Novel Robot Manipulator Adaptive Artificial Control: Design a Novel SISO Adaptive Fuzzy Sliding Algorithm Inverse Dynamic Like Method

Farzin Piltan *SSP.ROBOTIC@yahoo.com*
Department of Electrical and Electronic Engineering,
Faculty of Engineering, Universiti Putra Malaysia
43400 Serdang, Selangor, Malaysia

N. Sulaiman *nasri@eng.upm.edu.my*
Department of Electrical and Electronic Engineering,
Faculty of Engineering, Universiti Putra Malaysia
43400 Serdang, Selangor, Malaysia

Hajar Nasiri *SSP.ROBOTIC@yahoo.com*
Industrial Electrical and Electronic Engineering
SanatkadeheSabze Pasargad. CO (S.S.P. Co),
NO:16 , PO.Code 71347-66773, Fourth floor
Dena Apr , Seven Tir Ave , Shiraz , Iran

Sadeq Allahdadi *SSP.ROBOTIC@yahoo.com*
Industrial Electrical and Electronic Engineering
SanatkadeheSabze Pasargad. CO (S.S.P. Co),
NO:16 , PO.Code 71347-66773, Fourth floor
Dena Apr , Seven Tir Ave , Shiraz , Iran

Mohammad A. Bairami *SSP.ROBOTIC@yahoo.com*
Industrial Electrical and Electronic Engineering
SanatkadeheSabze Pasargad. CO (S.S.P. Co),
NO:16 , PO.Code 71347-66773, Fourth floor
Dena Apr , Seven Tir Ave , Shiraz , Iran

Abstract

Refer to the research, design a novel SISO adaptive fuzzy sliding algorithm inverse dynamic like method (NAIDLC) and application to robot manipulator has proposed in order to design high performance nonlinear controller in the presence of uncertainties. Regarding to the positive points in inverse dynamic controller, fuzzy logic controller and self tuning fuzzy sliding method, the output has improved. The main objective in this research is analyses and design of the adaptive robust controller based on artificial intelligence and nonlinear control. Robot manipulator is nonlinear, time variant and a number of parameters are uncertain, so design the best controller for this plant is the main target. Although inverse dynamic controller have acceptable performance with known dynamic parameters but regarding to uncertainty, this controller's output has fairly fluctuations. In order to solve this problem this research is focoused on two methodology the first one is design a fuzzy inference system as a estimate nonlinear part of main controller but this method caused to high computation load in fuzzy rule base and the second method is focused on design novel adaptive method to reduce the computation in fuzzy algorithm.

Keywords: Inverse Dynamic Control, Sliding Mode Algorithm, Fuzzy Estimator Sliding Mode Control, Adaptive Method, Adaptive Fuzzy Sliding Mode Inverse Dynamic like Method, Fuzzy Inference System, Robot Manipulator

1. INTRODUCTION

Robot manipulator is collection of links that connect to each other by joints, these joints can be revolute and prismatic that revolute joint has rotary motion around an axis and prismatic joint has linear motion around an axis. Each joint provides one or more degrees of freedom (DOF). From the mechanical point of view, robot manipulator is divided into two main groups, which called; serial robot links and parallel robot links. In serial robot manipulator, links and joints is serially connected between base and final frame (end-effector). Parallel robot manipulators have many legs with some links and joints, where in these robot manipulators base frame has connected to the final frame. Most of industrial robots are serial links, which in serial robot manipulator the axis of the first three joints has a known as major axis, these axes show the position of end-effector, the axis number four to six are the minor axes that use to calculate the orientation of end-effector, at last the axis number seven to n use to avoid the bad situation. Dynamic modeling of robot manipulators is used to describe the behavior of robot manipulator, design of model based controller, 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 widely used in industry and academic. It has a nonlinear and uncertain dynamic parameters serial link 6 degrees of freedom (DOF) robot manipulator. A non linear robust controller design is major subject in this work [1-3].

In modern usage, the word of control has many meanings, this word is usually taken to mean regulate, direct or command. The word feedback plays a vital role in the advance engineering and science. The conceptual frame work in Feed-back theory has developed only since world war II. In the twentieth century, there was a rapid growth in the application of feedback controllers in process industries. According to Ogata, to do the first significant work in three-term or PID controllers which Nicholas Minorsky worked on it by automatic controllers in 1922. In 1934, Stefen Black was invention of the feedback amplifiers to develop the negative feedback amplifier[2]. Negative feedback invited communications engineer Harold Black in 1928 and it occurs when the output is subtracted from the input. Automatic control has played an important role in advance science and engineering and its extreme importance in many industrial applications, i.e., aerospace, mechanical engineering and robotic systems. The first significant work in automatic control was James Watt's centrifugal governor for the speed control in motor engine in eighteenth century[2]. There are several methods for controlling a robot manipulator, which all of them follow two common goals, namely, hardware/software implementation and acceptable performance. However, the mechanical design of robot manipulator is very important to select the best controller but in general two types schemes can be presented, namely, a joint space control schemes and an operation space control schemes[1]. Joint space and operational space control are closed loop controllers which they have been used to provide robustness and rejection of disturbance effect. The main target in joint space controller is to design a feedback controller which the actual motion ($q_a(t)$) and desired motion ($q_d(t)$) as closely as possible. This control problem is classified into two main groups. Firstly, transformation the desired motion $X_d(t)$ to joint variable $q_d(t)$ by inverse kinematics of robot manipulators[6]. This control include simple PD control, PID control, inverse dynamic control, Lyapunov-based control, and passivity based control that explained them in the following section. The main target in operational space controller is to design a feedback controller to allow the actual end-effector motion $X_a(t)$ to track the desired endeffector motion $X_d(t)$. This control methodology requires a greater algorithmic complexity and the inverse kinematics used in the feedback control loop. Direct measurement of operational space variables are very expensive that caused to limitation used of this controller in industrial robot manipulators[4-8]. One of the simplest ways to analysis control of multiple DOF robot manipulators are analyzed each joint separately such as SISO systems and design an independent joint controller for each joint. In this controller, inputs only depends on the velocity and displacement of the corresponding joint and the other parameters between joints such as coupling presented by disturbance input. Joint space controller has many advantages such as one type controllers design for all joints with the same formulation, low cost hardware, and simple structure. Nonlinear control provides a methodology of nonlinear methodologies for nonlinear uncertain systems (e.g., robot manipulators) to have an acceptable performance. These controllers divided into seven groups, namely, inverse dynamic control, computed-torque control, passivity-based control, sliding mode control (variable structure control), artificial intelligence control, lyapunov-based control and adaptive control[9-14]. Inverse dynamic controller (IDC) is a powerful nonlinear controller which it widely used in control robot manipulator. It is based on Feed-back linearization and computes the required arm torques using the nonlinear feedback control law. This controller works very well when all dynamic and physical parameters are known but when the robot manipulator has variation in dynamic parameters, the controller has no acceptable performance[14]. In practice, most of physical systems (e.g., robot manipulators) parameters are unknown or time variant, therefore, inverse dynamic like controller used to compensate dynamic equation of robot manipulator[1, 6]. Research on inverse dynamic controller is significantly growing on robot manipulator application which has been reported in [1, 6, 9, 11, 63-65]. Vivas and Mosquera [63]have proposed a predictive functional controller and compare to inverse dynamic controller for tracking response in uncertain environment. However both controllers have been used in Feed-back linearization, but predictive strategy gives better result as a performance. An inverse dynamic control with non parametric regression models have been presented for a robot arm[64]. This controller also has been problem in uncertain dynamic models. Based on [1, 6]and [63-65] inverse dynamic controller is a significant nonlinear controller to certain systems which it is based on feedback linearization and computes the required arm torques using the nonlinear feedback control law. When all dynamic and physical parameters are known the controller works fantastically; practically a large amount of systems have uncertainties and complicated by artificial intelligence or applied on line tuning in this controller decrease this kind of challenge.

Zadeh [31] introduced fuzzy sets in 1965. After 40 years, fuzzy systems have been widely used in different fields, especially on control problems. Fuzzy systems transfer expert knowledge to mathematical models. Fuzzy systems used fuzzy logic to estimate dynamics of proposed systems. Fuzzy controllers including fuzzy if-then rules are used to control proposed systems. Conventional control methods use mathematical models to controls systems [31-40]. Fuzzy control methods replace the mathematical models with fuzzy if then-rules and fuzzy membership function to controls systems. Both fuzzy and conventional control methods are designed to meet system requirements of stability and convergence. When mathematical models are unknown or partially unknown, fuzzy control models can used fuzzy systems to estimate the unknown models. This is called the model-free approach [31-40]. Conventional control models can use adaptive control methods to achieve the model-free approach. When system dynamics become more complex, nonlinear systems are difficult to handle by conventional control methods. From the universal approximation theorem, fuzzy systems can approximate arbitrary nonlinear systems. In practical problems, systems can be controlled perfectly by expert. Experts provide linguistic description about systems. Conventional control methods cannot design controllers combined with linguistic information. When linguistic information is important for designing controllers, we need to design fuzzy controllers for our systems. Fuzzy control methods are easy to understand for designers. The design process of fuzzy controllers can be simplified with simple mathematical models. Research on applied fuzzy logic methodology in inverse dynamic controller (FIDLC) to compensate the unknown system dynamics considerably improves the robot manipulator control process [15-30, 41-47].

Adaptive control uses a learning method to self-learn the parameters of systems. For system whose dynamics are varying, adaptive control can learn the parameters of system dynamics. In traditional adaptive control, we need some information about our system such as the structure of system or the order of the system. In adaptive fuzzy control we can deal with uncertain systems. Due to the linguistic characteristic, adaptive fuzzy controllers behave like operators: adaptively controlling the system under various conditions. Adaptive fuzzy control provides a good tool for making use of expert knowledge to adjust systems. This is important for a complex unknown system with changing dynamics. We divide adaptive fuzzy control into two categories: direct adaptive fuzzy control and indirect adaptive fuzzy control. A direct adaptive fuzzy controller adjusts the parameters of the control input. An indirect adaptive fuzzy controller adjusts the parameters of the control system based on the estimated dynamics of the plant. This research is used fuzzy indirect method to estimate the nonlinear equivalent part in order to used sliding mode fuzzy algorithm to tune and adjust the sliding function (direct adaptive). Research on applied fuzzy logic methodology in sliding mode controller (FSMC) to reduce or eliminate the high frequency oscillation (chattering), to compensate the unknown system dynamics and also to adjust the linear sliding surface slope in pure sliding mode controller considerably improves the robot manipulator control process [41-62]. H.Temeltas [46] has proposed fuzzy adaption techniques for SMC to achieve robust tracking of nonlinear systems and solves the chattering problem. Conversely system's performance is better than sliding mode controller; it is depended on nonlinear dynamic equqation. C. L. Hwang *et al.* [47]have proposed a Tagaki-Sugeno (TS) fuzzy model based sliding mode control based on *N* fuzzy based linear state-space to estimate the uncertainties. A multi-input multi-output FSMC reduces the chattering phenomenon and reconstructs the approximate the unknown system has been presented for a robot manipulator [42].

In this research we will highlight the SISO adaptive fuzzy sliding algorithm to on line tuning inverse dynamic like controller with estimates the nonlinear dynamic part derived in the Lyapunov sense. This algorithm will be analyzed and evaluated on robotic manipulators. Section 2, serves as an introduction to the classical inverse dynamic control algorithm and its application to a 3 degree of-freedom robot manipulator, introduced sliding mode controller to design adaptive part, describe the objectives and problem statements. Part 3, introduces and describes the methodology algorithms and proves Lyapunov stability. Section 4 presents the simulation results of this algorithm applied to a 2 degree-of-freedom robot manipulator and the final section is describe the conclusion.

2. OBJECTIVES, PROBLEM STATEMENTS, INVERSE DYNAMIC METHODOLOGY AND SLIDING MODE ALGORITM

When system works with various parameters and hard nonlinearities design linear controller technique is very useful in order to be implemented easily but it has some limitations such as working near the system operating point[2-20]. Inverse dynamic controller is used in wide range areas such as in robotics, in control process, in aerospace applications and in power converters because it has an acceptable control performance and solve some main challenging topics in control such as resistivity to the external disturbance. Even though, this controller is used in wide range areas but, classical inverse dynamic controller has nonlinear part disadvantage which this challenge must be estimated by fuzzy method [20]. Conversely pure FLC works in many areas, it cannot guarantee the basic requirement of stability and acceptable performance[31-40]. Although both inverse dynamic controller and FLC have been applied successfully in many applications but they also have some limitations. Fuzzy estimator is used instead of dynamic uncertaint equation to implement easily and avoid mathematical model base controller. To reduce the effect of uncertainty in proposed method, adaptive fuzzy sliding mode method is applied in inverse dynamic like controller in robot manipulator in order to solve above limitation.

Robot Manipulator Formulation

The equation of a multi degrees of freedom (DOF) robot manipulator is calculated by the following equation[6]:

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

Where τ is $n \times 1$ vector of actuation torque, M (q) is $n \times n$ symmetric and positive define inertia matrix, $N(q,\dot{q})$ is the vector of nonlinearity term, and q is $n \times 1$ position vector. In equation 1 if vector of nonlinearity term derive as Centrifugal, Coriolis and Gravity terms, as a result robot manipulator dynamic equation can also be written as [9-14]:

$$N(q,\dot{q}) = V(q,\dot{q}) + G(q) \quad (2)$$

$$V(q,\dot{q}) = B(q)[\dot{q}\,\dot{q}] + C(q)[\dot{q}]^2 \quad (3)$$

$$\tau = M(q)\ddot{q} + B(q)[\dot{q}\,\dot{q}] + C(q)[\dot{q}]^2 + G(q) \quad (4)$$

Where,

$B(q)$ is matrix of coriolis torques, $C(q)$ is matrix of centrifugal torque, $[\dot{q}\,\dot{q}]$ is vector of joint velocity that it can give by: $[\dot{q}_1.\dot{q}_2,\dot{q}_1.\dot{q}_3,\dots,\dot{q}_1.\dot{q}_n,\dot{q}_2.\dot{q}_3,\dots]^T$, and $[\dot{q}]^2$ is vector, that it can given by: $[\dot{q}_1^2,\dot{q}_2^2,\dot{q}_3^2,\dots]^T$. In robot manipulator dynamic part the inputs are torques and the outputs are actual displacements, as a result in (4) it can be written as [1, 6, 80-81];

$$\ddot{q} = M^{-1}(q).\{\tau - N(q,\dot{q})\} \quad (5)$$

To implementation (5) the first step is implement the kinetic energy matrix (M) parameters by used of Lagrange's formulation. The second step is implementing the Coriolis and Centrifugal matrix which they can calculate by partial derivatives of kinetic energy. The last step to implement the dynamic equation of robot manipulator is to find the gravity vector by performing the summation of Lagrange's formulation.

The kinetic energy equation (M) is a $n \times n$ symmetric matrix that can be calculated by the following equation;

$$M(\theta) = m_1 J_{v1}^T J_{v1} + J_{\omega 1}^{TC1} I_1 J_{\omega 1} + m_2 J_{v2}^T J_{v2} + J_{\omega 2}^{TC2} I_2 J_{\omega 2} + m_3 J_{v3}^T J_{v3} + J_{\omega 3}^{TC3} I_3 J_{\omega 3} + m_4 J_{v4}^T J_{v4} \\ m_5 J_{v5}^T J_{v5} + J_{\omega 5}^{TC5} I_5 J_{\omega 5} + m_6 J_{v6}^T J_{v6} + J_{\omega 6}^{TC6} I_6 J_{\omega 6} \quad (6)$$

As mentioned above the kinetic energy matrix in n DOF is a $n \times n$ matrix that can be calculated by the following matrix [1, 6]

$$M(q) = \begin{bmatrix} M_{11} & M_{12} & \dots & \dots & \dots & M_{1n} \\ M_{21} & \dots & \dots & \dots & \dots & M_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ M_{n.1} & \dots & \dots & \dots & \dots & M_{n.n} \end{bmatrix} \quad (7)$$

The Coriolis matrix (B) is a $n \times \frac{n(n-1)}{2}$ matrix which calculated as follows;

$$B(q) = \begin{bmatrix} b_{112} & b_{113} & \dots & b_{11n} & b_{123} & \dots & b_{12n} & \dots & \dots & b_{1n-1.n} \\ b_{212} & \dots & \dots & b_{21n} & b_{223} & \dots & \dots & \dots & \dots & b_{2n-1.n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ b_{n.1.2} & \dots & \dots & b_{n.1.n} & \dots & \dots & \dots & \dots & \dots & b_{n.n-1.n} \end{bmatrix} \quad (8)$$

and the Centrifugal matrix (C) is a $n \times n$ matrix;

$$C(q) = \begin{bmatrix} C_{11} & \cdots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{n1} & \cdots & C_{nn} \end{bmatrix} \quad (9)$$

And last the Gravity vector (G) is a $n \times 1$ vector;

$$G(q) = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix} \quad (10)$$

Inverse Dynamic Control Formulation

Inverse dynamics control is based on cancelling decoupling and nonlinear terms of dynamics of each link. Inverse dynamics control has the form:

$$\tau = M(q).V + B(q)[\dot{q}\dot{q}] + C(q)\,[\dot{q}]^2 + G(q) \quad (11)$$

where typical choices for V are:

$$V = \ddot{q}_d + K_v(\dot{q}_d - \dot{q}_a) + K_p(q_d - q_a) \quad (12)$$

or with an integral term

$$v = \ddot{q}_d + K_v(\dot{q}_d - \dot{q}_a) + K_p(q_d - q_a) + K_I \int (q_d - q_a)dt \quad (13)$$

where $e = (q_d - q_a)$, the resulting error dynamics is [9, 11, 63-65]

$$\dddot{q}_d + K_v \ddot{e} + K_p \dot{e} + K_I e = 0 \quad (14)$$

where K_p , K_v and K_I are the controller gains. The result schemes is shown in Figure 1, in which two feedback loops, namely, inner loop and outer loop, which an inner loop is a compensate loop and an outer loop is a tracking error loop. However, mostly parameter $N(q,\dot{q})$ is all unknown. So the control cannot be implementation because non linear parameters cannot be determined. In the following section computed torque like controller will be introduced to overcome the problems.

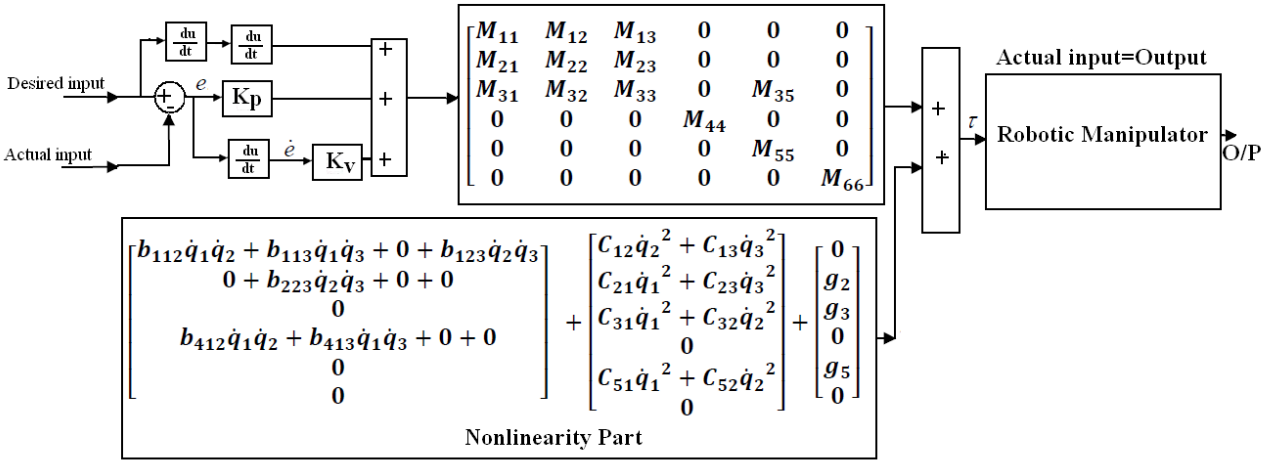

FIGURE 1: Classical inverse dynamic controller: applied to three-link robotic manipulator

The application of proportional-plus-derivative (PD) inverse dynamic controller to control of PUMA robot manipulator introduced in this part.
Suppose that in (13) the nonlinearity term defined by the following term

$$N(q,\dot{q}) = B(q)\dot{q}\dot{q} + C(q)\dot{q}^2 + g(q) = \begin{bmatrix} b_{112}\dot{q}_1\dot{q}_2 + b_{113}\dot{q}_1\dot{q}_3 + 0 + b_{123}\dot{q}_2\dot{q}_3 \\ 0 + b_{223}\dot{q}_2\dot{q}_3 + 0 + 0 \\ 0 \\ b_{412}\dot{q}_1\dot{q}_2 + b_{413}\dot{q}_1\dot{q}_3 + 0 + 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} C_{12}\dot{q}_2^2 + C_{13}\dot{q}_3^2 \\ C_{21}\dot{q}_1^2 + C_{23}\dot{q}_3^2 \\ C_{31}\dot{q}_1^2 + C_{32}\dot{q}_2^2 \\ 0 \\ C_{51}\dot{q}_1^2 + C_{52}\dot{q}_2^2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ g_2 \\ g_3 \\ 0 \\ g_5 \\ 0 \end{bmatrix} \quad (15)$$

Therefore the equation of PD-inverse dynamic controller for control of PUMA robot manipulator is written as the equation of (16);

$$\begin{bmatrix} \hat{\tau}_1 \\ \hat{\tau}_2 \\ \hat{\tau}_3 \\ \hat{\tau}_4 \\ \hat{\tau}_5 \\ \hat{\tau}_6 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} & 0 & 0 & 0 \\ M_{21} & M_{22} & M_{23} & 0 & 0 & 0 \\ M_{31} & M_{32} & M_{33} & 0 & M_{35} & 0 \\ 0 & 0 & 0 & M_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & M_{66} \end{bmatrix} \begin{bmatrix} \ddot{q}_{d1} + K_{v1}\dot{e}_1 + K_{p1}e_1 \\ \ddot{q}_{d2} + K_{v2}\dot{e}_2 + K_{p2}e_2 \\ \ddot{q}_{d3} + K_{v3}\dot{e}_3 + K_{p3}e_3 \\ \ddot{q}_{d4} + K_{v4}\dot{e}_4 + K_{p4}e_4 \\ \ddot{q}_{d5} + K_{v5}\dot{e}_5 + K_{p5}e_5 \\ \ddot{q}_{d6} + K_{v6}\dot{e}_6 + K_{p6}e_6 \end{bmatrix} + \begin{bmatrix} b_{112}\dot{q}_1\dot{q}_2 + b_{113}\dot{q}_1\dot{q}_3 + 0 + b_{123}\dot{q}_2\dot{q}_3 \\ 0 + b_{223}\dot{q}_2\dot{q}_3 + 0 + 0 \\ 0 \\ b_{412}\dot{q}_1\dot{q}_2 + b_{413}\dot{q}_1\dot{q}_3 + 0 + 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} C_{12}\dot{q}_2^2 + C_{13}\dot{q}_3^2 \\ C_{21}\dot{q}_1^2 + C_{23}\dot{q}_3^2 \\ C_{31}\dot{q}_1^2 + C_{32}\dot{q}_2^2 \\ 0 \\ C_{51}\dot{q}_1^2 + C_{52}\dot{q}_2^2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ g_2 \\ g_3 \\ 0 \\ g_5 \\ 0 \end{bmatrix} \quad (16)$$

The controller based on a formulation (16) is related to robot dynamics therefore it has problems in uncertain conditions.

Sliding Mode Control Formulation

We define the tracking error as

$$e = q - q_d \quad (17)$$

Where $q = [q_1, q_2]^T$, $q_d = [q_{1d}, q_{2d}]^T$. The sliding surface is expressed as

$$s = \dot{e} + \lambda e \tag{18}$$

Where $\lambda = diag[\lambda_1, \lambda_2]$, λ_1 and λ_2 are chosen as the bandwidth of the robot controller.
We need to choose τ to satisfy the sufficient condition (9). We define the reference state as

$$\frac{1}{2}\frac{d}{dt}s^2(x,t) = \dot{S} \cdot S = [f - \hat{f} - K\text{sgn}(s)] \cdot S = (f - \hat{f}) \cdot S - K|S| \tag{19}$$

$$\dot{q}_e = \dot{q} - s = \dot{q}_d - \lambda e \tag{20}$$

Now we pick the control input τ as

$$\tau = M^{\wedge} \ddot{q}_r + C_1^{\wedge} \dot{q}_r - As - Ksgn(s) \tag{21}$$

Where $M^{\wedge}$ and $C_1^{\wedge}$ are the estimations of $M(q)$ and $C_1(q, \dot{q})$; $A = diag[a_1, a_2]$ and $K = diag[k_1, k_2]$ are diagonal positive definite matrices. From (17) and (21), we can get

$$M\dot{s} + (C_1 + A)s = \Delta f - Ksgn(s) \tag{22}$$

Where $\Delta f = \Delta M \ddot{q}_r + \Delta C_1 \dot{q}_r$, $\Delta M = M^{\wedge} - M$ and $\Delta C_1 = C_1^{\wedge} - C_1$. We assume that the bound $|\Delta f_i|_{bound}$ of $\Delta f_i (i = 1,2)$ is known. We choose K as

$$K_i \geq |\Delta f_i|_{bound} \tag{23}$$

We pick the Lyapunov function candidate to be

$$V = \frac{1}{2} s^T M s \tag{24}$$

Which is a skew-systemmetric matrix satisfying

$$s^T(\dot{M} - 2C_1)s = 0 \tag{25}$$

Then $\dot{V}$ becomes

$$\begin{aligned} \dot{V} &= s^T M\dot{s} + \frac{1}{2} s^T \dot{M} s \\ &= s^T (M\dot{s} + C_1 s) \\ &= s^T [-As + \Delta f - Ksgn(s)] \\ &= \sum_{i=1}^{2} (s_i[\Delta f_i - K_i sgn(s_i)]) - s^T As \end{aligned} \tag{26}$$

For $K_i \geq |\Delta f_i|$, we always get $s_i[\Delta f_i - K_i sgn(s_i)] \leq 0$. We can describe $\dot{V}$ as

$$\dot{V} = \sum_{i=1}^{2} (s_i[\Delta f_i - K_i sgn(s_i)]) - s^T As \leq -s^T As < 0 \quad (s \neq 0) \tag{27}$$

To attenuate chattering problem, we introduce a saturation function in the control law instead of the sign function in (22). The control law becomes

$$\tau = M^{\wedge} \ddot{q}_r + C_1^{\wedge} \dot{q}_r - As - Ksat(s/\Phi) \tag{28}$$

In this classical sliding mode control method, the model of the robotic manipulator is partly unknown. To attenuate chattering, we use the saturation function described in (20). Our control law changes to

$$\tau = M^{\wedge} \ddot{q}_r + C_1^{\wedge} \dot{q}_r - As - Ksat(s) \tag{29}$$

The main goal is to design a position controller for robot manipulator with acceptable performances (e.g., trajectory performance, torque performance, disturbance rejection, steady state error and RMS error). Robot manipulator has nonlinear dynamic and uncertain parameters consequently; following objectives have been pursuit in this research:

- To develop an inverse dynamic control and applied to robot manipulator.
- To design and implement a position fuzzy estimator inverse dynamic like controller in order to solve the uncertain nonlinear problems in the pure inverse dynamic control.

- To develop a position adaptive fuzzy sliding mode fuzzy estimator inverse dynamic like controller in order to solve the disturbance rejection and reduce the fuzzy load computation.

Figure 2 is shown the classical sliding mode methodology with linear saturation function to eliminate the chattering.

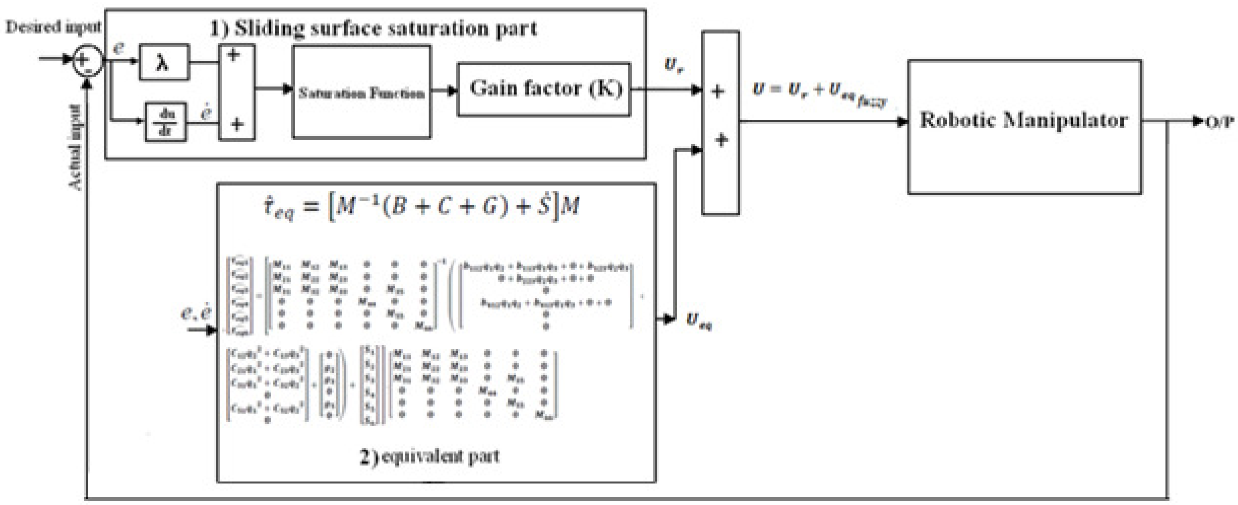

FIGURE 2: Classical sliding mode controller: applied to two-link robotic manipulator

3. METHODOLOGY: DESIGN A NOVEL SISO ADAPTIVE FUZZY SLIDING ALGORITHM INVERSE DYNAMIC LIKE METHOD

First parts are focused on design inverse dynamic like method using fuzzy inference system and estimate or compensate the nonlinear uncertain part. Inverse dynamics control has the form:

$$U = M(q).[\ddot{q}_d + K_v(\dot{q}_d - \dot{q}_a) + K_p(q_d - q_a)] + B(q)[\dot{q}\dot{q}] + C(q)[\dot{q}]^2 + G(q) \quad \textbf{(30)}$$

If nonlinear part is introduced by (31)

$$U_{nonlinear} = B(q)[\dot{q}\dot{q}] + C(q)[\dot{q}]^2 + G(q) \quad \textbf{(31)}$$

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
- Output defuzzification (fuzzy-to-binary[F/B]conversion) [30-40].

The basic structure of a fuzzy controller is shown in Figure 3.

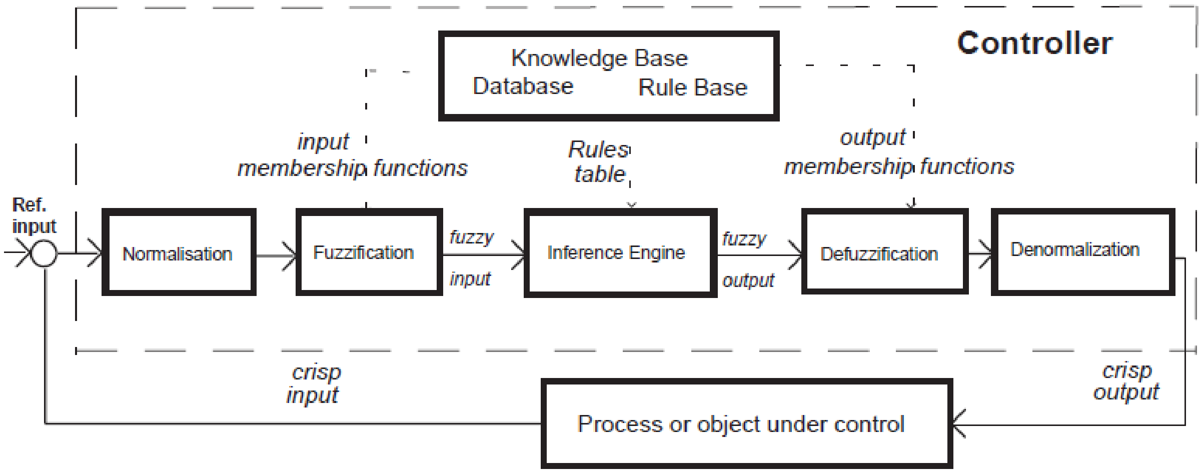

FIGURE 3: Block diagram of a fuzzy controller with details.

The fuzzy system can be defined as below [38-40]

$$f(x) = U_{fuzzy} = \sum_{l=1}^{M} \theta^T \zeta(x) = \psi(S) \quad \textbf{(32)}$$

where $\theta = (\theta^1, \theta^2, \theta^3, \dots\dots, \theta^M)^T, \zeta(x) = (\zeta^1(x), \zeta^2(x), \zeta^3(x), \dots\dots, \zeta^M(x))^T$

$$\zeta^1(x) = \frac{\sum_i \mu_{(xi)x_i}}{\sum_i \mu_{(xi)}} \quad \textbf{(33)}$$

where $\theta = (\theta^1, \theta^2, \theta^3, \dots\dots, \theta^M)$ is adjustable parameter in (8) and $\mu_{(xi)}$ is membership function.

error base fuzzy controller can be defined as

$$U_{fuzzy} = \psi(S) \quad \textbf{(34)}$$

Proposed method is used to a SISO fuzzy system which can approximate the residual coupling effect and alleviate the nonlinear part. The robotic manipulator used in this algorithm is defined as below: the tracking error is defined as:

$$e = q - q_d \quad \textbf{(35)}$$

The control input is given by

$$U_{fuzzy} = \hat{B}[\dot{q}_r \dot{q}_r] + \hat{C}\dot{q}_r + \hat{G} - M(q).[\ddot{q}_d + K_v(\dot{q}_d - \dot{q}_a) + K_p(q_d - q_a)]$$

The fuzzy if-then rules for the jth joint of the robotic manipulator are defined as

$$R^{(l)}: if\ e_j\ is\ A_1^l\ , then\ y\ is\ B_j^l \quad \textbf{(36)}$$

Where $j = 1, \dots, m$ and $l = 1, \dots, M$.

We define K_j by

$$K_j = \frac{\sum_{l=1}^{M} \theta_j^l \left[\mu_{A_j^l}(e_j)\right]}{\sum_{l=1}^{M} \left[\mu_{A_j^l}(e_j)\right]} = \theta_j^T \varepsilon_j(e_j) \quad \textbf{(37)}$$

Where

$$\varepsilon_j(e_j) = \left[\varepsilon_j^1(e_j), \varepsilon_j^2(e_j), \dots, \varepsilon_j^M(e_j)\right]^T, \quad \textbf{(38)}$$

$$\varepsilon_j^l(e_j) = \frac{\sum_{l=1}^{M} \mu_{A_j^l}(e_j)}{\sum_{l=1}^{M} \left[\mu_{A_j^l}(e_j)\right]} \quad \textbf{(39)}$$

The membership function $\mu_{A_j^l}(s_j)$ is a Gaussian membership function defined in bellows:

$$\mu_{A_j^l}(e_j) = \exp\left[-\left(\frac{e_j - \alpha_j^l}{\delta_j^l}\right)^2\right] (j = 1, \dots, m). \quad (40)$$

The fuzzy estimator can be written as follow;

$$U = M(q).[\ddot{q}_d + K_v(\dot{q}_d - \dot{q}_a) + K_p(q_d - q_a)] + B(q)[\dot{q}\dot{q}] + C(q)[\dot{q}]^2 + G(q)$$
$$= \hat{B}[\dot{q}_r\dot{q}_r] + \hat{C}\dot{q}_r + \hat{G} - M(q).[\ddot{q}_d + K_v(\dot{q}_d - \dot{q}_a) + K_p(q_d - q_a)] \quad (41)$$

Since $\dot{q}_r = \dot{q} - e$ and $\ddot{q}_r = \ddot{q} - \dot{e}$ in (41) and (40), we get

$$M\dot{e} + (C_1 + A)s = \Delta F - K \quad (42)$$

Where $\Delta F = \Delta B[\dot{q}_r\dot{q}_r] + \Delta C_1 \dot{q}_r{}^2 + \Delta G$, $\Delta M = M^{\wedge} - M$, $\Delta C_1 = C_1^{\wedge} - C_1$ *and* $G = G^{\wedge} - G$, then $\dot{V}$ becomes

$$\dot{V} = e^T(M\dot{e} + C_1 e) + \sum_{j=1}^{m} \frac{1}{\gamma_{sj}} \phi_j^T \dot{\phi}_j^1$$
$$= -e^T(-Ae + \Delta f - K) + \sum_{j=1}^{m} \frac{1}{\gamma_{sj}} \phi_j^T \dot{\phi}_j^1$$
$$= \sum_{j=1}^{m}[e_j(\Delta f_j - K_j)] - e^T Ae + \sum_{j=1}^{m} \frac{1}{\gamma_{sj}} \phi_j^T \dot{\phi}_j^1$$
$$= \sum_{j=1}^{m}[e_j(\Delta f_j - \theta_j^T \varepsilon_j(e_j))] - e^T Ae + \sum_{j=1}^{m} \frac{1}{\gamma_{sj}} \phi_j^T \dot{\phi}_j^1$$
$$= \sum_{j=1}^{m}(e_j [\Delta f_j - (\theta_j^*)\varepsilon_j(e_j) + \phi_j^T \varepsilon_j(e_j)]) - e^T Ae + \sum_{j=1}^{m} \frac{1}{\gamma_{sj}} \phi_j^T \dot{\phi}_j^1$$
$$= \sum_{j=1}^{m}(e_j [\Delta f_j - (\theta_j^*)\varepsilon_j(e_j)]) - e^T Ae + \sum_{j=1}^{m}\left(\frac{1}{\gamma_{sj}} \phi_j^T [\gamma_{sj} e_j \varepsilon_j(e_j) + \dot{\phi}_j]\right)$$

Based on (3) the formulation of proposed fuzzy sliding mode controller can be written as;

$$U = U_{nonlinear_{fuzzy}} + U_r \quad (43)$$

Where $U_{nonlinear_{fuzzy}} = \hat{B}[\dot{q}_r\dot{q}_r] + \hat{C}\dot{q}_r + \hat{G} - M(q).[\ddot{q}_d + K_v(\dot{q}_d - \dot{q}_a) + K_p(q_d - q_a)] + \sum_{l=1}^{M} \theta^T \zeta(x) + K$

Figure 4 is shown the proposed fuzzy inverse dynamic controller.

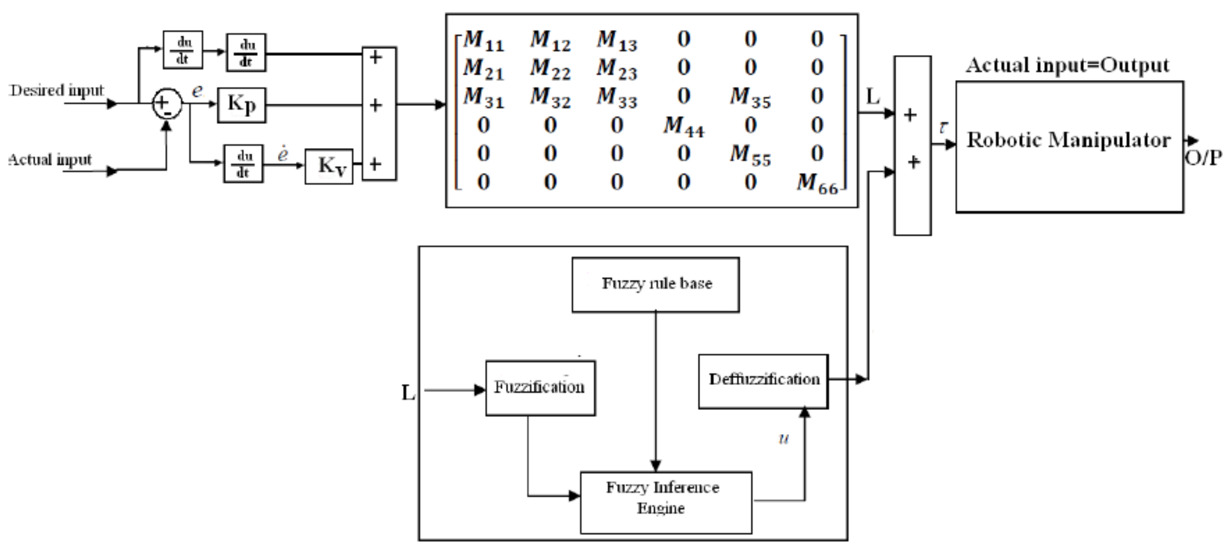

FIGURE 4: Proposed fuzzy estimator inverse dynamic algorithm: applied to robot manipulator

Second part is focuses on design fuzzy sliding mode fuzzy adaptive algorithm, fuzzy algorithm is compensator to estimate nonlinear equivalent part. Adaptive control uses a learning method to self-learn the parameters of systems. For system whose dynamics are varying, adaptive control can learn the parameters of system dynamics. In traditional adaptive control, we need some information about our system such as the structure of system or the order of the system. In adaptive fuzzy control we can deal with uncertain systems. Due to the linguistic characteristic, adaptive fuzzy controllers behave like operators: adaptively controlling the system under various conditions. Adaptive fuzzy control provides a good tool for making use of expert knowledge to adjust systems. This is important for a complex unknown system with changing dynamics. The adaptive fuzzy systems is defined by

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

Where $\theta = (\theta^1, \dots, \theta^M)^T, \mathcal{E}(x) = (\mathcal{E}^1(x), \dots, \mathcal{E}^M(x))^T$, $and\ \mathcal{E}^l(x) =: \prod_{i=1}^{n} \mu_{A_i^l}(x_i) / \sum_{l=1}^{M}(\prod_{i=1}^{n} \mu_{A_i^l}(x_i))$ define in the previous part. $\theta^1, \dots, \theta^M$ are adjustable parameters in (40) . $\mu_{A_1^1}(x_1), \dots, \mu_{A_n^m}(x_n)$ 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 (42). We define $f^{\wedge}(x|\theta)$ as the approximator of the real function $f(x)$.

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

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

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

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.
The fuzzy system can be defined as below

$$f(x) = \tau_{fuzzy} = \sum_{l=1}^{M} \theta^T \zeta(x) = \psi(e, \dot{e}) \quad (48)$$

where $\theta = (\theta^1, \theta^2, \theta^3, \dots\dots, \theta^M)^T, \zeta(x) = (\zeta^1(x), \zeta^2(x), \zeta^3(x), \dots\dots, \zeta^M(x))^T$

$$\zeta^1(x) = \frac{\sum_i \mu_{(xi)} x_i}{\sum_i \mu_{(xi)}} \quad (49)$$

where $\theta = (\theta^1, \theta^2, \theta^3, \dots\dots, \theta^M)$ is adjustable parameter in (44) and $\mu_{(xi)}$ is membership function.
error base fuzzy controller can be defined as

$$\tau_{fuzzy} = \psi(e, \dot{e}) \quad (50)$$

According to the formulation sliding function

$$if\ S = 0\ then\ -\dot{e} = \lambda e \quad (51)$$

the fuzzy division can be reached the best state when $S.\dot{S} < 0$ and the error is minimum by the following formulation

$$\theta^* = \arg\min\left[Sup_{x \in U} \left| \sum_{l=1}^{M} \theta^T \zeta(x) - \tau_{equ} \right|\right] \quad (52)$$

Where $\theta^{*}$is the minimum error, $sup_{x\in U}|\sum_{l=1}^{M}\theta^{T}\zeta(x)-\tau_{equ}|$ is the minimum approximation error. The adaptive controller is used to find the minimum errors of $\theta-\theta^{*}$.

suppose K_j is defined as follows

$$K_j=\frac{\sum_{l=1}^{M}\theta_j^l[\mu_A(S_j)]}{\sum_{l=1}^{M}[\mu_A(S_j)]}=\theta_j^T\zeta_j(S_j) \quad (53)$$

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

$$\zeta_j^1(S_j)=\frac{\mu_{(A)}{}_j^l(S_j)}{\sum_i\mu_{(A)}{}_j^l(S_j)} \quad (54)$$

the adaption low is defined as

$$\dot{\theta}_j=\gamma_{sj}S_j\zeta_j(S_j) \quad (55)$$

where the $\gamma_{sj}$is the positive constant.

According to the formulation (53) and (54) in addition from (50) and (48)

$$M(q)\ddot{q}+V(q,\dot{q})\dot{q}+G(q)=\sum_{l=1}^{M}\theta^{T}\zeta(x)-\lambda S-K \quad (56)$$

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

$$M\dot{S}=-VS+M\dot{S}+VS+G-\tau \quad (57)$$

It is supposed that

$$S^T(\dot{M}-2V)S=0 \quad (58)$$

it can be shown that

$$M\dot{S}+(V+\lambda)S=\Delta f-K \quad (59)$$

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

as a result $\dot{V}$ is became

$$\dot{V}=\frac{1}{2}S^T\dot{M}S-S^TVS+\sum\frac{1}{\gamma_{sj}}\phi_j^T\dot{\phi}_j$$

$$=S^T(-\lambda S+\Delta f-K)+\sum\frac{1}{\gamma_{sj}}\phi_j^T\dot{\phi}_j$$

$$=\sum_{j=1}^{m}[S_j(\Delta f_j-K_j)]-S^T\lambda S+\sum\frac{1}{\gamma_{sj}}\phi_j^T\dot{\phi}_j$$

$$=\sum_{j=1}^{m}[S_j(\Delta f_j-\theta_j^T\zeta_j(S_j))]-S^T\lambda S+\sum\frac{1}{\gamma_{sj}}\phi_j^T\dot{\phi}_j$$

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

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

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

consequently $\dot{V}$ can be considered by

$$\dot{V}=\sum_{j=1}^{m}[S_j\Delta f_j-((\theta_j^*)^T\zeta_j(S_j))]-S^T\lambda S \quad (60)$$

the minimum error can be defined by

$$e_{mj} = \Delta f_j - \left((\theta_j^*)^T \zeta_j(S_j)\right) \quad (61)$$

$\dot{V}$ is intended as follows

$$\dot{V} = \sum_{j=1}^{m} [S_j e_{mj}] - S^T \lambda S$$
$$\leq \sum_{j=1}^{m} |S_j||e_{mj}| - S^T \lambda S$$
$$= \sum_{j=1}^{m} |S_j||e_{mj}| - \lambda_j S_j^{\ 2}$$
$$= \sum_{j=1}^{m} |S_j|(|e_{mj}| - \lambda_j S_j) \quad (62)$$

For continuous function $g(x)$, and suppose $\varepsilon > 0$ it is defined the fuzzy logic system in form of (46) such that

$$Sup_{x \in U} |f(x) - g(x)| < \epsilon \quad (63)$$

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

$$if\ \lambda_j = \alpha \quad that \quad \alpha|S_j| > e_{mj}\ (S_j \neq 0) \quad then \quad \dot{V} < 0\ for\ (S_j \neq 0) \quad (64)$$

Figure 5 is shown the proposed method which it has an acceptable performance.

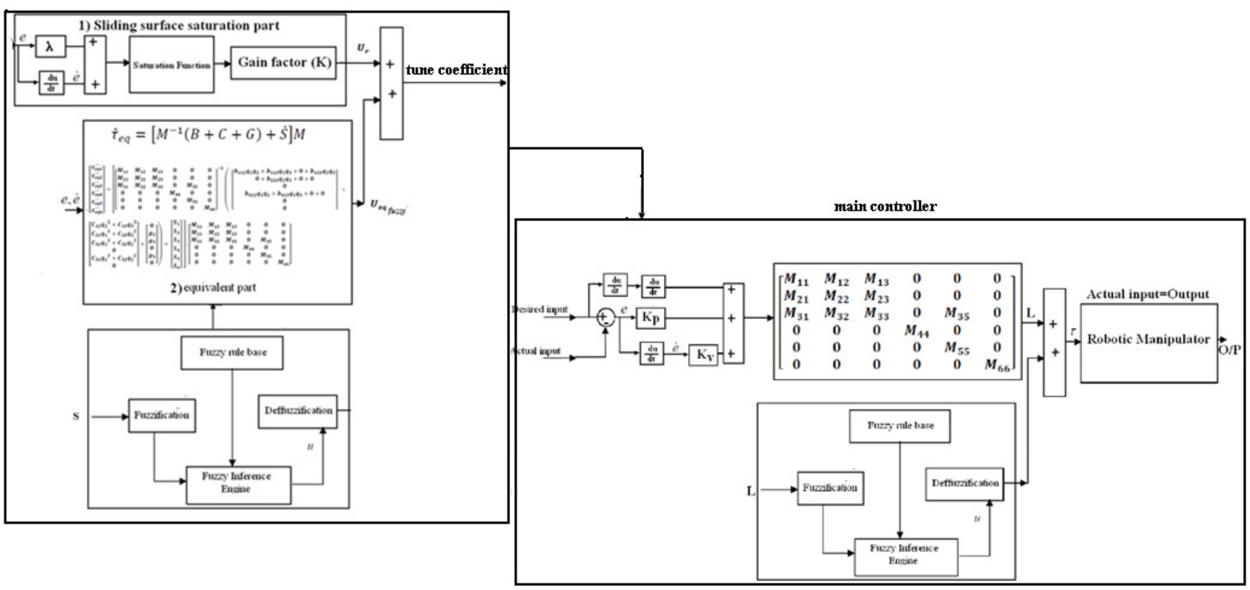

Figure 5: Proposed adaptive fuzzy sliding mode algorithm applied to inverse dynamic like controller: applied to robot manipulator

4. SIMULATION RESULTS

Inverse dynamic controller and SISO proposed adaptive inverse dynamic like controller were tested to ramp response trajectory. This simulation applied to three degrees of freedom robot arm therefore the first, second and third joints are moved from home to final position without and with external disturbance. The simulation was implemented in Matlab/Simulink environment. Trajectory performance, torque performance, disturbance rejection, steady state error and RMS error are compared in these controllers. It is noted that, these systems are tested by band limited white noise with a predefined 40% of relative to the input signal amplitude. This type of noise is used to external disturbance in continuous and hybrid systems.

Tracking Performances

Figure 6 is shown tracking performance for first, second and third link in inverse dynamic control and adaptive inverse dynamic like control without disturbance for ramp trajectories. By comparing ramp response trajectory without disturbance in inverse dynamic controller and adaptive inverse dynamic like controller it is found that the inverse dynamic controller's overshoot (1%) is higher than adaptive inverse dynamic like controller (0%), although almost both of them have about the same rise time.

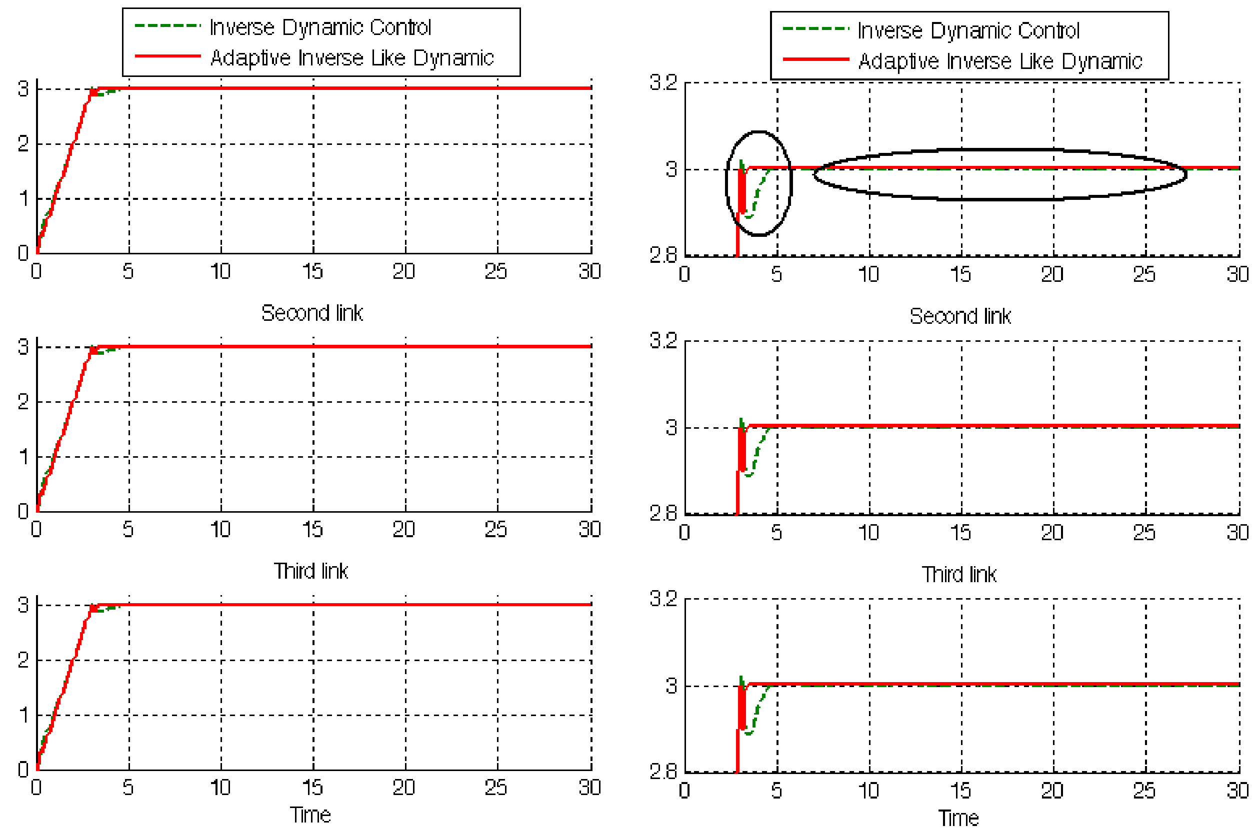

FIGURE 6: Inverse dynamic control Vs. adaptive inverse dynamic like controller trajectory: applied to robot manipulator.

Disturbance Rejection

Figure 7 has shown the power disturbance elimination in inverse dynamic control and adaptive inverse dynamic like control. The main target in these controllers is disturbance rejection as well as reduces the oscillation. A band limited white noise with predefined of 40% the power of input signal is applied to above controllers. It found fairly fluctuations in inverse dynamic control trajectory responses.

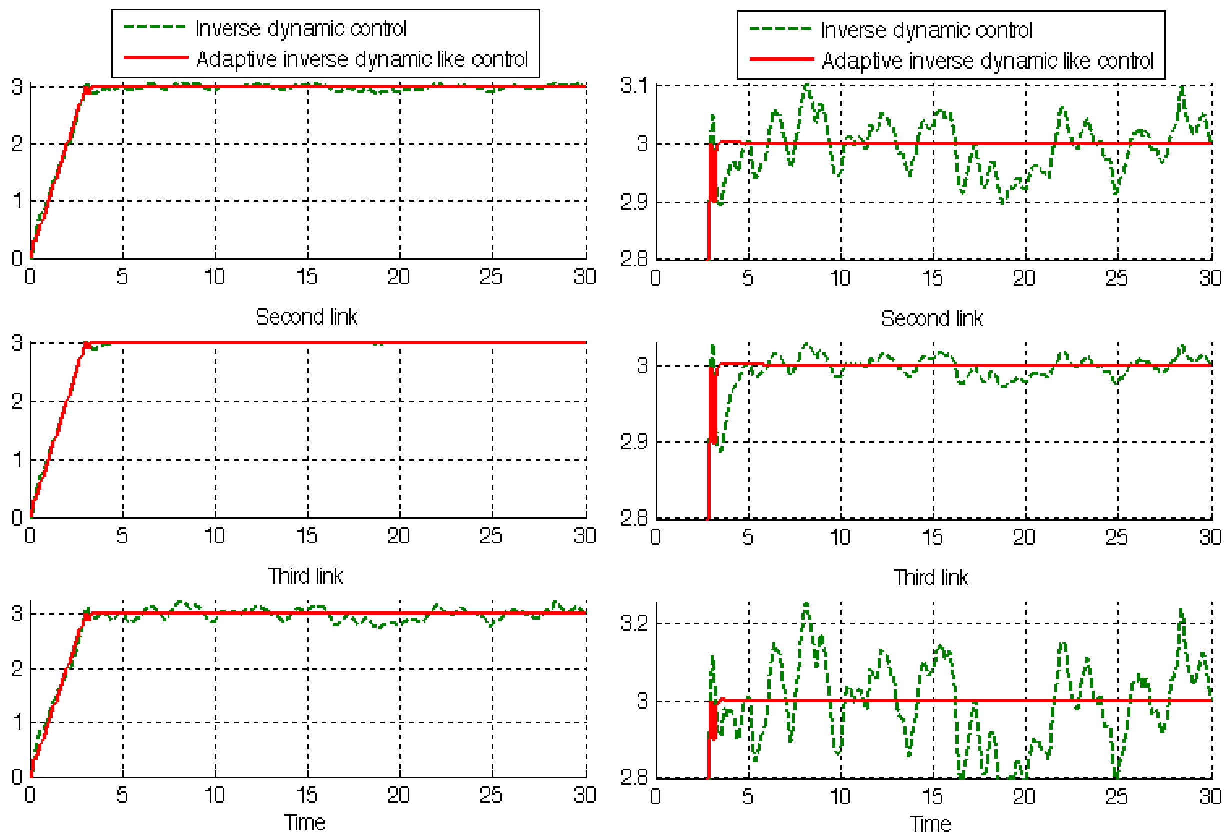

FIGURE 7: Inverse dynamic controller Vs inverse dynamic like controller trajectory with external disturbance: applied to robot manipulator

Among above graph relating to trajectory following with external disturbance, inverse dynamic controller has fairly fluctuations. By comparing some control parameters such as overshoot and rise time it found that the inverse dynamic control's overshoot (10%) is higher than adaptive inverse dynamic like controller (0%).

Error Calculation

Figure 8 and Table 1 are shown error performance in inverse dynamic controller and adaptive inverse dynamic like controller in presence of external disturbance. Inverse dynamic controller has oscillation in tracking which causes instability. As it is obvious in Table 2 the integral of absolute error is calculated to compare between classical method and proposed adaptive classic combined by artificial intelligence method. Figure 8 is shown steady state and RMS error in inverse dynamic control and adaptive inverse dynamic like control in presence of external disturbance.

TABLE 1: RMS Error Rate of Presented controllers

RMS Error Rate	**Inverse dynamic controller**	**Adaptive inverse dynamic like controller**
Without Noise	**1.8e-3**	**1e-4**
With Noise	**0.012**	**1.3e-4**

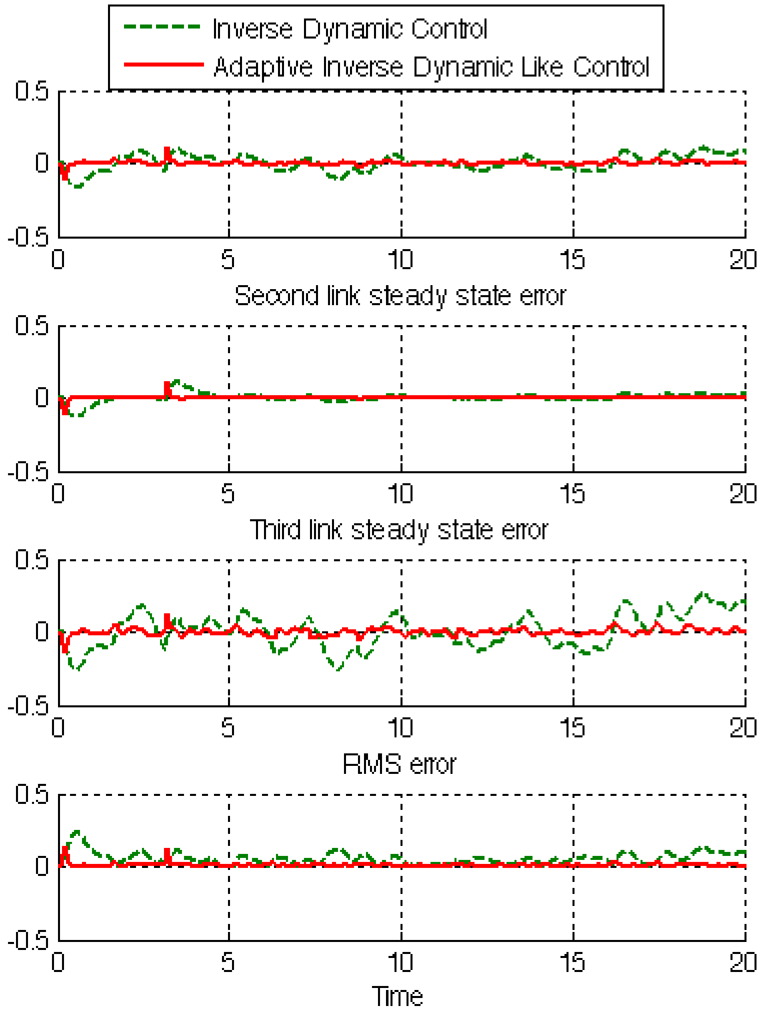

FIGURE 8: Adaptive inverse dynamic like controller Vs. inverse dynamic controller error performance with external disturbance: applied to robot manipulator

In these methods if integration absolute error (IAE) is defined by (75), table 2 is shown comparison between these two methods.

$$IAE = \int_{0}^{\infty} |e(t)|\, dt \tag{65}$$

TABLE 2: Calculate IAE

Method	Traditional IDC	Fuzzy Estimator IDC	AIDLC
IAE	490.1	411	202

5. CONCLUSIONS

In this research, a novel SISO adaptive fuzzy sliding algorithm inverse dynamic like method design and application to robotic manipulator has proposed in order to design high performance nonlinear controller in the presence of uncertainties. Each method by adding to the previous controller has covered negative points. The main target in this research is analyses and design of adaptive inverse dynamic like controller for robot manipulator to reach an acceptable performance. Robot manipulators are nonlinear and a number of parameters are uncertain, this research focuses on implement these controllers as accurate as possible using both analytical and empirical paradigms and the advantages and disadvantages of each one is presented through a comparative study, inverse dynamic controller and adaptive inverse dynamic like controller is used to selected the best controller for the industrial manipulator. In the first part studies about inverse dynamic controller show that: although this controller has acceptable performance with known dynamic parameters such as stability and robustness but there are two important disadvantages as below: oscillation and mathematical nonlinear dynamic in controller part. Second step focuses on applied fuzzy inference method as estimate in inverse dynamic controller to solve the dynamic nonlinear part problems in classical inverse dynamic controller. This controller works very well in certain and sometimes in uncertain environment but it has high computation in uncertain area. The system performance in inverse dynamic control and inverse dynamic like controller are sensitive to the controller gain, area and external disturbance. Therefore, compute the optimum value of controller gain for a system is the third important challenge work. This problem has solved by adjusting controller gain of the adaptive method continuously in real-time. In this way, the overall system performance has improved with respect to the classical inverse dynamic controller. This controller solved oscillation as well as mathematical nonlinear dynamic part by applied fuzzy supervisory estimated method in fuzzy inverse dynamic like controller and tuning the controller gain. By comparing between adaptive inverse dynamic like controller and inverse dynamic like controller, found that adaptive fuzzy inverse dynamic like controller has steadily stabilised in output response (e.g., disturbance rejection) but inverse dynamic controller has slight oscillation in the presence of uncertainties.

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] Piltan, F., et al., "Design sliding mode controller for robot manipulator with artificial tunable gain," Canaidian Journal of pure and applied science, 5 (2): 1573-1579, 2011.

[4] L. Cheng*, et al.*, "Multi-agent based adaptive consensus control for multiple manipulators with kinematic uncertainties," 2008, pp. 189-194.

[5] J. J. D'Azzo*, et al.*, *Linear control system analysis and design with MATLAB*: CRC, 2003.

[6] B. Siciliano and O. Khatib, *Springer handbook of robotics*: Springer-Verlag New York Inc, 2008.

[7] I. Boiko*, et al.*, "Analysis of chattering in systems with second-order sliding modes," *IEEE Transactions on Automatic Control,* vol. 52, pp. 2085-2102, 2007.

[8] J. Wang*, et al.*, "Indirect adaptive fuzzy sliding mode control: Part I: fuzzy switching," *Fuzzy Sets and Systems,* vol. 122, pp. 21-30, 2001.

[9] Farzin Piltan, A. R. Salehi and Nasri B Sulaiman.," Design artificial robust control of second order system based on adaptive fuzzy gain scheduling," International Journal of Robotics and Automation (IJRA), 2 (4), 2011

[10] F. Piltan*, et al.*, "Artificial Control of Nonlinear Second Order Systems Based on AFGSMC," *Australian Journal of Basic and Applied Sciences, 5(6)*, pp. 509-522, 2011.

[11] Piltan, F., et al., "Design Artificial Nonlinear Robust Controller Based on CTLC and FSMC with Tunable Gain," International Journal of Robotic and Automation, 2 (3): 205-220, 2011.

[12] Piltan, F., et al., "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] Piltan, F., et al., "Design of FPGA based sliding mode controller for robot manipulator," International Journal of Robotic and Automation, 2 (3): 183-204, 2011.

[14] Piltan, F., et al., "A Model Free Robust Sliding Surface Slope Adjustment in Sliding Mode Control for Robot Manipulator," World Applied Science Journal, 12 (12): 2330-2336, 2011.

[15] Harashima F., Hashimoto H., and Maruyama K, 1986. Practical robust control of robot arm using variable structure system, IEEE conference, P.P:532-539

[16] Piltan, F., et al., "Design Adaptive Fuzzy Robust Controllers for Robot Manipulator," World Applied Science Journal, 12 (12): 2317-2329, 2011.

[17] V. Utkin, "Variable structure systems with sliding modes," *Automatic Control, IEEE Transactions on,* vol. 22, pp. 212-222, 2002.

[18] R. A. DeCarlo*, et al.*, "Variable structure control of nonlinear multivariable systems: a tutorial," *Proceedings of the IEEE,* vol. 76, pp. 212-232, 2002.

[19] K. D. Young*, et al.*, "A control engineer's guide to sliding mode control," 2002, pp. 1-14.

[20] O. Kaynak, "Guest editorial special section on computationally intelligent methodologies and sliding-mode control," *IEEE Transactions on Industrial Electronics,* vol. 48, pp. 2-3, 2001.

[21] J. J. Slotine and S. Sastry, "Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators†," *International Journal of Control,* vol. 38, pp. 465-492, 1983.

[22] J. J. E. Slotine, "Sliding controller design for non-linear systems," *International Journal of Control,* vol. 40, pp. 421-434, 1984.

[23] R. Palm, "Sliding mode fuzzy control," 2002, pp. 519-526.

[24] C. C. Weng and W. S. Yu, "Adaptive fuzzy sliding mode control for linear time-varying uncertain systems," 2008, pp. 1483-1490.

[25] M. Ertugrul and O. Kaynak, "Neuro sliding mode control of robotic manipulators," *Mechatronics,* vol. 10, pp. 239-263, 2000.

[26] P. Kachroo and M. Tomizuka, "Chattering reduction and error convergence in the sliding-mode control of a class of nonlinear systems," *Automatic Control, IEEE Transactions on,* vol. 41, pp. 1063-1068, 2002.

[27] H. Elmali and N. Olgac, "Implementation of sliding mode control with perturbation estimation (SMCPE)," *Control Systems Technology, IEEE Transactions on,* vol. 4, pp. 79-85, 2002.

[28] J. Moura and N. Olgac, "A comparative study on simulations vs. experiments of SMCPE," 2002, pp. 996-1000.

[29] Y. Li and Q. Xu, "Adaptive Sliding Mode Control With Perturbation Estimation and PID Sliding Surface for Motion Tracking of a Piezo-Driven Micromanipulator," *Control Systems Technology, IEEE Transactions on,* vol. 18, pp. 798-810, 2010.

[30] B. Wu*, et al.*, "An integral variable structure controller with fuzzy tuning design for electro-hydraulic driving Stewart platform," 2006, pp. 5-945.

[31] L. A. Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets and Systems,* vol. 90, pp. 111-127, 1997.

[32] L. Reznik, *Fuzzy controllers*: Butterworth-Heinemann, 1997.

[33] J. Zhou and P. Coiffet, "Fuzzy control of robots," 2002, pp. 1357-1364.

[34] S. Banerjee and P. Y. Woo, "Fuzzy logic control of robot manipulator," 2002, pp. 87-88.

[35] K. Kumbla*, et al.*, "Soft computing for autonomous robotic systems," *Computers and Electrical Engineering,* vol. 26, pp. 5-32, 2000.

[36] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller. I," *IEEE Transactions on systems, man and cybernetics,* vol. 20, pp. 404-418, 1990.

[37] R. J. Wai*, et al.*, "Implementation of artificial intelligent control in single-link flexible robot arm," 2003, pp. 1270-1275.

[38] R. J. Wai and M. C. Lee, "Intelligent optimal control of single-link flexible robot arm," *Industrial Electronics, IEEE Transactions on,* vol. 51, pp. 201-220, 2004.

[39] M. B. Menhaj and M. Rouhani, "A novel neuro-based model reference adaptive control for a two link robot arm," 2002, pp. 47-52.

[40] S. Mohan and S. Bhanot, "Comparative study of some adaptive fuzzy algorithms for manipulator control," *International Journal of Computational Intelligence,* vol. 3, pp. 303–311, 2006.

[41] F. Barrero*, et al.*, "Speed control of induction motors using a novel fuzzy sliding-mode structure," *Fuzzy Systems, IEEE Transactions on,* vol. 10, pp. 375-383, 2002.

[42] Y. C. Hsu and H. A. Malki, "Fuzzy variable structure control for MIMO systems," 2002, pp. 280-285.

[43] Y. C. Hsueh*, et al.*, "Self-tuning sliding mode controller design for a class of nonlinear control systems," 2009, pp. 2337-2342.

[44] R. Shahnazi*, et al.*, "Position control of induction and DC servomotors: a novel adaptive fuzzy PI sliding mode control," *Energy Conversion, IEEE Transactions on,* vol. 23, pp. 138-147, 2008.

[45] C. C. Chiang and C. H. Wu, "Observer-Based Adaptive Fuzzy Sliding Mode Control of Uncertain Multiple-Input Multiple-Output Nonlinear Systems," 2007, pp. 1-6.

[46] H. Temeltas, "A fuzzy adaptation technique for sliding mode controllers," 2002, pp. 110-115.

[47] C. L. Hwang and S. F. Chao, "A fuzzy-model-based variable structure control for robot arms: theory and experiments," 2005, pp. 5252-5258.

[48] C. G. Lhee*, et al.*, "Sliding mode-like fuzzy logic control with self-tuning the dead zone parameters," *Fuzzy Systems, IEEE Transactions on,* vol. 9, pp. 343-348, 2002.

[49] Lhee. C. G., J. S. Park, H. S. Ahn, and D. H. Kim, "Sliding-Like Fuzzy Logic Control with Self-tuning the Dead Zone Parameters," *IEEE International fuzzy systems conference proceeding*, 1999,pp.544-549.

[50] X. Zhang*, et al.*, "Adaptive sliding mode-like fuzzy logic control for high order nonlinear systems," pp. 788-792.

[51] M. R. Emami, *et al.*, "Development of a systematic methodology of fuzzy logic modeling," *IEEE Transactions on Fuzzy Systems,* vol. 6, 1998.

[52] H.K.Lee, K.Fms, "A Study on the Design of Self-Tuning Sliding Mode Fuzzy Controller. Domestic conference," *IEEE Conference*, *1994,* vol. 4, pp. 212-218.

[53] Z. Kovacic and S. Bogdan, *Fuzzy controller design: theory and applications*: CRC/Taylor & Francis, 2006.

[54] F. Y. Hsu and L. C. Fu, "Nonlinear control of robot manipulators using adaptive fuzzy sliding mode control," 2002, pp. 156-161.

[55] R. G. Berstecher, *et al.*, "An adaptive fuzzy sliding-mode controller," *Industrial Electronics, IEEE Transactions on,* vol. 48, pp. 18-31, 2002.

[56] V. Kim, "Independent joint adaptive fuzzy control of robot manipulator," 2002, pp. 645-652.

[57] Y. Wang and T. Chai, "Robust adaptive fuzzy observer design in robot arms," 2005, pp. 857-862.

[58] B. K. Yoo and W. C. Ham, "Adaptive control of robot manipulator using fuzzy compensator," *Fuzzy Systems, IEEE Transactions on,* vol. 8, pp. 186-199, 2002.

[59] H. Medhaffar, *et al.*, "A decoupled fuzzy indirect adaptive sliding mode controller with application to robot manipulator," *International Journal of Modelling, Identification and Control,* vol. 1, pp. 23-29, 2006.

[60] Y. Guo and P. Y. Woo, "An adaptive fuzzy sliding mode controller for robotic manipulators," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* vol. 33, pp. 149-159, 2003.

[61] C. M. Lin and C. F. Hsu, "Adaptive fuzzy sliding-mode control for induction servomotor systems," *Energy Conversion, IEEE Transactions on,* vol. 19, pp. 362-368, 2004.

[62] Iordanov, H. N., B. W. Surgenor, 1997. Experimental evaluation of the robustness of discrete sliding mode control versus linear quadratic control, IEEE Trans. On control system technology, 5(2):254-260.

[63] A. Vivas and V. Mosquera, "Predictive functional control of a PUMA robot," 2005.

[64] D. Nguyen-Tuong, *et al.*, "Computed torque control with nonparametric regression models," 2008, pp. 212-217.

[65] Farzin Piltan, *et al.*, "Design of model free adaptive fuzzy computed torque controller for a nonlinear second order system," International Journal of Robotics and Automation (IJRA), 2(4), 2011.