(0)}{\zeta} \quad (13)$$

And

$$\text{if } \mathbf{S}(0) < 0 \rightarrow 0 - \mathbf{S}(0) \leq -\eta(t_{reach}) \rightarrow \mathbf{S}(0) \leq -\zeta(t_{reach}) \rightarrow t_{reach} \leq \frac{|\mathbf{S}(0)|}{\eta} \quad (14)$$

Equation (14) guarantees time to reach the sliding surface is smaller than $\frac{|\mathbf{S}(0)|}{\zeta}$ since the trajectories are outside of $\mathbf{S}(t)$.

$$\text{if } S_{t_{reach}} = S(0) \rightarrow \text{error}(x - x_d) = 0 \quad (15)$$

suppose S is defined as

$$s(x, t) = \left(\frac{d}{dt} + \lambda\right) \tilde{x} = (\dot{x} - \dot{x}_d) + \lambda(x - x_d) \quad (16)$$

The derivation of S, namely, \dot{S} can be calculated as the following;

$$\dot{S} = (\ddot{x} - \ddot{x}_d) + \lambda(\dot{x} - \dot{x}_d) \quad (17)$$

suppose the second order system is defined as;

$$\ddot{x} = f + u \rightarrow \dot{S} = f + U - \ddot{x}_d + \lambda(\dot{x} - \dot{x}_d) \quad (18)$$

Where f is the dynamic uncertain, and also since $S = 0$ and $\dot{S} = 0$, to have the best approximation, \hat{U} is defined as

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

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

$$U_{dis} = \hat{U} - K(\vec{x}, t) \cdot \text{sgn}(s) \quad (20)$$

where the switching function $\text{sgn}(S)$ is defined as

$$\text{sgn}(s) = \begin{cases} 1 & s > 0 \\ -1 & s < 0 \\ 0 & s = 0 \end{cases} \quad (21)$$

and the $K(\vec{x}, t)$ is the positive constant. Suppose by (10) the following equation can be written as,

$$\frac{1}{2} \frac{d}{dt} s^2(x, t) = \dot{S} \cdot S = [f - \hat{f} - K \text{sgn}(s)] \cdot S = (f - \hat{f}) \cdot S - K|S| \quad (22)$$

and if the equation (14) instead of (13) the sliding surface can be calculated as

$$s(x, t) = \left(\frac{d}{dt} + \lambda\right)^2 \left(\int_0^t \tilde{x} dt\right) = (\dot{x} - \dot{x}_d) + 2\lambda(\dot{x} - \dot{x}_d) - \lambda^2(x - x_d) \quad (23)$$

in this method the approximation of U is computed as

$$\hat{U} = -\hat{f} + \ddot{x}_d - 2\lambda(\dot{x} - \dot{x}_d) + \lambda^2(x - x_d) \quad (24)$$

Based on above discussion, the sliding mode control law for a multi degrees of freedom robot manipulator is written as:

$$\tau = \tau_{eq} + \tau_{dis} \quad (25)$$

Where, the model-based component τ_{eq} is the nominal dynamics of systems calculated as follows:

$$\tau_{eq} = [M^{-1}(B + C + G) + \dot{S}]M \quad (26)$$

and τ_{dis} is computed as;

$$\tau_{dis} = K \cdot \text{sgn}(S) \quad (27)$$

By (27) and (26) the sliding mode control of robot manipulator is calculated as;

$$\tau = [M^{-1}(B + C + G) + \dot{S}]M + K \cdot \text{sgn}(S) \quad (28)$$

where $S = \lambda e + \dot{e}$ in PD-SMC and $S = \lambda e + \dot{e} + (\frac{\lambda}{2})^2 \sum e$ in PID-SMC.

The Lyapunov formulation can be written as follows,

$$V = \frac{1}{2} S^T \cdot M \cdot S \quad (29)$$

the derivation of V can be determined as,

$$\dot{V} = \frac{1}{2} S^T \cdot \dot{M} \cdot S + S^T M \dot{S} \quad (30)$$

the dynamic equation of robot manipulator can be written based on the sliding surface as

$$M\dot{S} = -VS + M\dot{S} + B + C + G \quad (31)$$

it is assumed that

$$S^T(\dot{M} - 2B + C + G)S = 0 \quad (32)$$

by substituting (31) in (30)

$$\dot{V} = \frac{1}{2} S^T \dot{M} S - S^T B + CS + S^T(M\dot{S} + B + CS + G) = S^T(M\dot{S} + B + CS + G) \quad (33)$$

suppose the control input is written as follows

$$\hat{U} = U_{\widehat{Nonlinear}} + \widehat{U}_{dis} = [\widehat{M}^{-1}(B + C + G) + \dot{S}]\widehat{M} + K \cdot \text{sgn}(S) + B + CS + G \quad (34)$$

by replacing the equation (34) in (29)

$$\dot{V} = S^T(M\dot{S} + B + C + G - \widehat{M}\dot{S} - \widehat{B} + \widehat{C}S + G - K\text{sgn}(S)) = S^T(\widetilde{M}\dot{S} + \widetilde{B} + \widetilde{C}S + G - K\text{sgn}(S)) \quad (35)$$

And

$$|\widetilde{M}\dot{S} + \widetilde{B} + CS + G| \leq |\widetilde{M}\dot{S}| + |\widetilde{B} + CS + G| \quad (36)$$

the Lemma equation in robot arm system can be written as follows

$$K_u = [|\widetilde{M}\dot{S}| + |B + CS + G| + \eta]_i, i = 1, 2, 3, 4, \dots \quad (37)$$

and finally;

$$\dot{V} \leq - \sum_{i=1}^n \eta_i |S_i| \quad (38)$$

Based on foundation of fuzzy logic methodology; fuzzy logic controller has played important rule to design nonlinear controller for nonlinear and uncertain systems. However the application area for fuzzy control is really wide, the basic form for all command types of controllers consists of;

Input fuzzification (binary-to-fuzzy [B/F] conversion)

Fuzzy rule base (knowledge base), Inference engine and Output defuzzification (fuzzy-to-binary [F/B] conversion). Figure 1 shows the fuzzy controller part.

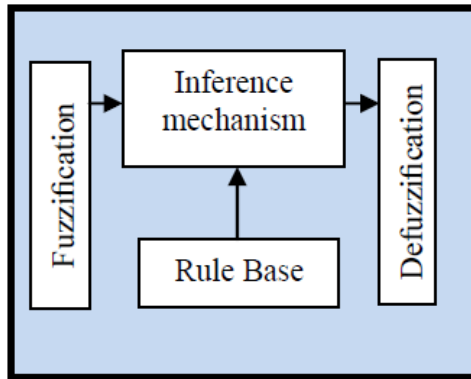


Figure 1. Fuzzy Controller Part

The fuzzy inference engine offers a mechanism for transferring the rule base in fuzzy set which it is divided into two most important methods, namely, Mamdani method and Sugeno method. Mamdani method is one of the common fuzzy inference systems and he designed one of the first fuzzy controllers to control of system engine. Mamdani's fuzzy inference system is divided into four major steps: fuzzification, rule evaluation, aggregation of the rule outputs and defuzzification. Michio Sugeno use a singleton as a membership function of the rule consequent part. The following definition shows the Mamdani and Sugeno fuzzy rule base

$$\begin{aligned} & \text{if } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z \text{ is } C \text{ 'mamdani'} \\ & \text{if } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z \text{ is } f(x, y) \text{ 'sugeno'} \end{aligned} \quad (39)$$

When x and y have crisp values fuzzification calculates the membership degrees for antecedent part. Rule evaluation focuses on fuzzy operation (AND/OR) in the antecedent of the fuzzy rules. The aggregation is used to calculate the output fuzzy set and several methodologies can be used in fuzzy logic controller aggregation, namely, Max-Min aggregation, Sum-Min aggregation, Max-bounded product, Max-drastic product, Max-bounded sum, Max-algebraic sum

and Min-max. Defuzzification is the last step in the fuzzy inference system which it is used to transform fuzzy set to crisp set. Consequently defuzzification's input is the aggregate output and the defuzzification's output is a crisp number. Centre of gravity method (COG) and Centre of area method (COA) are two most common defuzzification methods.

Based on the dynamic formulation of robot manipulator, (3), and the industrial PD law (5) in this paper we discuss about regulation problem, the desired position is constant, *i.e.*, $\dot{q}_d = 0$. In most robot manipulator control, desired joint positions are generated by the trajectory planning. The objective of robot control is to design the input torque in (1) such that the tracking error

$$e = q_d - q_a \quad (40)$$

When the dynamic parameters of robot formulation known, the PD control formulation (11) should include a compensator as

$$\tau = -k_p e - k_d \dot{e} + (G + F) \quad (41)$$

Where G is gravity and F is appositve definite diagonal matrix friction term (coulomb friction).

If we use a Lyapunov function candidate as

$$V_{pd} = \frac{1}{2} \dot{q}^T M \dot{q} + \frac{1}{2} e^T k_p e \quad (42)$$

$$\dot{V}_{pd} = -\dot{q}^T k_d \dot{q} \leq 0 \quad (43)$$

It is easy to known $\dot{q} = 0$ and $e = 0$ are only initial conditions in $\Omega = \{[\dot{q}, e]: \dot{V} = 0\}$, for which $[\dot{q}, e] \in \Omega$ for al $lt \leq 0$. By the LaSalle's invariance principle, $e \rightarrow 0$ and $\dot{e} \rightarrow 0$. When G and F in (11) are unknown, a fuzzy logic can be used to approximate them as

$$f(x) = \sum_{l=1}^M \theta^l \mathcal{E}^l(x) = \theta^T \mathcal{E}(x) \quad (44)$$

Where

$\theta = (\theta^1, \dots, \theta^M)^T$, $\mathcal{E}(x) = (\mathcal{E}^1(x), \dots, \mathcal{E}^M(x))^T$, and $\mathcal{E}^l(x) =: \prod_{i=1}^n \frac{\mu_{A_i^l}(x_i)}{\sum_{l=1}^M \prod_{i=1}^n \mu_{A_i^l}(x_i)}$. $\theta^1, \dots, \theta^M$ are adjustable parameters in (44). $\mu_{A_1^1}(x_1), \dots, \mu_{A_n^m}(x_n)$ are given membership functions whose parameters will not change over time.

The second type of fuzzy systems is given by

$$f(x) = \frac{\sum_{l=1}^M \theta^l \left[\prod_{i=1}^n \exp\left(-\left(\frac{x_i - \alpha_i^l}{\delta_i^l}\right)^2\right) \right]}{\sum_{l=1}^M \left[\prod_{i=1}^n \exp\left(-\left(\frac{x_i - \alpha_i^l}{\delta_i^l}\right)^2\right) \right]} \quad (45)$$

Where θ^l , α_i^l and δ_i^l are all adjustable parameters. From the universal approximation theorem, we know that we can find a fuzzy system to estimate any continuous function. For the first type of fuzzy systems, we can only adjust θ^l in (45). We define $f^\wedge(x|\theta)$ as the approximator of the real function $f(x)$.

$$f^\wedge(x|\theta) = \theta^T \varepsilon(x) \tag{46}$$

We define θ^* as the values for the minimum error:

$$\theta^* = \arg \min_{\theta \in \Omega} \left[\sup_{x \in U} |f^\wedge(x|\theta) - g(x)| \right] \tag{47}$$

Where Ω is a constraint set for θ . For specific x , $\sup_{x \in U} |f^\wedge(x|\theta^*) - f(x)|$ is the minimum approximation error we can get.

We used the first type of fuzzy systems (44) to estimate the nonlinear system (12) the fuzzy formulation can be write as below;

$$f(x|\theta) = \theta^T \varepsilon(x) \tag{48}$$

$$= \frac{\sum_{l=1}^n \theta^l [\mu_{A^l}(x)]}{\sum_{l=1}^n [\mu_{A^l}(x)]}$$

Where $\theta^1, \dots, \theta^n$ are adjusted by an adaptation law. The adaptation law is designed to minimize the parameter errors of $\theta - \theta^*$. The SISO fuzzy system is define as

$$f(x) = \Theta^T \varepsilon(x) \tag{49}$$

Where

$$\Theta^T = (\theta_1, \dots, \theta_m)^T = \begin{bmatrix} \theta_1^1, \theta_1^2, \dots, \theta_1^M \\ \theta_2^1, \theta_2^2, \dots, \theta_2^M \\ \vdots \\ \theta_m^1, \theta_m^2, \dots, \theta_m^M \end{bmatrix} \tag{50}$$

$\varepsilon(x) = (\varepsilon^1(x), \dots, \varepsilon^M(x))^T$, $\varepsilon^1(x) = \prod_{i=1}^n \mu_{A_i^1}(x_i) / \sum_{l=1}^M (\prod_{i=1}^n \mu_{A_i^l}(x_i))$, and $\mu_{A_i^l}(x_i)$ is defined in (48). To reduce the number of fuzzy rules, we divide the fuzzy system in to three parts:

$$F^1(q, \dot{q}) = \Theta^1{}^T \varepsilon(q, \dot{q}) \tag{51}$$

$$= \left[\theta_1^1{}^T \varepsilon(q, \dot{q}), \dots, \theta_m^1{}^T \varepsilon(q, \dot{q}) \right]^T$$

$$F^2(q, \ddot{q}_r) = \Theta^2{}^T \varepsilon(q, \ddot{q}_r) \tag{52}$$

$$= \left[\theta_1^2{}^T \varepsilon(q, \ddot{q}_r), \dots, \theta_m^2{}^T \varepsilon(q, \ddot{q}_r) \right]^T$$

$$F^3(q, \ddot{q}) = \Theta^3{}^T \varepsilon(q, \ddot{q}) \tag{53}$$

$$= \left[\theta_1^{3T} \varepsilon(q, \dot{q}), \dots, \theta_m^{3T} \varepsilon(q, \ddot{q}) \right]^T$$

The control security input is given by

$$\tau = M \ddot{q}_r + B(q)\dot{q}\dot{q} + C(q)\dot{q}^2 + g(q) + F^1(q, \dot{q}) + F^2(q, \ddot{q}_r) + F^3(q, \ddot{q}) - K_p e - K_v \dot{e} \quad (54)$$

Where M^{\wedge} , $B(q)\dot{q}\dot{q}$, $C(q)\dot{q}^2$, $g(q)$ are the estimations of $M(q)$.

Based on sliding mode formulation (28) and PD linear methodology (5);

$$S_{New} = (\dot{e} + \lambda e) \quad (55)$$

And U_{switch} is obtained by

$$U_{switch} = K(\vec{x}, t) \cdot \text{sgn}(S_{New}) = K(\vec{x}, t) \cdot \text{sgn}(K(\dot{e} + \lambda e)) \quad (56)$$

The Lyapunov function in this design is defined as

$$V = \frac{1}{2} S^T M S + \frac{1}{2} \sum_{j=1}^M \frac{1}{\gamma_{sj}} \phi^T \cdot \phi_j \quad (57)$$

where γ_{sj} is a positive coefficient, $\phi = \theta^* - \theta$, θ^* is minimum error and θ is adjustable parameter. Since $\dot{M} - 2V$ is skew-symmetric matrix;

$$S^T M \dot{S} + \frac{1}{2} S^T \dot{M} S = S^T (M \dot{S} + V S) \quad (58)$$

If the dynamic formulation of robot manipulator defined by

$$\tau = M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) \quad (59)$$

the controller formulation is defined by

$$\tau = \widehat{M}\ddot{q}_r + \widehat{V}\dot{q}_r + \widehat{G} - \lambda S - K \quad (60)$$

According to (58) and (59)

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) = \widehat{M}\ddot{q}_r + \widehat{V}\dot{q}_r + \widehat{G} - \lambda S - K \quad (61)$$

Since $\dot{q}_r = \dot{q} - S$ and $\ddot{q}_r = \ddot{q} - \dot{S}$

$$M \dot{S} + (V + \lambda)S = \Delta f - K \quad (62)$$

$$M\dot{S} = \Delta f - K - VS - \lambda S$$

The derivation of V is defined

$$\begin{aligned} \dot{V} &= S^T M \dot{S} + \frac{1}{2} S^T \dot{M} S + \sum_{j=1}^M \frac{1}{\gamma_{sj}} \phi^T \cdot \dot{\phi}_j \quad (63) \\ \dot{V} &= S^T (M\dot{S} + VS) + \sum_{j=1}^M \frac{1}{\gamma_{sj}} \phi^T \cdot \dot{\phi}_j \end{aligned}$$

Based on (61) and (62)

$$\dot{V} = S^T (\Delta f - K - VS - \lambda S + VS) + \sum_{j=1}^M \frac{1}{\gamma_{sj}} \phi^T \cdot \dot{\phi}_j \quad (64)$$

where $\Delta f = [M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q)] - \sum_{l=1}^M \theta^T \zeta(x)$

$$\begin{aligned} \dot{V} &= \sum_{j=1}^M [S_j (\Delta f_j - K_j)] - S^T \lambda S \\ &\quad + \sum_{j=1}^M \frac{1}{\gamma_{sj}} \phi^T \cdot \dot{\phi}_j \end{aligned}$$

suppose K_j is defined as follows

$$K_j = \frac{\sum_{l=1}^M \theta_j^l [\mu_A(S_j)]}{\sum_{l=1}^M [\mu_A(S_j)]} = \theta_j^T \zeta_j(S_j) \quad (65)$$

Where $\zeta_j(S_j) = [\zeta_j^1(S_j), \zeta_j^2(S_j), \zeta_j^3(S_j), \dots, \zeta_j^M(S_j)]^T$

$$\zeta_j^1(S_j) = \frac{\mu_{(A)_j^1}(S_j)}{\sum_i \mu_{(A)_j^i}(S_j)} \quad (66)$$

where $\mu_{(xi)}$ is membership function.

The fuzzy system is defined as

$$f(x) = \tau_{fuzzy} = \sum_{l=1}^M \theta^T \zeta(x) = \psi(e, \dot{e}) \quad (67)$$

where $\theta = (\theta^1, \theta^2, \theta^3, \dots, \theta^M)$ is adjustable parameter in (65) according to (62), (63) and (65);

$$\dot{V} = \sum_{j=1}^M [S_j (\Delta f_j - \theta^T \zeta(S_j))] - S^T \lambda S + \sum_{j=1}^M \frac{1}{\gamma_{sj}} \phi^T \cdot \dot{\phi}_j \quad (68)$$

Based on $\phi = \theta^* - \theta \rightarrow \dot{\theta} = -\dot{\phi}$

$$\dot{V} = \sum_{j=1}^M [S_j (\Delta f_j - \theta^{*T} \zeta_j(S_j) + \phi^T \zeta_j(S_j))] - S^T \lambda S + \sum_{j=1}^M \frac{1}{\gamma_{sj}} \phi^T \cdot \dot{\phi}_j \quad (69)$$

$$\dot{V} = \sum_{j=1}^M [S_j (\Delta f_j - (\theta^*)^T \zeta_j(S_j))] - S^T \lambda S + \sum_{j=1}^M \frac{1}{\gamma_{sj}} \phi_j^T [\gamma_{sj} \cdot S_j \cdot \zeta_j(S_j) + \dot{\phi}_j]$$

where $\dot{\theta}_j = \gamma_{sj} S_j \zeta_j(S_j)$ is adaption law, $\dot{\phi}_j = -\dot{\theta}_j = -\gamma_{sj} S_j \zeta_j(S_j)$
 \dot{V} is considered by

$$\dot{V} = \sum_{j=1}^m [S_j \Delta f_j - ((\theta_j^*)^T \zeta_j(S_j))] - S^T \lambda S \quad (70)$$

The minimum error is defined by

$$e_{mj} = \Delta f_j - ((\theta_j^*)^T \zeta_j(S_j)) \quad (71)$$

Therefore \dot{V} is computed as

$$\dot{V} = \sum_{j=1}^m [S_j e_{mj}] - S^T \lambda S \quad (72)$$

$$\begin{aligned} &\leq \sum_{j=1}^m |S_j| |e_{mj}| - S^T \lambda S \\ &= \sum_{j=1}^m |S_j| |e_{mj}| - \lambda_j S_j^2 \\ &= \sum_{j=1}^m |S_j| (|e_{mj}| - \lambda_j S_j) \end{aligned} \quad (73)$$

For continuous function $g(x)$, and suppose $\epsilon > 0$ it is defined the fuzzy logic system in form of

$$\text{Sup}_{x \in U} |f(x) - g(x)| < \epsilon \quad (74)$$

the minimum approximation error (e_{mj}) is very small.

$$\text{if } \lambda_j = \alpha \quad \text{that } \alpha |S_j| > e_{mj} \quad (S_j \neq 0) \quad \text{then } \dot{V} < 0 \text{ for } (S_j \neq 0) \quad (75)$$

This method has two main controller's coefficients, K_p and K_v . To tune and optimize these parameters mathematical formulation is used

$$U = U_{fuzzy} + U_{sliding} + U_{PD} \quad (76)$$

$$U = U_{fuzzy} + U_{switch} = \quad (77)$$

$$\begin{aligned}
 & \left[\mathbf{M}^{-1}(\mathbf{B} + \mathbf{C} + \mathbf{G}) + \dot{\mathbf{S}} \right] \mathbf{M} + \mathbf{K} \cdot \text{sgn}(\mathbf{S}) + \frac{\sum_{l=1}^M \theta^l \left[\prod_{i=1}^n \exp\left(-\left(\frac{x_i - \alpha_i^l}{\delta_i^l}\right)^2\right) \right]}{\sum_{l=1}^M \left[\prod_{i=1}^n \exp\left(-\left(\frac{x_i - \alpha_i^l}{\delta_i^l}\right)^2\right) \right]} + \mathbf{K}_{p_a} \mathbf{e}_1 + \\
 & \mathbf{K}_{v_a} \dot{\mathbf{e}}_1 + \mathbf{K}_{I_a} \sum \mathbf{e}_1
 \end{aligned}$$

The most important different between PD+SMC and PD+SMC+FL is the uncertainty. In PD+SMC the uncertainty is $d = G + F + f$. The sliding mode gain must be bigger than its upper bound. It is not an easy job because this term includes tracking errors e_1 and \dot{q}_1 . While in PD+SMC+FL, the uncertainty η is the fuzzy approximation error for $G + F + f$.

$$G + F + f = \frac{\sum_{l=1}^M \theta^l \left[\prod_{i=1}^n \exp\left(-\left(\frac{x_i - \alpha_i^l}{\delta_i^l}\right)^2\right) \right]}{\sum_{l=1}^M \left[\prod_{i=1}^n \exp\left(-\left(\frac{x_i - \alpha_i^l}{\delta_i^l}\right)^2\right) \right]} \tag{78}$$

It is usually is smaller than $G + F + f$; and the upper bound of it is easy to be estimated.

Fuzzy controller generally uses diagonal type rule base which is called as PI type rule base. Next we discuss formation of rule base. $N \times N$ is size of rule base for N membership functions. For generation of rule base we will define a term M such that it is given by

$$\begin{bmatrix}
 R(-M, -M) & R(-M, -M + 1) & \dots & R(-M, M - 1) & R(-M, M) \\
 R(-M + 1, -M) & \dots & \dots & \dots & R(-M + 1, M) \\
 \dots & \dots & \dots & \dots & \dots \\
 R(M - 1, -M) & \dots & \dots & \dots & R(M - 1, M) \\
 R(M, -M) & R(M, -M + 1) & \dots & R(M, M - 1) & R(M, M)
 \end{bmatrix}$$

where i is index for error value and j is index for change in error value. So for one value of error change in error value varies over range $(-M, M)$. For $N = 9$ a rule base matrix has the form. Thus error and change in error value varies over interval $(-4, 4)$. Using the above matrix, we have

$$\begin{bmatrix}
 R(-4, -4) & R(-4, -3) & R(-4, -2) & R(-4, -1) & R(-4, 0) & R(-4, 1) & R(-4, 2) & R(-4, 3) & R(-4, 4) \\
 R(-3, -4) & R(-3, -3) & R(-3, -2) & R(-3, -1) & R(-3, 0) & R(-3, 1) & R(-3, 2) & R(-3, 3) & R(-3, 4) \\
 R(-2, -4) & R(-2, -3) & R(-2, -2) & R(-2, -1) & R(-2, 0) & R(-2, 1) & R(-2, 2) & R(-2, 3) & R(-2, 4) \\
 R(-1, -4) & R(-1, -3) & R(-1, -2) & R(-1, -1) & R(-1, 0) & R(-1, 1) & R(-1, 2) & R(-1, 3) & R(-1, 4) \\
 R(0, -4) & R(0, -3) & R(0, -2) & R(0, -1) & R(0, 0) & R(0, 1) & R(0, 2) & R(0, 3) & R(0, 4) \\
 R(1, -4) & R(1, -3) & R(1, -2) & R(1, -1) & R(1, 0) & R(1, 1) & R(1, 2) & R(1, 3) & R(1, 4) \\
 R(2, -4) & R(2, -3) & R(2, -2) & R(2, -1) & R(2, 0) & R(2, 1) & R(2, 2) & R(2, 3) & R(2, 4) \\
 R(3, -4) & R(3, -3) & R(3, -2) & R(3, -1) & R(3, 0) & R(3, 1) & R(3, 2) & R(3, 3) & R(3, 4) \\
 R(4, -4) & R(4, -3) & R(4, -2) & R(4, -1) & R(4, 0) & R(4, 1) & R(4, 2) & R(4, 3) & R(4, 4)
 \end{bmatrix}$$

which has total 81 rules. Solving the $R(i, j)$ using we get

$$\begin{bmatrix} -4 & -4 & -4 & -4 & -4 & -3 & -2 & -1 & 0 \\ -4 & -4 & -4 & -4 & -3 & -2 & -1 & 0 & 1 \\ -4 & -4 & -4 & -3 & -2 & -1 & 0 & 1 & 2 \\ -4 & -4 & -3 & -2 & -1 & 0 & 1 & 2 & 3 \\ -4 & -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 \\ -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 & 4 \\ -2 & -1 & 0 & 1 & 2 & 3 & 4 & 4 & 4 \\ -1 & 0 & 1 & 2 & 3 & 4 & 4 & 4 & 4 \\ 0 & 1 & 2 & 3 & 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

4. Results and Discussion

In this section, we use a benchmark model, PUMA-560 robot manipulator, to evaluate our control algorithms. We compare the following controllers: classical PD controller, PD fuzzy controller and serial fuzzy sliding mode PD controller which is proposed in this paper. The simulation was implemented in MATLAB/SIMULINK environment.

Close loop response of tracking result without any disturbance: Figure 2 illustrates the tracking performance in three types of controller; linear PD controller, linear PD controller based on fuzzy logic estimator and nonlinear estimator based on fuzzy logic and sliding mode controller.

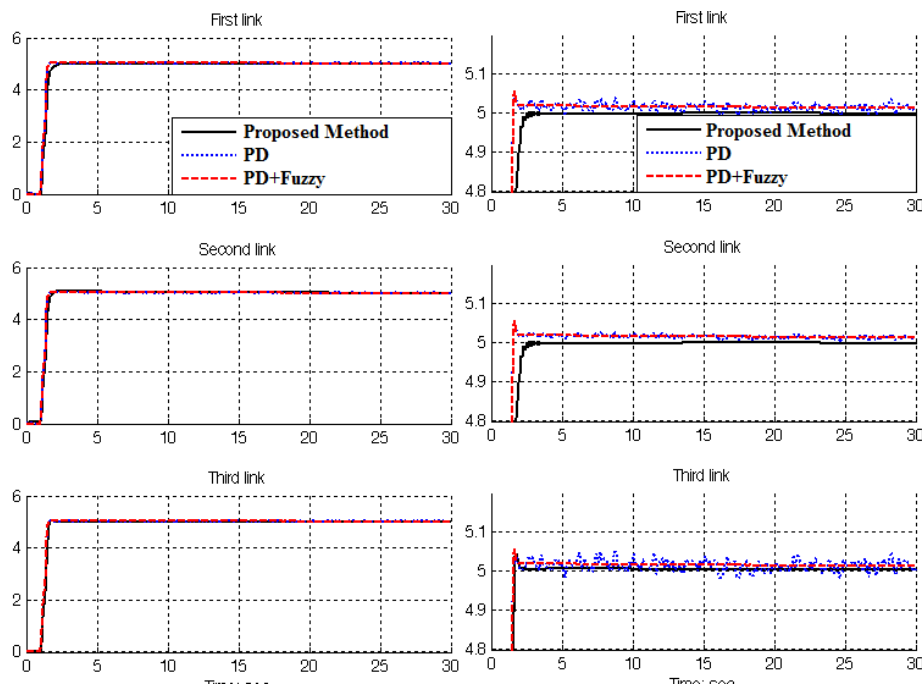


Figure 2. Linear PD, PD+FLC and Proposed Method Trajectory Following Without Disturbance

Based on Figure 2; pure PD controller has oscillation in first and three links, because robot manipulator is a highly nonlinear controller and control of this system by linear method is very difficult. Based on above graph, however PD+FUZZY controller is a nonlinear methodology but it has difficulty to control this plant because it is a model base controller.

Close loop response of trajectory following in presence of load disturbance: Figure 3 demonstrates the power disturbance elimination in three types of controller in presence of disturbance for robot manipulator. The disturbance rejection is used to test the robustness comparisons of these three methodologies.

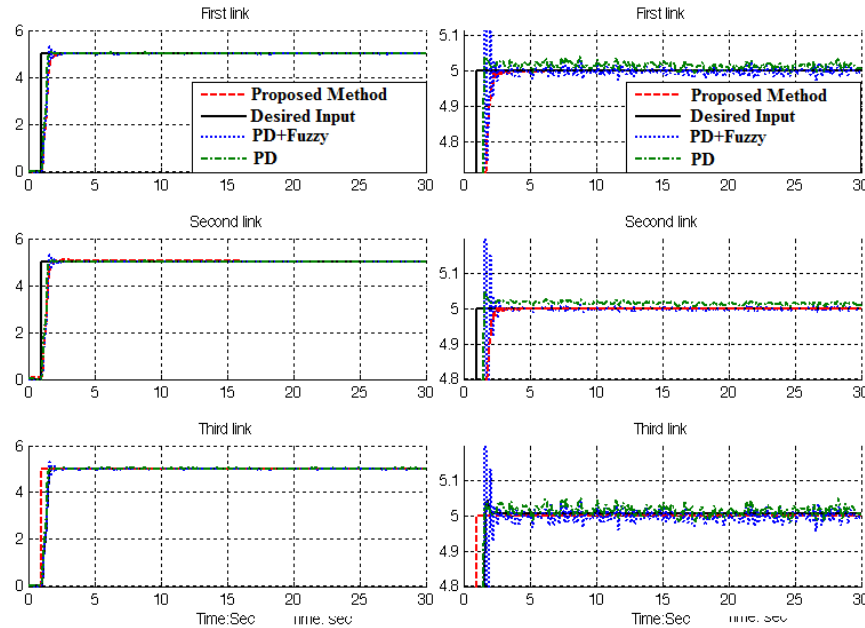


Figure 3. Linear PD, PD+FLC and Proposed Method Trajectory Following with Disturbance

Based on Figure 3; by comparison with the PD and PD+FLC, proposed serial compensator PD+Fuzzy+SMC is more stable and robust and our method doesn't have any chattering and oscillation.

5. Conclusion

In the present work a mathematical relationship has been developed to utilize the shifting of elements in a typical fuzzy rule base based on factor obtained from nonlinearity parameters of the second order process. The rule bases obtained after the shifting of elements are also given in table form and are tested on different combinations of the robot and sliding mode control models. The simulation results obtained are using the modified rule bases given improvement than the typical fuzzy controller. Also the results obtained for large changes in the time constant value and delay time values using modified rule base controller is better than conventional fuzzy controller. The limitation of the method is that it is not effective for rule base of smaller size such as (3×3) and (5×5) . But it is common practice to use minimum (7×7) rule base for fuzzy controller. The main contributions of the paper are twofold. The structure of fuzzy PD control with sliding mode compensation is new. We propose parallel structure: parallel compensation. The key technique is dead-zone, such that fuzzy control and sliding mode control can be switched automatically. The stability analysis of fuzzy sliding mode PD control is also new. Stability analysis of fuzzy PD control with first-order or second-order sliding mode is not published in the literature. The benefits of the proposed method; the chattering effects of fuzzy sliding mode PD control, the slow convergence of the fuzzy PD and the chattering problem of sliding mode PD control are avoided effectively.

References

- [1] T. R. Kurfess, Robotics And Automation Handbook: Crc, 2005.
- [2] J. J. E. Slotine And W. Li, Applied Nonlinear Control Vol. 461: Prentice Hall Englewood Cliffs, Nj, 1991.
- [3] K. Ogata, Modern Control Engineering: Prentice Hall, 2009.
- [4] J. J. D'azzo, C. H. Houpis And S. N. Sheldon, Linear Control System Analysis And Design With Matlab: Crc, 2003.
- [5] B. Siciliano And O. Khatib, Springer Handbook Of Robotics: Springer-Verlag New York Inc, 2008.
- [6] F. T. Cheng, T. L. Hour, Y. Y. Sun And T. H. Chen, "Study And Resolution Of Singularities For A 6-Dof Puma Manipulator," Systems, Man, And Cybernetics, Part B: Cybernetics, Ieee Transactions On, No. 2, Vol. 27, Pp. 332-343, 2002.
- [7] M. W. Spong And M. Vidyasagar, Robot Dynamics And Control: Wiley-India, 2009.
- [8] A. Vivas And V. Mosquera, "Predictive Functional Control Of A Puma Robot," Conference Proceedings, 2005.
- [9] D. Nguyen-Tuong, M. Seeger And J. Peters, "Computed Torque Control With Nonparametric Regression Models," Ieee Conference Proceeding, 2008, Pp. 212-217.
- [10] Farzin Piltan , N. Sulaiman, Zahra Tajpaykar, Payman Ferdosali, Mehdi Rashidi, "Design Artificial Nonlinear Robust Controller Based On Ctlc And Fsmc With Tunable Gain," International Journal Of Robotic And Automation, 2 (3): 205-220, 2011.
- [11] Farzin Piltan, A. R. Salehi And Nasri B Sulaiman., "Design Artificial Robust Control Of Second Order System Based On Adaptive Fuzzy Gain Scheduling," World Applied Science Journal (Wasj), 13 (5): 1085-1092, 2011.
- [12] Farzin Piltan, N. Sulaiman, Atefeh Gavahian, Samira Soltani, Samaneh Roosta, "Design Mathematical Tunable Gain Pid-Like Sliding Mode Fuzzy Controller With Minimum Rule Base," International Journal Of Robotic And Automation, 2 (3): 146-156, 2011.
- [13] B. K. Yoo And W. C. Ham, "Adaptive Control Of Robot Manipulator Using Fuzzy Compensator," Fuzzy Systems, Ieee Transactions On, No. 2, Vol. 8, Pp. 186-199, 2002.
- [14] Y. S. Kung, C. S. Chen And G. S. Shu, "Design And Implementation Of A Servo System For Robotic Manipulator," Cacs, 2005.

Authors



Mina Mirzaie is a computer researcher of research and development company SSP. Co. She is an expert artificial intelligence and computer engineer in this company. Her research activities deal with the robotic control, artificial intelligence and expert system.



Farzin Piltan was born on 1975, Shiraz, Iran. In 2004 he is joined Institute of Advance Science and Technology, Research and Development Center, IRAN SSP. Now he is a dean of Intelligent Control and Robotics Lab. In addition to 7 textbooks, Farzin Piltan is the main author of more than 100 scientific papers in refereed journals. He is editorial review board member for 'international journal of control and automation (IJCA), Australia, ISSN: 2005-4297; 'International Journal of Intelligent System and Applications (IJISA)', Hong Kong, ISSN: 2074-9058; 'IAES international journal of robotics and automation, Malaysia, ISSN:2089-4856; 'International Journal of Reconfigurable and Embedded Systems', Malaysia, ISSN:2089-4864. His current research interests are nonlinear control, artificial control system and applied to FPGA, robotics and artificial nonlinear control and IC engine modeling and control.



Dr. Nasri Sulaiman, received The B. Eng From University Of Putra Malaysia In 1994, M.Sc., From University Of Southampton, Uk In1999, And Phd Degrees From University Of Edinburgh, Uk In 2007, Respectively. He has more than 65 journal and conference papers. He Is Currently A Senior Lecturer In The Department Of Electrical Engineering At University Putra Malaysia Of The Program For Signal Processing, And Evolvable Harware (EHW) And Also Is Head Of Control And Automation Laboratory, Iranian Institute Of Advanced Science And Technology, Shiraz, Iran. He Is An Advisor In Spherical Motor Control.



Shahnaz Tayebi Haghghi, is an electronic researcher of research and development company SSP. Co. In 2004 she is jointed the research and development company, SSP Co, Shiraz, Iran. In addition to 3 textbooks, she is the main author of more than 5 scientific papers in refereed journals. Her main areas of research interests are nonlinear control, artificial control system, robotics and artificial nonlinear control and IC engine modelling and control.

