Evolutionary Design of Mathematical tunable FPGA Based MIMO Fuzzy Estimator Sliding Mode Based Lyapunov Algorithm: Applied to Robot Manipulator

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

Amin Jalali Max.Jalali@gmail.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

Koorosh Aslansefat koo.ec2008@gmail.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

In this research, a Multi Input Multi Output (MIMO) position Field Programmable Gate Array (FPGA)-based fuzzy estimator sliding mode control (SMC) design with the estimation laws derived in Lyapunov sense and application to robotic manipulator has proposed in order to design high performance nonlinear controller in the presence of uncertainties. Regarding to the positive points in sliding mode controller, fuzzy inference methodology and Lyapunov based method, the controllers output has improved. The main target in this research is analyses and design of the position MIMO artificial Lyapunov FPGA-based controller for robot manipulator in order to solve uncertainty, external disturbance, nonlinear equivalent part, chattering phenomenon, time to market and controller size using FPGA. Robot manipulators are nonlinear, time variant and a number of parameters are uncertain therefore design robust and stable controller based on Lyapunov based is discussed in this research. Studies about classical sliding mode controller (SMC) 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: chattering phenomenon and mathematical nonlinear dynamic equivalent controller part. The first challenge; nonlinear dynamic part; is applied by inference estimator method in sliding mode controller in order to solve the nonlinear problems in classical sliding mode controller. And the second challenge; chattering phenomenon; is removed by linear method. Asymptotic stability of the closed loop system is also proved in the sense of Lyapunov. In the last part it can find the implementation of MIMO fuzzy estimator sliding mode controller on FPGA; FPGA-based fuzzy estimator sliding mode controller has many advantages such as high speed, low cost, short time to market and small device size. One of the most important drawbacks is limited capacity of available cells which this research focuses to solve this challenge. FPGA can be used to design a controller in a single chip Integrated Circuit (IC). In this research the SMC is designed using Very High Description Language (VHDL) for implementation on FPGA device (XA3S1600E-Spartan-3E), with minimum chattering.

Keywords: Mathematical Tunable, FPGA, MIMO Fuzzy Estimator, Fuzzy Sliding Mode Based Lyapunov Algorithm, Robot Manipulator, Chattering Phenomenon, VHDL language.

1. INTRODUCTION, MOTIVATION AND BACKGROUND

Robot manipulators have many applications in aerospace, manufacturing, automotive, medicine and other industries. Robot manipulators consist of three main parts: mechanical, electrical, and control. In the mechanical point of view, robot manipulators are collection of serial or parallel links which have connected by revolute and/or prismatic joints between base and end-effector frame. The robot manipulators electrical parts are used to links motion, which including the following subparts: power supply to supply the electrical and control parts, power amplifier to amplify the signal and driving the actuators, DC/stepper/servo motors or hydraulic/pneumatic cylinders to motion the links, and transmission part to transfer data between robot manipulator subparts. Control part is used to adjust the timing between the subparts of robot manipulator to reach the best performance (trajectory). It provides four main abilities in robot manipulators: controlling the manipulators movement in correct workspace, sensing the information from the environment, being able to intelligent control behavior and processing the data and information between all subparts. Research about mechanical parts and control methodologies in robotic system is shown; the mechanical design, type of actuators, and type of systems drive play important roles to have the best performance controller. More over types of kinematics chain, i.e., serial Vs. parallel manipulators, and types of connection between link and join actuators, i.e., highly geared systems Vs. direct-drive systems are presented in the following sentences because these topics played important roles to select and design the best acceptable performance controllers[1-6]. A serial link robot is a sequence of joints and links which begins with a base frame and ends with an end-effector. This type of robot manipulators, comparing with the load capacitance is more weightily because each link must be supported the weights of all next links and actuators between the present link and end-effector[6]. Serial robot manipulators have been used in automotive industry, medical application, and also in research laboratories. In contrast, parallel robot manipulators design according to close loop which base frame is connected to the end-effector frame with two or more kinematic chains[6]. In the other words, a parallel link robot has two or more branches with some joints and links, which support the load in parallel. Parallel robot have been used in many applications such as expensive flight simulator, medical robotics (I.e., high accuracy, high repeatability, high precision robot surgery), and machinery tools. With comparison between serial and parallel links robot manipulators, parallel robots are used in higher speed loads, better accuracy, with used lighter weigh robot manipulator but one of the most important handicaps is limitation the workspace compared to serial robot. From control point of view, the coupling between different kinematic chains can generate the uncertainty problems which cause difficult controller design of parallel robot manipulator[7-12]. One of the most important classifications in controlling the robot manipulator is how the links have connected to the actuators. This classification divides into two main groups: highly geared (e.g., 200 to 1) and direct drive (e.g., 1 to 1). High gear ratios reduce the nonlinear coupling dynamic parameters in robot manipulator. In this case, each joint is modeled the same as Single Input Single Output (SISO) systems. In high gear robot manipulators which generally are used in industry, the couplings are modeled as a disturbance for SISO systems. Direct drive increases the coupling of nonlinear dynamic parameters of robot manipulators. This effect should be considered in the design of control systems. As a result some control and robotic researchers' works on nonlinear robust controller design[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]. Sliding mode controller (SMC) is a significant nonlinear controller in certain and uncertain dynamic parameters systems. This controller is used to present a systematic solution for stability and robustness which they can play important role to select the best controller. Conversely, pure sliding mode controller is used in many applications; it has two important drawbacks namely; chattering phenomenon, and nonlinear equivalent dynamic formulation in uncertain dynamic parameter [1, 6, and 20]. To reduce or eliminate the chattering this research is used linear saturation boundary layer function [13-20]. In boundary layer saturation method, the basic idea is the discontinuous method replacement by saturation (linear) method with small neighborhood of the switching surface. This replacement caused to increase the error performance against with the considerable chattering reduction. Slotine and Sastry have introduced boundary layer method instead of discontinuous method to reduce the chattering[21]. Slotine has presented sliding mode with boundary layer to improve the industry application [22]. R. Palm has presented a fuzzy method to nonlinear approximation instead of linear approximation inside the boundary layer to improve the chattering and control the result performance[23]. Moreover, C. C. Weng and W. S. Yu improved the previous method by using a new method in fuzzy nonlinear approximation inside the boundary layer and adaptive method[24]. As mentioned [24]sliding mode fuzzy controller (SMFC) is fuzzy controller based on sliding mode technique to simple implement, most exceptional stability and robustness. Conversely above method has the following advantages; reducing the number of fuzzy rule base and increasing robustness and stability, the main disadvantage of SMFC is need to define the sliding surface slope coefficient very carefully. To eliminate the above problems control researchers have applied artificial intelligence method (e.g., fuzzy logic) in nonlinear robust controller (e.g., sliding mode controller) besides this technique is very useful in order to implement easily. One of the most important techniques to reduce or remove above two challenges is applying non-classical (artificial intelligence) method in robust classical such as sliding mode controller method. Estimated uncertainty method used in term of uncertainty estimator to compensation of the system uncertainties. It has been used to solve the chattering phenomenon and also nonlinear equivalent dynamic. If estimator has an acceptable performance to compensate the uncertainties, the chattering is reduced. Research on estimated uncertainty to reduce the chattering is significantly growing as their applications such as industrial automation and robot manipulator. For instance, the applications of artificial intelligence, neural networks and fuzzy logic on estimated uncertainty method have been reported in [25-28]. Wu et al. [30] have proposed a simple fuzzy estimator controller beside the discontinuous and equivalent control terms to reduce the chattering. Their design had three main parts i.e. equivalent, discontinuous and fuzzy estimator tuning part which has reduced the chattering very well. Elmali et al. [27]and Li and Xu [29]have addressed sliding mode control with perturbation estimation method (SMCPE) to reduce the classical sliding mode chattering. This method was tested for the tracking control of the first two links of a SCARA type HITACHI robot. In this technique, digital controller is used to increase the system's response quality. Conversely this method has the following advantages; increasing the controller's response speed and reducing dependence on dynamic system model by on-line control, the main disadvantage are chattering phenomenon and need to improve the performance. In order to solve the uncertain dynamic parameters and complex parameters systems with an artificial intelligence theory, fuzzy logic is one of the best choice which it is used in this research. However fuzzy logic method is useful to control complicated nonlinear dynamic mathematical models but the response quality may not always be so high. This controller can be used in main part of controller (e.g., pure fuzzy logic controller), it can be used to design adaptive controller (e.g., adaptive fuzzy controller), tuning parameters and finally applied to the classical controllers [31-40]. Research on combinations of fuzzy logic systems with sliding mode method is significantly growing as nonlinear control applications. For instance, the applications of fuzzy logic on sliding mode controller have reported in [24, 41-45]. 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 [42-43]. 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].

As mentioned above sliding mode controller has some limitations which applied fuzzy logic in sliding mode controller can causes to reduce the limitations [48-53]. However FSMC has an acceptable performance but calculate the sliding surface slope by experience knowledge is difficult, particularly when system has structure or unstructured uncertainties, mathematical model free on-line tunable gain is recommended. F Y Hsu et al. [54]have presented adaptive fuzzy sliding mode control which can update fuzzy rules to compensate nonlinear parameters and guarantee the stability robot manipulator controller. Y.C. Hsueh et al. [43] have presented self tuning sliding mode controller which can resolve the chattering problem without to using saturation function. For nonlinear dynamic systems (e.g., robot manipulators) with various parameters, adaptive control technique can train the dynamic parameter to have an acceptable controller performance. Calculate several scale factors are common challenge in classical sliding mode controller and fuzzy logic controller, as a result it is used to adjust and tune coefficient. Research on adaptive fuzzy control is significantly growing, for instance, different adaptive fuzzy controllers have been reported in [40, 55-57]. Research on adaptive fuzzy sliding mode controller is significantly growing as many applications and it can caused to improve the tracking performance by online tuning the parameters. The adaptive sliding mode controller is used to estimate the unknown dynamic parameters and external disturbances. For instance, the applications of adaptive fuzzy sliding mode controller to control the robot manipulators have been reported in [24, 29, 45]. Generally, adaptive fuzzy sliding mode control of robot manipulator is classified into two main groups' i.e. multi-input multi-output (MIMO) and single-input single-output (SISO) fuzzy systems.

Commonly, most of nonlinear controllers in robotic applications need a real time operation. FPGA-based controller has been used in this application because it is small device in size, high speed, low cost, and short time to market. Therefore FPGA-based controller can have a short execution time because it has parallel architecture. Research on FPGA-based control of systems is considerably growing as their applications such as industrial automation, robotic surgery, and space station's robot arm demand more accuracy, reliability, high performance. For instance, the FPGA-based controls of robot manipulator have been reported in [63-70]. Shao and Sun [64]have proposed an adaptive control algorithm based on FPGA for control of SCARA robot manipulator. They are designed this controller into two micro base controller, the linear part controller is implemented in the FPGA and the nonlinear estimation controller is implemented in DSP. Moreover this controller is implemented in a Xilinx-FPGA XC3S400 with the 20 KHz position loop frequency. The FPGA based servo control and inverse kinematics for Mitsubishi RV-M1 micro robot is presented in[65, 67] which to reduce the limitation of FPGA capacitance they are used 42 steps finite state machine (FSM) in 840 n second. Meshram and Harkare [68-69] have presented a multipurpose FPGA-based 5 DOF robot manipulator using VHDL coding in Xilinx ISE 11.1. This controller has two most important advantages: easy to implement and flexible. Zeyad Assi Obaid et al. [71] have proposed a digital PID fuzzy logic controller using FPGA for tracking tasks that yields semi-global stability of all closed-loop signals. The basic information about FPGA have been reported in [63, 69-73]. A review of design and implementation of FPGA-based systems has been presented in [63]. The FPGA-based sliding mode control of systems has been reported in [74-77]. Lin et al. [74] have presented low cost and high performance FPGA-based fuzzy sliding mode controller for linear induction motor with 80% of flip flops. The fuzzy inference system has 2 inputs $(S \,\&\, \dot{S})$ and one output K_f with nine rules. Ramos et al. [75] have reported FPGA-based fixed frequency quasi sliding mode control algorithm to control of power inverter. Their proposed controller is implemented in XC4010E-3-PC84 FPGA from XILINX with acceptable experimental and theoretical performance. FPGA-based robust adaptive backstepping sliding mode control for verification of induction motor is reported in [76]. A FPGA chip has programmed by Hardware Description Language (HDL) which contains two types of languages, Very High Description Language (VHDL) and Verilog. VHDL is one of the powerful programming languages that can be used to describe the hardware design. VHDL was developed by the Institute of Electrical and Electronics Engineers (IEEE) in 1987 and Verilog was developed by Gateway Design Automation in 1984 [63, 72]. This research focuses on FPGA-based sliding mode control of robot manipulator and it is implemented in XA3S1600E FPGA from Xilinx in Xilinx-ISE 9.2i software using VHDL code.

In this research we will highlight the MIMO mathematical model-free adaptive fuzzy estimator sliding mode algorithm with estimates the equivalent part derived in the Lyapunov sense. This algorithm will be analyzed and evaluated on robotic manipulators. Section 2, serves as an introduction to problem formulation of controller and its application to a three degree of-freedom robot manipulator, 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 3 degree-of-freedom robot manipulator and the final section is describe the conclusion.

2. CONTROL FORMULATION, OBJECTIVES AND PROBLEM STATEMENTS

Fist part is focused on nonlinear dynamic of PUMA 560 robot manipulator. Dynamic equation is the study of motion with regard to forces. Dynamic modeling is vital for control, mechanical design, and simulation. It is used to describe dynamic parameters and also to describe the relationship between displacement, velocity and acceleration to force acting on robot manipulator. To calculate the dynamic parameters which introduced in the following lines, four algorithms are very important: **Inverse dynamics**, in this algorithm, joint actuators are computed (e.g., force/torque or voltage/current) from endeffector position, velocity, and acceleration. It is used in feed forward control. **Forward dynamics** used to compute the joint acceleration from joint actuators. This algorithm is required for simulations. **The joint-space inertia matrix**, necessary for maps the joint acceleration to the joint actuators. It is used in analysis, feedback control and in some integral part of forward dynamics formulation. **The operational-space inertia matrix**, this algorithm maps the task accelerations to task actuator in Cartesian space. It is required for control of end-effector.

The field of dynamic robot manipulator has a wide literature that published in professional journals and established textbooks[1, 6]. Several different methods are available to compute robot manipulator dynamic equations. These methods include the Newton-Euler (N-E) methodology, the Lagrange-Euler (L-E) method, and Kane's methodology[1]. The Newton-Euler methodology is based on Newton's second law and several different researchers are signifying to develop this method[1]. This equation can be described the behavior of a robot manipulator link-by-link and joint-by-joint from base to endeffector, called forward recursion and transfer the essential information from end-effector to base frame, called backward recursive. The literature on Euler-Lagrange's is vast but a good starting point to learn about it is in[1]. Euler-Lagrange is a method based on calculation kinetic energy. Calculate the dynamic equation robot manipulator using E-L method is easier because this equation is derivation of nonlinear coupled and quadratic differential equations. The Kane's method was introduced in 1961 by Professor Thomas Kane[1, 6]. This method used to calculate the dynamic equation of motion without any differentiation between kinetic and potential energy functions. 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 [78-79]:

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

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

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

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;

$$M(\theta) = m_1 J_{v1}^T J_{v1} + J_{\omega1}^{TC1} I_1 J_{\omega1} + m_2 J_{v2}^T J_{v2} + J_{\omega2}^{TC2} I_2 J_{\omega2} + m_3 J_{v3}^T J_{v3} + J_{\omega3}^{TC3} I_3 J_{\omega3} + m_4 J_{v4}^T J_{v4} \\ m_5 J_{v5}^T J_{v5} + J_{\omega5}^{TC5} I_5 J_{\omega5} + m_6 J_{v6}^T J_{v6} + J_{\omega6}^{TC6} I_6 J_{\omega6} \quad (6)$$

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;

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

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;

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

Robotic control is one of the most active research areas in the field of robotics and one of the well-known robot manipulator in the field of academic and industries is PUMA 560 robot manipulator. To position control of robot manipulator, the second three axes are locked the dynamic equation of PUMA robot manipulator is given by [78-79];

$$M(\theta) \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + B(\theta) \begin{bmatrix} \dot{\theta}_1 \dot{\theta}_2 \\ \dot{\theta}_1 \dot{\theta}_3 \\ \dot{\theta}_2 \dot{\theta}_3 \end{bmatrix} + C(\theta) \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \\ \dot{\theta}_3^2 \end{bmatrix} + G(\theta) = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad (11)$$

Where

$$M(q) = \begin{bmatrix} M_{11} & M_{12} & M_{13} & 0 & 0 & 0 \\ M_{21} & M_{22} & M_{23} & 0 & 0 & 0 \\ M_{31} & M_{32} & M_{33} & 0 & M_{35} & 0 \\ 0 & 0 & 0 & M_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & M_{66} \end{bmatrix} \quad (12)$$

M is computed as

$$M_{11} = I_{m1} + I_1 + I_3 \times \cos(\theta_2)\cos(\theta_2) + I_7 \sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3) + I_{10}\sin(\theta_2+\theta_3) \\ I_{11}\sin(\theta_2)\cos(\theta_2) + I_{21}\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3) + 2 + [I_5\cos(\theta_2)\sin(\theta_2+\theta_3) + I_{12}\cos \\ \theta_3) + I_{15}\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3) + I_{16}\cos(\theta_2)\sin(\theta_2+\theta_3) + I_{22}\sin(\theta_2+\theta_3)\cos(\theta_2 + \quad (13)$$

$$M_{12} = I_4 \sin(\theta_2) + I_8 \cos(\theta_2+\theta_3) + I_9 \cos(\theta_2) + I_{13}\sin(\theta_2+\theta_3) - I_{18}\cos(\theta_2+\theta_3) \quad (14)$$

$$M_{13} = I_8 \cos(\theta_2+\theta_3) + I_{13}\sin(\theta_2+\theta_3) - I_{18}\cos(\theta_2+\theta_3) \quad (15)$$

$$M_{22} = I_{m2} + I_2 + I_6 + 2[I_5 \sin(\theta_3) + I_{12}\cos(\theta_2) + I_{15} + I_{16}\sin(\theta_3) \quad (16)$$

$$M_{23} = I_5 \sin(\theta_3) + I_6 + I_{12}\cos(\theta_3) + I_{16}\sin(\theta_3) + 2I_{15} \quad (17)$$

$$M_{33} = I_{m3} + I_6 + 2I_{15} \quad (18)$$

$$M_{35} = I_{15} + I_{17} \quad (19)$$

$$M_{44} = I_{m4} + I_{14} \quad (20)$$

$$M_{55} = I_{m5} + I_{17} \quad (21)$$

$$M_{66} = I_{m6} + I_{23} \quad (22)$$

$$M_{21} = M_{12}\ , M_{31} = M_{13}\ and\ M_{32} = M_{23} \quad (23)$$

and Corilios ($\boldsymbol{B}$) matrix is calculated as the following

$$B(q) = \begin{bmatrix} b_{112} & b_{113} & 0 & b_{115} & 0 & b_{123} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{214} & 0 & 0 & b_{223} & 0 & b_{225} & 0 & 0 & b_{235} & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{314} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ b_{412} & b_{412} & 0 & b_{415} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{514} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (24)$$

Where,

$$b_{112} = 2[-I_3 \sin(\theta_2)\cos(\theta_2) + I_5\cos(\theta_2 + \theta_2 + \theta_3) + I_7\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) - I_{12}s \quad (25)$$
$$\theta_3) - I_{15} 2\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) + I_{16}\cos(\theta_2 + \theta_2 + \theta_3) + I_{21}\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta$$
$$2\sin(\theta_2 + \theta_3)\sin(\theta_2 + \theta_3))] + I_{10}(1 - 2\sin(\theta_2 + \theta_3)\sin(\theta_2 + \theta_3)) + I_{11}(1 - 2\sin(\theta_2)\sin$$

$$b_{113} = 2[I_5\cos(\theta_2)\cos(\theta_2 + \theta_3) + I_7\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) - I_{12}\cos(\theta_2)\sin(\theta_2 + \theta_2 \quad (26)$$
$$\theta_3)\cos(\theta_2 + \theta_3) + I_{16}\cos(\theta_2)\cos(\theta_2 + \theta_3) + I_{21}\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) + I_{22}(1 - 2\sin$$
$$\theta_3)\sin(\theta_2 + \theta_3))] + I_{10}(1 - 2\sin(\theta_2 + \theta_3)\sin(\theta_2 + \theta_3))$$

$$b_{115} = 2[-\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) + I_{15} 2\sin(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3) + I_{16}\cos(\theta_2)\cos(\theta \quad (27)$$
$$I_{22}\cos(\theta_2 + \theta_3)\cos(\theta_2 + \theta_3)]$$

$$b_{123} = 2[-I_8\sin(\theta_2 + \theta_3) + I_{13}\cos(\theta_2 + \theta_3) + I_{18}\sin(\theta_2 + \theta_3)] \quad (28)$$

$$b_{214} = I_{14}\sin(\theta_2 + \theta_3) + I_{19}\sin(\theta_2 + \theta_3) + 2I_{20}\sin(\theta_2 + \theta_3)(1 - 0.5) \quad (29)$$

$$b_{223} = 2[-I_{12}\sin(\theta_3) + I_5\cos(\theta_3) + I_{16}\cos(\theta_3)] \quad (30)$$

$$b_{235} = 2[I_{16}\cos(\theta_3) + I_{22}] \quad (31)$$

$$b_{314} = 2[I_{20}\sin(\theta_2 + \theta_3)(1 - 0.5)] + I_{14}\sin(\theta_2 + \theta_3) + I_{19}\sin(\theta_2 + \theta_3) \quad (32)$$

$$b_{412} = b_{214} = -[I_{14}\sin(\theta_2+\theta_3) + I_{19}\sin(\theta_2+\theta_3) + 2I_{20}\sin(\theta_2+\theta_3)(1-0.5)] \quad (33)$$

$$b_{413} = -b_{314} = -2[I_{20}\sin(\theta_2+\theta_3)(1-0.5)] + I_{14}\sin(\theta_2+\theta_3) + I_{19}\sin(\theta_2+\theta_3) \quad (34)$$

$$b_{415} = -I_{20}\sin(\theta_2+\theta_3) - I_{17}\sin(\theta_2+\theta_3) \quad (35)$$

$$b_{514} = -b_{415} = I_{20}\sin(\theta_2+\theta_3) + I_{17}\sin(\theta_2+\theta_3) \quad (36)$$

consequently coriolis matrix is shown as bellows;

$$B(q).\dot{q}\dot{q} = \begin{bmatrix} b_{112}.\dot{q}_1\dot{q}_2 + b_{113}.\dot{q}_1\dot{q}_3 + 0 + b_{123}.\dot{q}_2\dot{q}_3 \\ 0 + b_{223}.\dot{q}_2\dot{q}_3 + 0 + 0 \\ 0 \\ b_{412}.\dot{q}_1\dot{q}_2 + b_{413}.\dot{q}_1\dot{q}_3 + 0 \\ 0 \\ 0 \end{bmatrix} \quad (37)$$

Moreover Centrifugal (C) matrix is demonstrated as

$$C(q) = \begin{bmatrix} 0 & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{21} & 0 & C_{23} & 0 & 0 & 0 \\ C_{31} & C_{32} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ C_{51} & C_{52} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (38)$$

Where,

$$c_{12} = I_4\cos(\theta_2) - I_8\sin(\theta_2+\theta_3) - I_9\sin(\theta_2) + I_{13}\cos(\theta_2+\theta_3) + I_{18}\sin(\theta_2+\theta_3) \quad (39)$$

$$c_{13} = 0.5b_{123} = -I_8\sin(\theta_2+\theta_3) + I_{13}\cos(\theta_2+\theta_3) + I_{18}\sin(\theta_2+\theta_3) \quad (40)$$

$$c_{21} = -0.5b_{112} = I_3\sin(\theta_2)\cos(\theta_2) - I_5\cos(\theta_2+\theta_2+\theta_3) - I_7\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_{\cdots}$$
$$\theta_2+\theta_3) + I_{15}2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{16}\cos(\theta_2+\theta_2+\theta_3) - I_{21}\sin(\theta_2+\theta_3)\cos(\cdots$$
$$I_{22}(1-2\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3)) - 0.5I_{10}(1-2\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3)) -$$
$$0.5I_{11}(1-2\sin(\theta_2)\sin(\theta_2)) \quad (41)$$

$$c_{22} = 0.5b_{223} = -I_{12}\sin(\theta_3) + I_5\cos(\theta_3) + I_{16}\cos(\theta_3) \quad (42)$$

$$c_{23} = -0.5b_{113} = -I_5\cos(\theta_2)\cos(\theta_2+\theta_3) - I_7\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) + I_{12}\cos(\theta_2)\cdots$$
$$I_{15}2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{16}\cos(\theta_2)\cos(\theta_2+\theta_3) - I_{21}\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3)\cdots$$
$$2\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3)) - 0.5I_{10}(1-2\sin(\theta_2+\theta_3)\sin(\theta_2+\theta_3)) \quad (43)$$

$$c_{31} = -c_{23} = I_{12}\sin(\theta_3) - I_5\cos(\theta_3) - I_{16}\cos(\theta_3) \quad (44)$$

$$c_{32} = -0.5b_{115} = \sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{15}2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3) - I_{16}\cos(\theta_{2}\cdots$$
$$I_{22}\cos(\theta_2+\theta_3)\cos(\theta_2+\theta_3) \quad (45)$$

$$c_{52} = -0.5b_{225} = -I_{16}\cos(\theta_3) - I_{22} \quad (46)$$

In this research $q_4 = q_5 = q_6 = 0$, as a result

$$C(q).q'^{2} = \begin{bmatrix} c_{112}.q_2'^{2} + c_{13}.q_3'^{2} \\ c_{21}.q_1'^{2} + c_{23}.q_3'^{2} \\ c_{13}.q_1'^{2} + c_{32}.q_2'^{2} \\ 0 \\ c_{51}.q_1'^{2} + c_{52}.q_2'^{2} \\ 0 \end{bmatrix} \quad (47)$$

Gravity (G) Matrix can be written as

$$G(q) = \begin{bmatrix} 0 \\ g_2 \\ g_3 \\ 0 \\ g_5 \\ 0 \end{bmatrix} \quad (48)$$

Where,

$$G_2 = g_1 \cos(\theta_2) + g_2 \sin(\theta_2 + \theta_3) + g_3 \sin(\theta_2) + g_4 \cos(\theta_2 + \theta_3) + g_5 \sin(\theta_2 + \theta_3) \quad (49)$$

$$G_3 = g_2 \sin(\theta_2 + \theta_3) + g_4 \cos(\theta_2 + \theta_3) + g_5 \sin(\theta_2 + \theta_3) \quad (50)$$

$$G_5 = g_5 \sin(\theta_2 + \theta_3) \quad (51)$$

Suppose $\ddot{q}$ is written as follows

$$\ddot{q} = M^{-1}(q).\{\tau - [B(q)\dot{q}\dot{q} + C(q)\dot{q}^2 + g(q)]\} \quad (52)$$

and K is introduced as

$$K = \{\tau - [B(q)\dot{q}\dot{q} + C(q)\dot{q}^2 + g(q)]\} \quad (53)$$

$\ddot{q}$ can be written as

$$\ddot{q} = M^{-1}(q).K \quad (54)$$

Therefore K for PUMA robot manipulator is calculated by the following equations

$$K_1 = \tau_1 - [\, b_{112}\dot{q}_1\dot{q}_2 + b_{113}\dot{q}_1\dot{q}_3 + 0 + b_{123}\,\dot{q}_2\dot{q}_3] - [\, C_{12}\,\dot{q}_2{}^2 + C_{13}\,\dot{q}_3{}^2] - g_1 \quad (55)$$

$$K_2 = \tau_2 - [\, b_{223}\,\dot{q}_2\dot{q}_3] - [\, C_{21}\dot{q}_1{}^2 + C_{23}\,\dot{q}_3{}^2] - g_2 \quad (56)$$

$$K_3 = \tau_3 - [C_{31}\dot{q}_1{}^2 + C_{32}\,\dot{q}_2{}^2] - g_3 \quad (57)$$

$$K_4 = \tau_4 - [\, b_{412}\,\dot{q}_1\dot{q}_2 + b_{413}\dot{q}_1\dot{q}_3] - g_4 \quad (58)$$

$$K_5 = \tau_5 - [\, C_{51}\dot{q}_1{}^2 + C_{52}\,\dot{q}_2{}^2] - g_5 \quad (59)$$

$$K_6 = \tau_6 \quad (60)$$

An information about inertial constant and gravitational constant are shown in Tables 1 and 2 based on [78-79].

TABLE 1: Inertial constant reference (*Kg.m*^{2})

$I_1 = 1.43 \pm 0.05$	$I_2 = 1.75 \pm 0.07$
$I_3 = 1.38 \pm 0.05$	$I_4 = 0.69 \pm 0.02$
$I_5 = 0.372 \pm 0.031$	$I_6 = 0.333 \pm 0.016$
$I_7 = 0.298 \pm 0.029$	$I_8 = -0.134 \pm 0.014$
$I_9 = 0.0238 \pm 0.012$	$I_{10} = -0.0213 \pm 0.0022$
$I_{11} = -0.0142 \pm 0.0070$	$I_{12} = -0.011 \pm 0.0011$
$I_{13} = -0.00379 \pm 0.0009$	$I_{14} = 0.00164 \pm 0.000070$
$I_{15} = 0.00125 \pm 0.0003$	$I_{16} = 0.00124 \pm 0.0003$
$I_{17} = 0.000642 \pm 0.0003$	$I_{18} = 0.000431 \pm 0.00013$
$I_{19} = 0.0003 \pm 0.0014$	$I_{20} = -0.000202 \pm 0.0008$
$I_{21} = -0.0001 \pm 0.0006$	$I_{22} = -0.000058 \pm 0.00001$
$I_{23} = 0.00004 \pm 0.00002$	$I_{m1} = 1.14 \pm 0.27$
$I_{m2} = 4.71 \pm 0.54$	$I_{m3} = 0.827 \pm 0.093$
$I_{m4} = 0.2 \pm 0.016$	$I_{m5} = 0.179 \pm 0.014$
$I_{m6} = 0.193 \pm 0.016$	

TABLE 2: Gravitational constant (*N.m*)

$g_1 = -37.2 \pm 0.5$	$g_2 = -8.44 \pm 0.20$
$g_3 = 1.02 \pm 0.50$	$g_4 = 0.249 \pm 0.025$
$g_5 = -0.0282 \pm 0.0056$	

Second part is focused on sliding mode formulation and its challenge. We define the tracking error as

$$e = q - q_d \quad \textbf{(61)}$$

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

$$s = \dot{e} + \lambda e \quad \textbf{(62)}$$

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 (63). We define the reference state as

$$\frac{1}{2}\frac{d}{dt}s^2(x,t) = \dot{S} \cdot S = [f - \hat{f} - K\,\mathrm{sgn}(s)] \cdot S = (f - \hat{f}) \cdot S - K|S| \quad \textbf{(63)}$$

$$\dot{q}_e = \dot{q} - s = \dot{q}_d - \lambda e \quad \textbf{(64)}$$

Now we pick the control input τ as

$$\tau = M^{\wedge}\ddot{q}_r + C^{\wedge}\dot{q}_r + \hat{B}[\dot{q}\dot{q}] + \hat{G} - As - Ksgn(s) \quad (65)$$

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 (61) and (65), we can get

$$M\dot{s} + (C + B + G + A)s = \Delta f - Ksgn(s) \quad (66)$$

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

$$K_i \geq |\Delta f_i|_{bound} \quad (67)$$

We pick the Lyapunov function candidate to be

$$V = \frac{1}{2}s^T Ms \quad (68)$$

Which is a skew-systemmetric matrix satisfying

$$s^T\left(\dot{M} - 2(C + B + G + A)\right)s = 0 \quad (69)$$

Then $\dot{V}$ becomes

$$\begin{aligned} \dot{V} &= s^T M\dot{s} + \frac{1}{2}s^T\dot{M}s \\ &= s^T(M\dot{s} + (C + B + G + A)s) \\ &= s^T[-As + \Delta f - Ksgn(s)] \\ &= \sum_{i=1}^{N}(s_i[\Delta f_i - K_i sgn(s_i)]) - s^T As \end{aligned} \quad (70)$$

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

$$\dot{V} = \sum_{i=1}^{N}(s_i[\Delta f_i - K_i sgn(s_i)]) - s^T As \leq -s^T As < 0 \quad (s \neq 0) \quad (71)$$

To attenuate chattering problem, we introduce a saturation function in the control law instead of the sign function in (63). The control law becomes

$$\tau = M^{\wedge}\ddot{q}_r + C^{\wedge}\dot{q}_r + \hat{B}[\dot{q}\dot{q}] + \hat{G} - As - Ksat(s/\Phi) \quad (72)$$

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 (72). Our control law changes to

$$\tau = M^{\wedge}\ddot{q}_r + C^{\wedge}\dot{q}_r + \hat{B}[\dot{q}\dot{q}] + \hat{G} - As - Ksat(s) \quad (73)$$

The control law for a multi degrees of freedom robot manipulator is written as:

$$U = U_{eq} + U_r \quad (74)$$

Where, the model-based component U_{eq} is the nominal dynamics of systems and it can be calculate as follows:

$$U_{eq} = \left[M^{-1}(B + C + G) + \dot{S}\right]M \quad (75)$$

Where $M(q)$ is an inertia matrix which it is symmetric and positive, $V(q,\dot{q}) = B + C$ is the vector of nonlinearity term and $G(q)$ is the vector of gravity force and U_r with minimum chattering based on [5-11] is computed as;

$$U_r = K \cdot (\mathrm{mu} + \mathrm{b})\left({}^{S}/_{\emptyset}\right) \quad (76)$$

Where $\emptyset_u = mu + b = saturation_{fuction}$ is a dead zone (saturation) function and, u and b are unlimited coefficient, by replace the formulation (76) in (74) the control output can be written as;

$$U = U_{eq} + K.(\mathrm{mu} + \mathrm{b})\left(S/_{\emptyset}\right) = \begin{cases} U_{eq} + K.\mathrm{sgn}(S) & ,|S| \geq \emptyset \\ U_{eq} + K.S/_{\emptyset} & ,|S| < \emptyset \end{cases} \quad \textbf{(77)}$$

Where the function of $sgn(S)$ defined as;

$$sgn(s) = \begin{cases} 1 & s > 0 \\ -1 & s < 0 \\ 0 & s = 0 \end{cases} \quad \textbf{(78)}$$

The main goal is to design a FPGA based adaptive mathematical model free MIMO fuzzy estimator sliding mode controller. Based on above robot manipulator has nonlinear and highly uncertain parameters consequently; following objectives have been pursuit in this paper.

- To develop a chattering in a position pure sliding mode controller against uncertainties via linear boundary layer method.
- To design and implement a MIMO fuzzy estimator sliding mode controller in order to solve the equivalent problems in the pure sliding mode control with minimum rule base based on Lyapunov formulation.
- To develop a mathematical model free adaptive fuzzy estimator sliding mode controller in order to solve the disturbance rejection and reduce the fuzzy rule base.
- To design and implement a FPGA based mathematical model free adaptive fuzzy estimator sliding mode controller

Figure 1 is shown the classical sliding mode methodology with linear saturation function to eliminate the chattering.

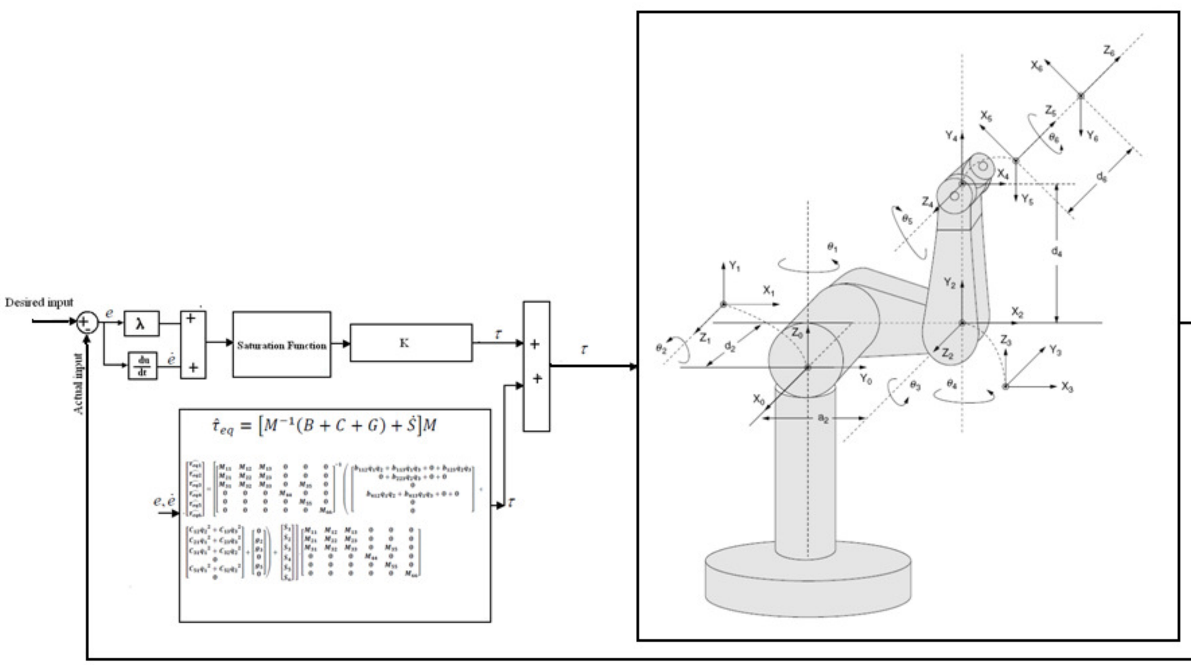

FIGURE 1: Classical sliding mode controller: applied to 6-link robotic manipulator

Zadeh 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 our systems. Fuzzy controllers including fuzzy if-then rules are used to control our 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
- Output defuzzification (fuzzy-to-binary [F/B] conversion) [30-40].

The basic structure of a fuzzy controller is shown in Figure 2.

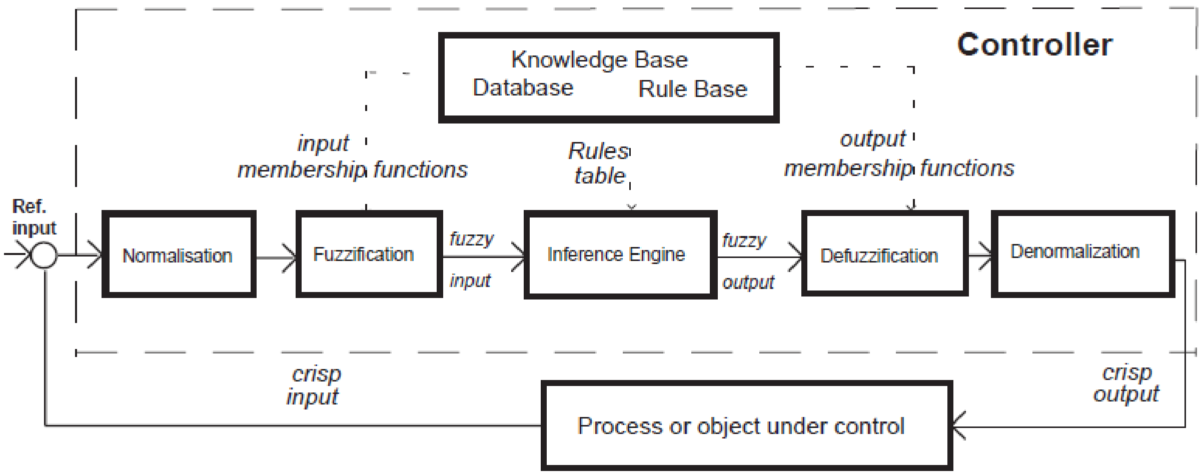

FIGURE 2: Block diagram of a fuzzy controller with details.

Conventional control methods use mathematical models to controls systems. 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, 35].

3. METHODOLOGY: DESIGN A NOVEL FPGA BASED MIMO ADAPTIVE MATHEMATICAL MODEL FREE LYAPUNOV BASED FUZZY ESTIMATE SLIDING MODE CONTROL

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. 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. 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.

We define fuzzy systems as two different types. The firs type of fuzzy systems is given by

$$f(x) = \sum_{l=1}^{M} \theta^l \mathcal{E}^l(x) = \theta^T \mathcal{E}(x) \tag{79}$$

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 (18) . $\mu_{A_1^1}(x_1), \dots, \mu_{A_n^m}(x_n)$ are given membership functions whose parameters will not change over time.

The second type of fuzzy systems is given by

$$f(x) = \frac{\sum_{l=1}^{M} \theta^l \left[\prod_{i=1}^{n} \exp\left(-\left(\frac{x_i - \alpha_i^l}{\delta_i^l}\right)^2\right)\right]}{\sum_{l=1}^{M} \left[\prod_{i=1}^{n} \exp\left(-\left(\frac{x_i - \alpha_i^l}{\delta_i^l}\right)^2\right)\right]} \tag{80}$$

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 (79). We define $f^{\wedge}(x|\theta)$ as the approximator of the real function $f(x)$.

$$f^{\wedge}(x|\theta) = \theta^T \varepsilon(x) \tag{81}$$

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

$$\theta^* = \arg\min_{\theta \in \Omega} \left[\sup_{x \in U} |f^{\wedge}(x|\theta) - g(x)|\right] \tag{82}$$

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 (79) to estimate the nonlinear system (75) the fuzzy formulation can be write as below;

$$f(x|\theta) = \theta^T \varepsilon(x) = \frac{\sum_{l=1}^{n} \theta^l [\mu_{A^l}(x)]}{\sum_{l=1}^{n} [\mu_{A^l}(x)]} \tag{83}$$

Where $\theta^1, \dots, \theta^n$ are adjusted by an adaptation law. The adaptation law is designed to minimize the parameter errors of $\theta - \theta^*$. A MIMO (multi-input multi-output) fuzzy system is designed to compensate the uncertainties of the robotic manipulator. The parameters of the fuzzy system are adjusted by adaptation laws. The tracking error and the sliding surface state are defined as (58-62)

$$e = q - q_d \tag{84}$$

$$s = \dot{e} + \lambda_e \tag{85}$$

We define the reference state as

$$\dot{q}_r = \dot{q} - s = \dot{q}_d - \lambda e \tag{86}$$

$$\ddot{q}_r = \ddot{q} - \dot{s} = \ddot{q}_d - \lambda \dot{e} \tag{87}$$

The general MIMO if-then rules are given by

$$R^l: if\ x_1\ is\ A_1^l\ , x_2\ is\ A_2^l\ , \dots, x_n\ is\ A_n^l\ , then\ y_1 is\ B_1^l\ , \dots, y_m\ is\ B_m^l \tag{88}$$

Where $l = 1,2, \dots, M$ are fuzzy if-then rules; $x = (x_1, \dots, x_n)^T$ and $y = (y_1, \dots, y_n)^T$ are the input and output vectors of the fuzzy system. The MIMO fuzzy system is define as

$$f(x) = \ominus^T \varepsilon(x) \tag{89}$$

Where

$$\ominus^T = (\theta_1, \dots, \theta_m)^T = \begin{bmatrix} \theta_1^1, \theta_1^2, \dots, \theta_1^M \\ \theta_2^1, \theta_2^2, \dots, \theta_2^M \\ \vdots \\ \theta_m^1, \theta_m^2, \dots, \theta_m^M \end{bmatrix} \tag{90}$$

$\varepsilon(x) = (\varepsilon^1(x), \dots, \varepsilon^M(x))^T$, $\varepsilon^1(x) = \prod_{i=1}^{n} \mu_{A_i^l}(x_i) / \sum_{i=1}^{M} (\prod_{i=1}^{n} \mu_{A_i^l}(x_i))$, and $\mu_{A_i^l}(x_i)$ is defined in (82). To reduce the number of fuzzy rules, we divide the fuzzy system in to three parts:

$$F^1(q,\dot{q}) = \ominus^{1^T} \varepsilon(q,\dot{q}) = \left[\theta_1^{1^T} \varepsilon(q,\dot{q}), \dots, \theta_m^{1^T} \varepsilon(q,\dot{q})\right]^T \tag{91}$$

$$F^2(q,\ddot{q}_r) = \ominus^{2^T} \varepsilon(q,\ddot{q}_r) = \left[\theta_1^{2^T} \varepsilon(q,\ddot{q}_r), \dots, \theta_m^{2^T} \varepsilon(q,\ddot{q}_r)\right]^T \tag{92}$$

$$F^3(q,\ddot{q}) = \ominus^{3^T} \varepsilon(q,\ddot{q}) = \left[\theta_1^{3^T} \varepsilon(q,\dot{q}), \dots, \theta_m^{3^T} \varepsilon(q,\ddot{q})\right]^T \tag{93}$$

The control input is given by

$$\tau = M^{\wedge}\ddot{q}_r + C_1^{\wedge}\dot{q}_r + G^{\wedge} + F^1(q,\dot{q}) + F^2(q,\ddot{q}_r) + F^3(q,\ddot{q}) - K_D s - W sgn(s) \tag{94}$$

Where $M^{\wedge}$, $C_1^{\wedge}$ are the estimations of $M(q)$ and $C_1(q,\dot{q})$; $K_D = diag\,[K_{D1}, \dots, K_{Dm}]$ and $K_{D1}, \dots, K_{Dm}$ are positive constants; $W = diag\,[W_1, \dots, W_m]$ and $W_1, \dots, W_m$ are positive constants. The adaptation law is given by

$$\begin{aligned} \dot{\theta}_j^1 &= -\Gamma_{1j} s_j \varepsilon(q,\dot{q}) \\ \dot{\theta}_j^2 &= -\Gamma_{2j} s_j \varepsilon(q,\ddot{q}_r) \\ \dot{\theta}_j^3 &= -\Gamma_{3j} s_j \varepsilon(q,\ddot{q}) \end{aligned} \tag{95}$$

Where $j = 1, \dots, m$ and $\Gamma_{1j} - \Gamma_{3j}$ are positive diagonal matrices.
The Lyapunov function candidate is presented as

$$V = \frac{1}{2}s^T M s + \frac{1}{2}\sum_{j=1}^{m} \frac{1}{\Gamma_{1j}} \emptyset_j^{1^T} \emptyset_j^1 + \frac{1}{2}\sum_{j=1}^{m} \frac{1}{\Gamma_{2j}} \emptyset_j^{2^T} \emptyset_j^2 + \frac{1}{2}\sum_{j=1}^{m} \frac{1}{\Gamma_{3j}} \emptyset_j^{13^T} \emptyset_j^3 \tag{96}$$

Where $\emptyset_j^1 = \emptyset_j^{1^*} - \emptyset_j^1, \emptyset_j^2 = \emptyset_j^{2^*} - \emptyset_j^2$ and $\emptyset_j^3 = \emptyset_j^{3^*} - \emptyset_j^3$ we define

$$F(q,\dot{q},\ddot{q}_r,\ddot{q}) = F^1(q,\dot{q}) + F^2(q,\ddot{q}_r) + F^3(q,\ddot{q}) \tag{97}$$

From (83) and (82), we get

$$M(q)\ddot{q} + C_1(q,\dot{q})\dot{q} + G(q) = M^{\wedge}\ddot{q}_r + C_1^{\wedge}\dot{q}_r + G^{\wedge} + F(q,\dot{q},\ddot{q}_r,\ddot{q}) - K_D s - W sgn(s) \tag{98}$$

Since $\dot{q}_r = \dot{q} - s$ and $\ddot{q}_r = \ddot{q} - \dot{s}$, we get

$$M\dot{s} + (C_1 + K_D)s + W sgn(s) = -\Delta F + F(q,\dot{q},\ddot{q}_r,\ddot{q}) \tag{99}$$

Then $M\dot{s} + C_1 s$ can be written as

$$M\dot{s} + C_1 s = -\Delta F + F(q, \dot{q}, \ddot{q}_r, \ddot{q}) - K_D s - W sgn(s) \qquad \textbf{(100)}$$

Where $\Delta F = \tilde{M}\ddot{q}_r + \tilde{C}_1\dot{q}_r + \tilde{G}$,$\tilde{M} = M - M^{\wedge}$,$\tilde{C}_1 = C_1 - \hat{C}_1$ and $\tilde{G} = G - G^{\wedge}$.
The derivative of V is

$$\dot{V} = s^T M\dot{s} + \frac{1}{2}s^T \dot{M}s + \sum_{j=1}^{m}\frac{1}{\Gamma_{1j}}\emptyset_j^{1^T}\dot{\emptyset}_j^1 + \sum_{j=1}^{m}\frac{1}{\Gamma_{2j}}\emptyset_j^{2^T}\dot{\emptyset}_j^2 + \sum_{j=1}^{m}\frac{1}{\Gamma_{3j}}\emptyset_j^{13^T}\dot{\emptyset}_j^3 \qquad \textbf{(101)}$$

We know that $s^T M\dot{s} + \frac{1}{2}s^T \dot{M}s = s^T(M\dot{s} + C_1 s)$ from (100). Then

$$\dot{V} = -s^T[-K_D s + W sgn(s) + \Delta F - F(q, \dot{q}, \ddot{q}_r, \ddot{q})] + \sum_{j=1}^{m}\frac{1}{\Gamma_{1j}}\emptyset_j^{1^T}\dot{\emptyset}_j^1 + \sum_{j=1}^{m}\frac{1}{\Gamma_{2j}}\emptyset_j^{2^T}\dot{\emptyset}_j^2 + \sum_{j=1}^{m}\frac{1}{\Gamma_{3j}}\emptyset \qquad \textbf{(102)}$$

We define the minimum approximation error as

$$\omega = \Delta F - [F^1(q, \dot{q} | \ominus^{1^*}) + F^2(q, \ddot{q}_r | \ominus^{2^*}) + F^3(q, \ddot{q} | \ominus^{3^*})] \qquad \textbf{(103)}$$

We plug (103) in to (102)

$$\dot{V} = -s^T[-K_D s + W sgn(s) + \Delta F - F(q, \dot{q}, \ddot{q}_r, \ddot{q})] + \sum_{j=1}^{m}\frac{1}{\Gamma_{1j}}\emptyset_j^{1^T}\dot{\emptyset}_j^1 + \sum_{j=1}^{m}\frac{1}{\Gamma_{2j}}\emptyset_j^{2^T}\dot{\emptyset}_j^2 + \sum_{j=1}^{m}\frac{1}{\Gamma_{3j}}\emptyset_j^{13^T}\dot{\emptyset}_j^3$$

$$= -s^T[-K_D s + W sgn(s) + \omega + F^1(q, \dot{q} | \ominus^{1^*}) + F^2(q, \ddot{q}_r | \ominus^{2^*}) + F^3(q, \ddot{q} | \ominus^{3^*}) - F^1(q, \dot{q}) + F^2(q, \ddot{q}_r) + F^3(q, \ddot{q})] + \sum_{j=1}^{m}\frac{1}{\Gamma_{1j}}\emptyset_j^{1^T}\dot{\emptyset}_j^1 + \sum_{j=1}^{m}\frac{1}{\Gamma_{2j}}\emptyset_j^{2^T}\dot{\emptyset}_j^2 + \sum_{j=1}^{m}\frac{1}{\Gamma_{3j}}\emptyset_j^{13^T}\dot{\emptyset}_j^3$$

$$= -s^T K_D s - s^T W sgn(s) - s^T\omega - \sum_{j=1}^{m} s_j \emptyset_j^{1^T}\varepsilon(q, \dot{q}) - \sum_{j=1}^{m} s_j \emptyset_j^{2^T}\varepsilon(q, \ddot{q}_r) - \sum_{j=1}^{m} s_j \emptyset_j^{3^T}\varepsilon(q, \ddot{q}) + \sum_{j=1}^{m}\frac{1}{\Gamma_{1j}}\emptyset_j^{1^T}\dot{\emptyset}_j^1 + \sum_{j=1}^{m}\frac{1}{\Gamma_{2j}}\emptyset_j^{2^T}\dot{\emptyset}_j^2 + \sum_{j=1}^{m}\frac{1}{\Gamma_{3j}}\emptyset_j^{3^T}\dot{\emptyset}_j^3$$

$$= -s^T K_D s - s^T W sgn(s) - s^T\omega - \sum_{j=1}^{m}\emptyset_j^{1^T}(s_j\varepsilon(q, \dot{q}) - \frac{1}{\Gamma_{1j}}\dot{\emptyset}_j^1) - \sum_{j=1}^{m}\emptyset_j^{2^T}(s_j\varepsilon(q, \ddot{q}_r) - \frac{1}{\Gamma_{2j}}\dot{\emptyset}_j^2) - \sum_{j=1}^{m}\emptyset_j^{3^T}(s_j\varepsilon(q, \ddot{q}_r) - \frac{1}{\Gamma_{3j}}\dot{\emptyset}_j^3)$$

$$= -s^T K_D s - s^T W sgn(s) - s^T\omega - \sum_{j=1}^{m}\emptyset_j^{1^T}(s_j\varepsilon(q, \dot{q}) + \frac{1}{\Gamma_{1j}}\dot{\emptyset}_j^1) - \sum_{j=1}^{m}\emptyset_j^{2^T}(s_j\varepsilon(q, \ddot{q}_r) + \frac{1}{\Gamma_{2j}}\dot{\emptyset}_j^2) - \sum_{j=1}^{m}\emptyset_j^{3^T}(s_j\varepsilon(q, \ddot{q}_r) + \frac{1}{\Gamma_{3j}}\dot{\emptyset}_j^3)$$

This section focuses on, self tuning gain updating factor for sliding function in SMC, namely, sliding surface slope (λ). The block diagram for this method is shown in Figure 2. In this controller the actual sliding surface gain (λ) is obtained by multiplying the sliding surface with gain updating factor (α). The gain updating factor (α) is calculated on-line by fuzzy dynamic model independent which has sliding surface (S) as its inputs. The gain updating factor is independent of any dynamic model of robotic manipulator parameters. It is a basic fact that the system performance in SMC is sensitive to gain updating factor, λ. Thus, determination of an optimum λ value for a system is an important problem. If the system parameters are unknown or uncertain, the problem becomes more highlighted. This problem is solved by adjusting the sliding function of the sliding mode controller continuously in real-time. In this way, the performance of the overall system is improved with respect to the classical sliding mode controller. Gain tuning-SMC has strong resistance and solves the uncertainty problems. In this controller the actual sliding function (λ_{new}) is obtained by multiplying the old sliding function (λ_{old}) with the output of supervisory mathematical free model controller(α).

$$\lambda^{new} = \lambda^{old} \times \alpha \qquad \textbf{(104)}$$

Tuning FPGA based SMC method can tune automatically the scale parameters using new method. To keep the structure of the controller as simple as possible and to avoid heavy computation, a mathematical supervisor tuner is selected [13-14]. In this method the tuneable controller tunes the input scaling factors using gain updating factors. In this method the sliding function, λ, is updated by a new coefficient factor, α, Where α is a function of system error. Figure 3 is shown the proposed method.

$$\alpha = e^2 - \frac{(r_v - r_{vmin})^2}{1+|e|} + r_{vmin} \tag{105}$$

$$r_v = \frac{\dot{e}(t) - e'(t-1)}{\dot{e}(0)}$$
$$if\ e'(0) = \begin{cases} \dot{e}(t) & if\ \ e'(t) \geq \dot{e}(t-1) \\ \dot{e}(t-1) & if\ \ e'(t) < \dot{e}(t-1) \end{cases} \tag{106}$$

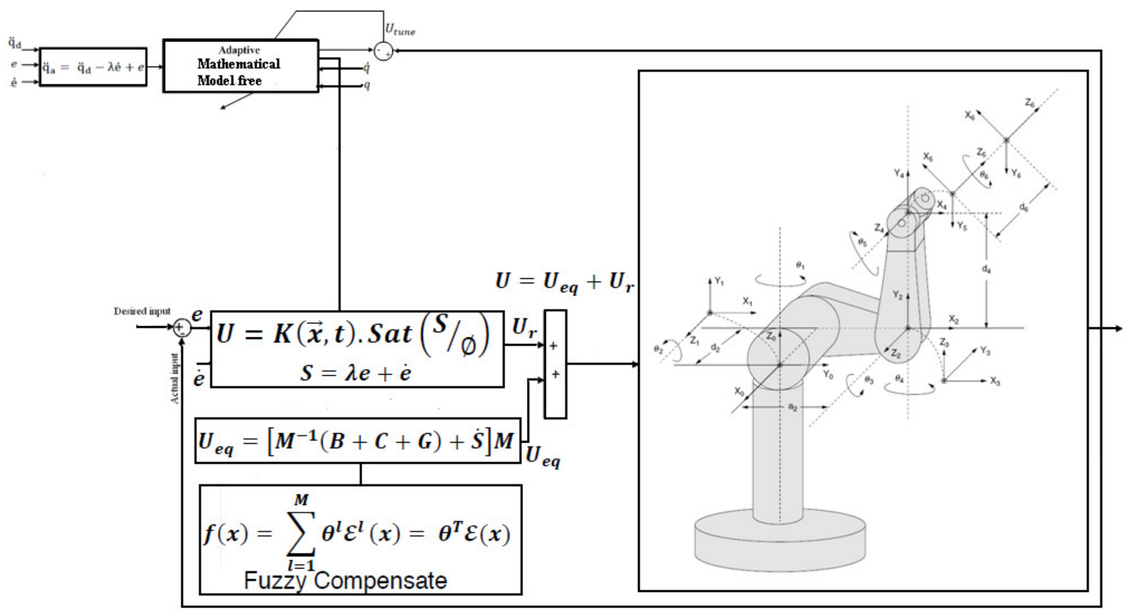

FIGURE 3: Adaptive MIMO Fuzzy Compensate Fuzzy Sliding Mode Algorithm

FPGA supports thousands of gates, it is a high operational speed, accurate in response, low cost, short time to market and small size device, research on FPGA is considerably growing as the application of nonlinear (e.g., robotic) systems. The block diagram and part of VHDL code of the FPGA-based sliding mode control systems for a robot manipulator is shown in Figure 4.

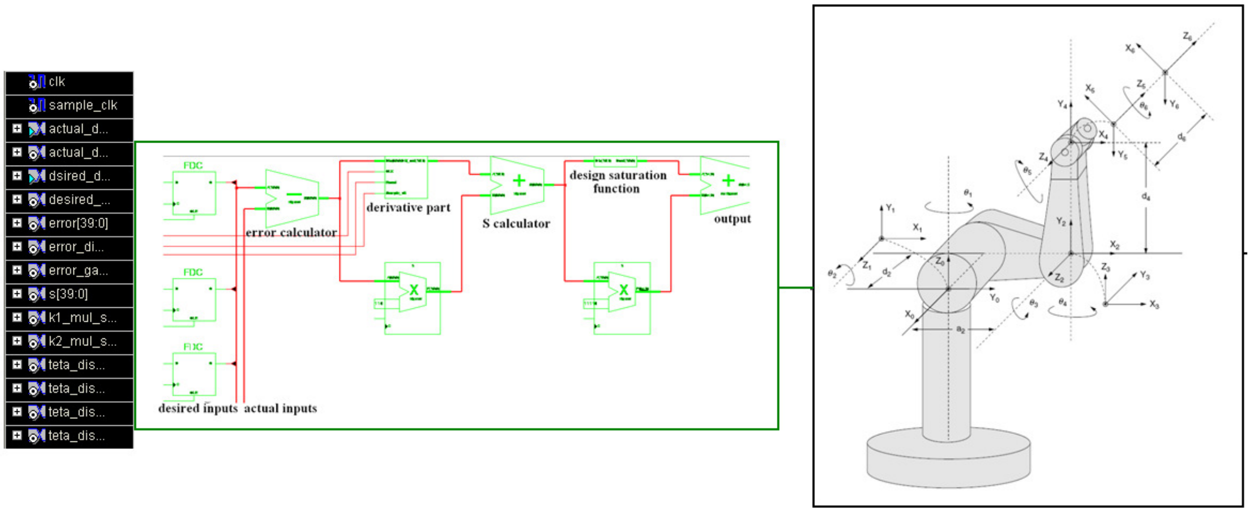

FIGURE 4: FPGA-based Adaptive MIMO Fuzzy Compensate Sliding Mode Algorithm

FPGAs Xilinx Spartan 3E families are one of the most powerful flexible Hardware Language Description (HDL) programmable IC's. The last part is focused on the design FPGA based sliding mode controller in Xilinx ISE 9.1. As a result the number of fundamental programmable functional element are used in the XA3S1600E FPGA equal: the LUT's (610 out of 29504), CLB (77 out of 3688), Slice (305 out of 14752), Multipliers (27 out of 36), registers (397), Block RAM memory (648 K) and as a Map report Peak memory usage is 175 MB.

4. RESULTS AND DISCUSSION

Sliding mode controller (SMC) and adaptive MIMO fuzzy compensate SMC and FPGA-based adaptive MIMO fuzzy compensate were tested to Step response trajectory. In this simulation 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 and Xilinx-ISE 9.1 environments. It is noted that, these systems are tested by band limited white noise with a predefined 40% of relative to the input signal amplitude which the sample time is equal to 0.1. This type of noise is used to external disturbance in continuous and hybrid systems.

Tracking performances: Figure 5 is shown the tracking performance in SMC and adaptive MIMO fuzzy compensate SMC without disturbance for Step trajectories. The best possible coefficients in Step SMC are; $K_p = K_v = K_i = 30,\ \phi_1 = \phi_2 = \phi_3 = 0.1, and\ \lambda_1 = 3, \lambda_2 = 6, \lambda_3 = 6$. From the simulation for first, second, and third links, different controller gains have the different result. Tuning parameters of SMC and adaptive MIMO fuzzy compensate SMC for this type trajectories in PUMA robot manipulator are shown in Table 3.

TABLE 3: Tuning parameters of Step SMC

	λ_1	k_1	ϕ_1	λ_2	k_2	ϕ_2	λ_3	k_3	ϕ_3	SS error $_1$	SS error $_2$	SS error $_3$	RMS error
data1	3	30	0.1	6	30	0.1	6	30	0.1	0	0	-5.3e-15	0
data2	30	30	0.1	60	30	0.1	60	30	0.1	-5.17	14.27	-1.142	0.05
data3	3	300	0.1	6	300	0.1	6	300	0.1	2.28	0.97	0.076	0.08

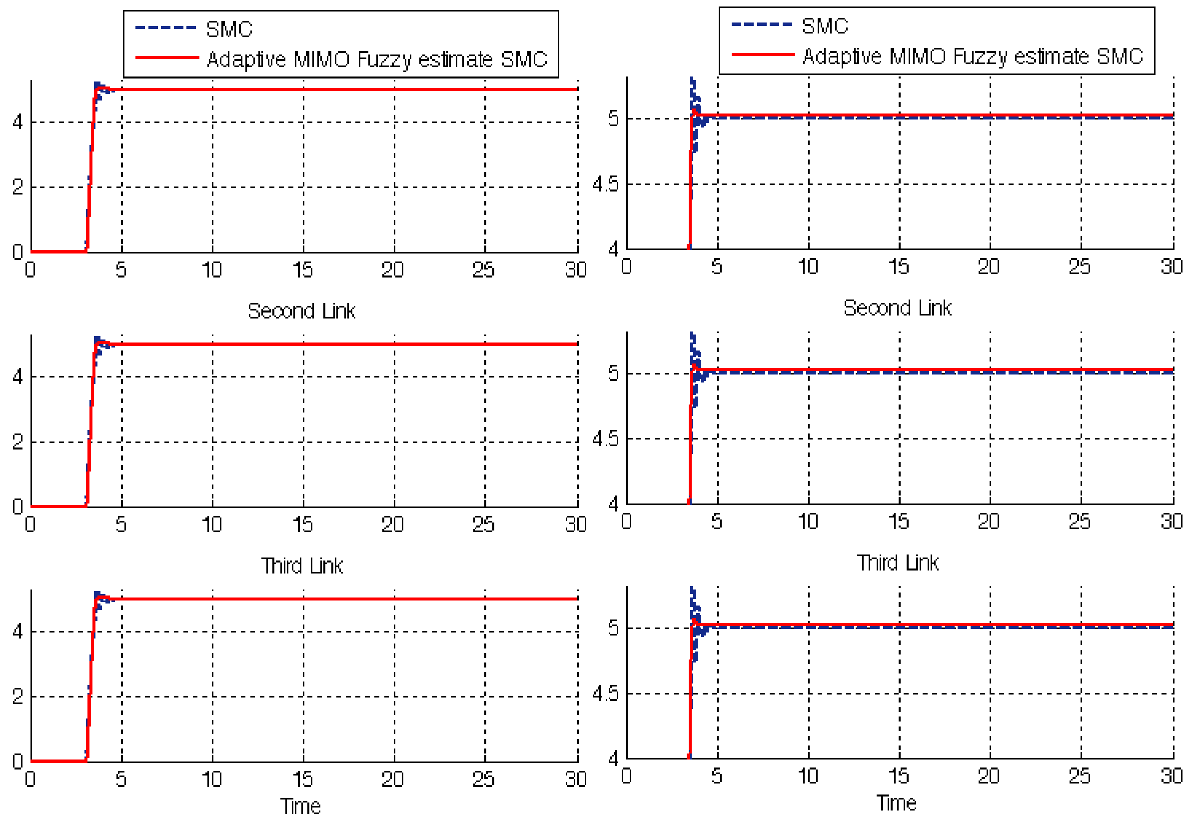

FIGURE 5: SMC Vs. Adaptive MIMO Fuzzy Compensate Sliding Mode Algorithm: Trajectory performance.

Figure 6 is shown the tracking performance in FPGA-based adaptive MIMO fuzzy compensate SMC without disturbance for Step trajectory.

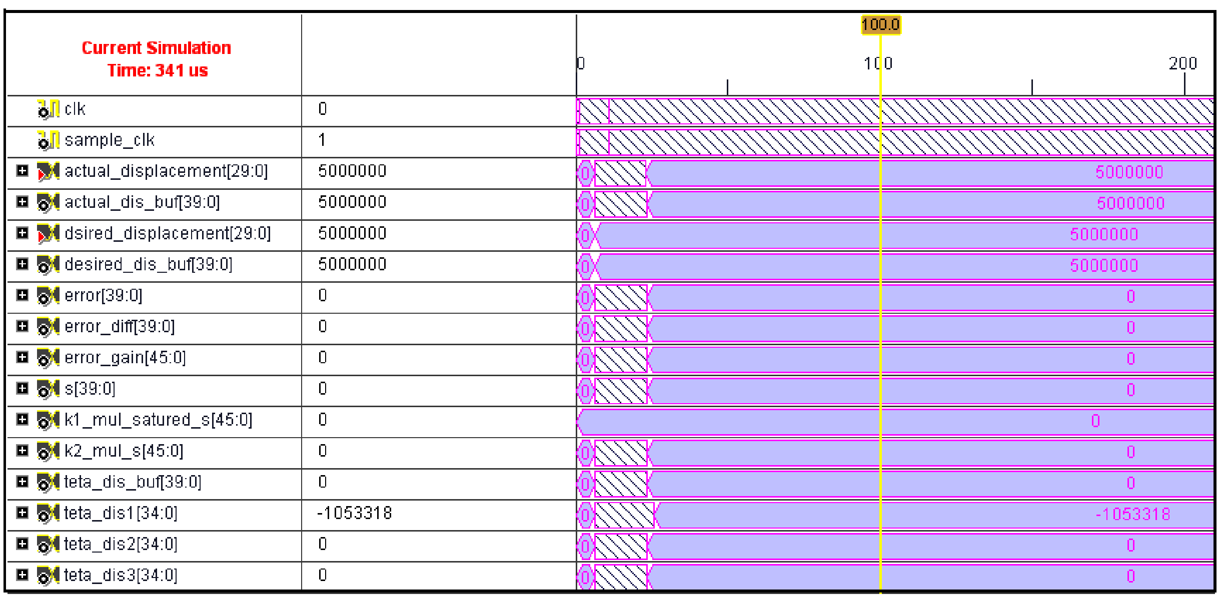

FIGURE 6: FPGA based Adaptive MIMO Fuzzy Compensate Sliding Mode Algorithm: Trajectory performance.

As mentioned above graphs (Figure 5 and 6), in certain (structured and unstructured) environment it is shown that both of sliding mode controller and adaptive MIMO fuzzy compensate sliding mode algorithm removed the chattering because these controller are used linear boundary layer saturation method.

Disturbance Rejection: Figure 7 is indicated the power disturbance removal in SMC and adaptive MIMO fuzzy compensate sliding mode algorithm. As mentioned before, SMC is one of the most important robust nonlinear controllers. Besides a band limited white noise with predefined of 40% the power of input signal is applied to the step SMC and adaptive MIMO fuzzy compensate sliding mode algorithm; it found slight oscillations in SMC trajectory responses. As a result, by comparing SMC and adaptive MIMO fuzzy compensate sliding mode algorithm, it found that adaptive MIMO fuzzy compensate sliding mode algorithm is more robust than SMC with regards to the same external disturbance.

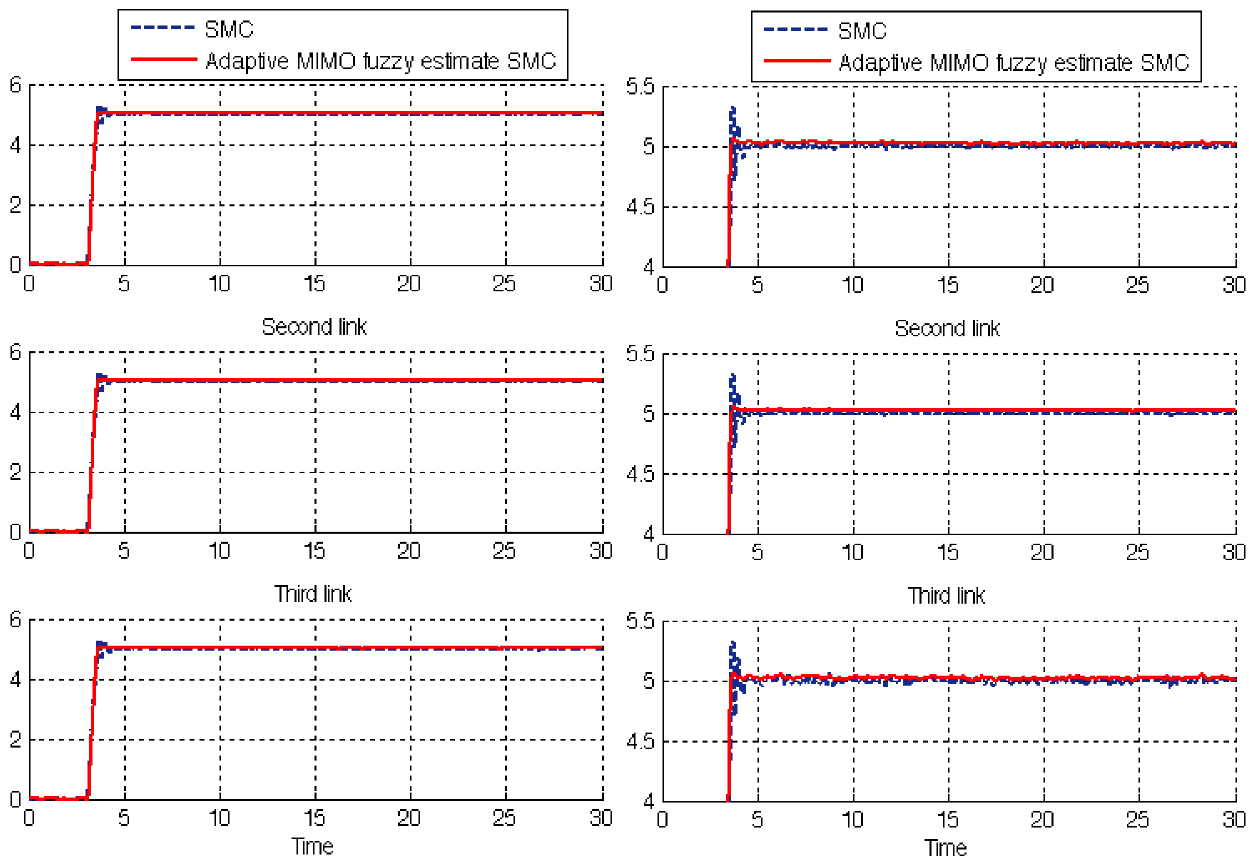

FIGURE 7: SMC Vs. Adaptive MIMO Fuzzy Compensate Sliding Mode Algorithm: Disturbance rejection.

Figure 8 is shown the tracking performance in FPGA-based adaptive MIMO fuzzy compensate SMC with external disturbance for Step trajectory.

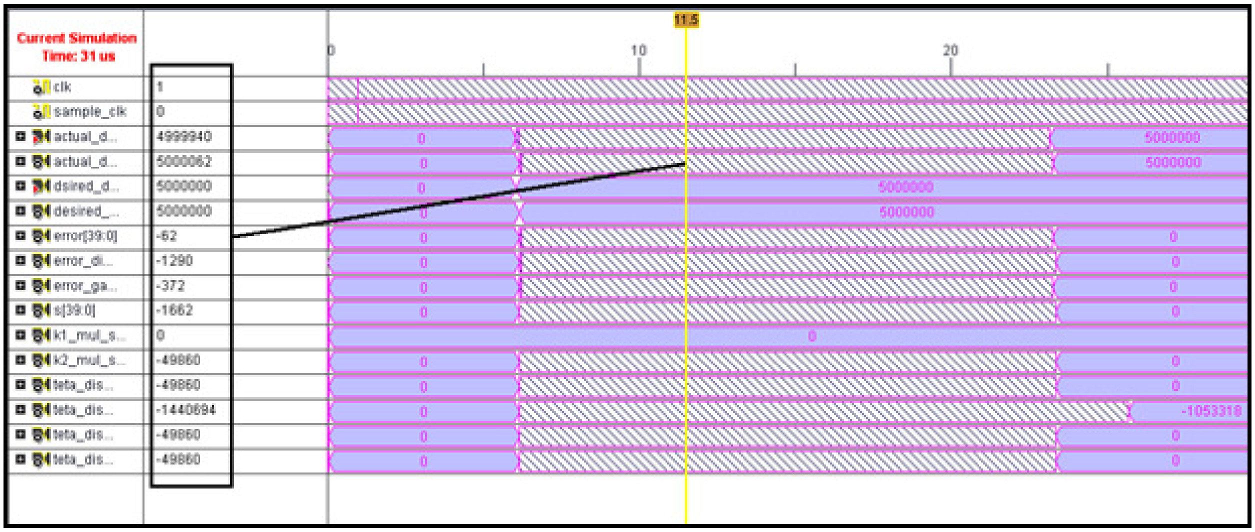

FIGURE 8: FPGA based Adaptive MIMO Fuzzy Compensate Sliding Mode Algorithm: Disturbance rejection

Chattering Phenomenon: Figure 9 has presented the power of adaptive MIMO fuzzy compensate SMC. These figures have illustrated the power chattering elimination in SMC as well as in adaptive MIMO fuzzy compensate SMC, with external disturbance. By comparing these controllers, conversely SMC has slight fluctuations; adaptive MIMO fuzzy compensate SMC is steadily stabilized. As a result, with respect to the external disturbance adaptive MIMO fuzzy compensate SMC has an acceptable performance.

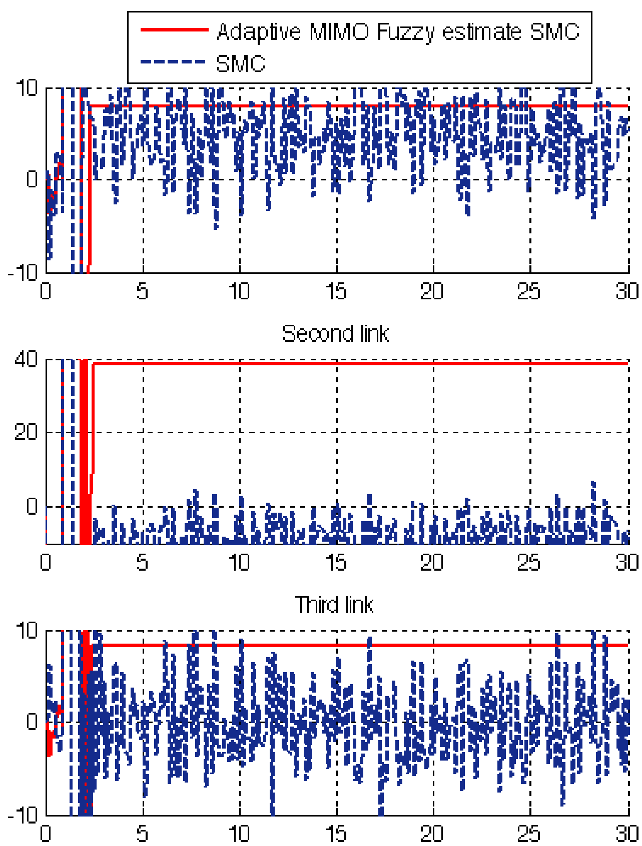

FIGURE 9: SMC Vs. Adaptive MIMO Fuzzy Compensate Sliding Mode Algorithm: Chattering.

Figure 10 is shown the chattering phenomenon in FPGA-based adaptive MIMO fuzzy compensate SMC with external disturbance for Step trajectory.

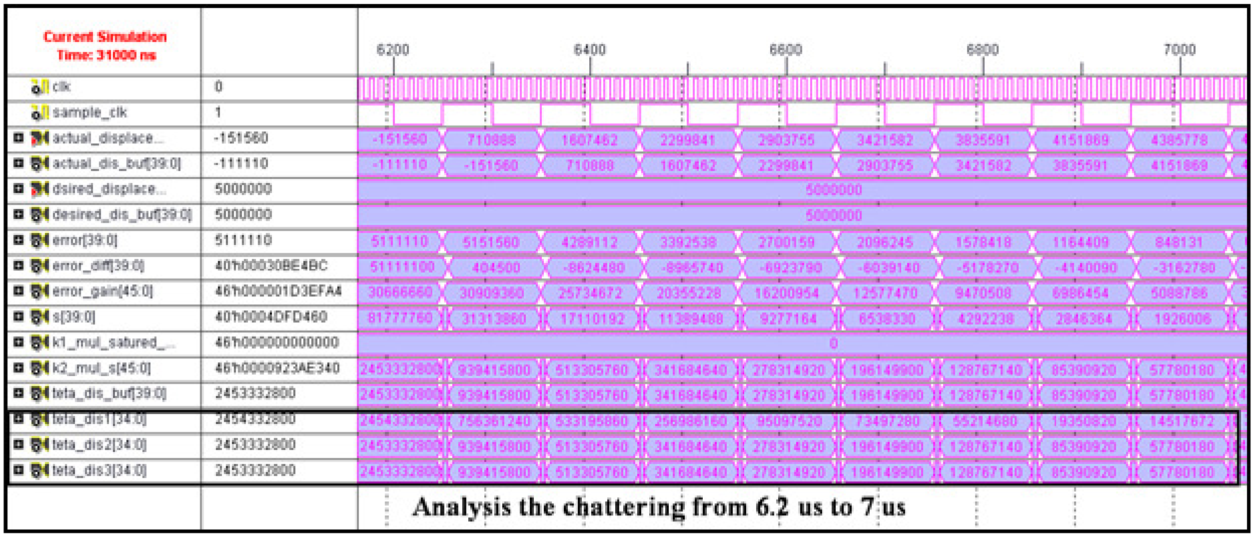

FIGURE 10: FPGA based Adaptive MIMO Fuzzy Compensate Sliding Mode Algorithm: chattering rejection

5. CONCLUSION AND EXTENSION

Refer to the research, a position FPGA-based mathematical model free adaptive fuzzy estimator sliding mode control Lyapunov based design and application to PUMA robot manipulator has proposed in order to design high performance robust and stable FPGA based nonlinear controller in the presence of structure and unstructured uncertainties. The stability of the closed-loop system is proved mathematically based on the Lyapunov method. The first objective in proposed method is removed the chattering which linear boundary layer method is used to solve this challenge. The second target in this work is compensate the model uncertainty by MIMO fuzzy inference system, in the case of the m-link robotic manipulator, if we define k_1 membership functions for each input variable, the number of fuzzy rules applied for each joint is K_1 which will result in a low computational load. The third target in this research is applied mathematical model free to MIMO fuzzy estimator sliding mode algorithm and eliminate the chattering with minimum computational load and the final main goal is design FPGA based proposed methodology which in this case the performance is improved by using the advantages of sliding mode algorithm, artificial intelligence compensate method, adaptive algorithm and FPGA while the disadvantages removed by added each method to previous method. Fuzzy logic method by adding to the sliding mode controller has covered negative points in fuzzy and sliding algorithms. Higher implementation speed and small chip size versus an acceptable performance is reached by designing FPGA-based sliding mode controller. This implementation considerably reduces the chattering phenomenon and error in the presence of certainties. The controller works with a maximum clock frequency of 63.29 MHz and the computation time (delay in activation) of this controller is 0.1μs. As a result, this controller will be able to control a wide range of robot manipulators with a high sampling rates because its small size versus high speed markets.

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,” world applied science journal (WASJ), 13 (5): 1085-1092, 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] N. Sulaiman*, et al.*, "Design and Implementation of FPGA-Based Systems-A Review," *Australian Journal of Basic and Applied Sciences,* vol. 3, pp. 3575-3596, 2009.

[64] X. Shao and D. Sun, "Development of an FPGA-based motion control ASIC for robotic manipulators," 2006, pp. 8221-8225.

[65] Y. S. Kung*, et al.*, "FPGA-implementation of inverse kinematics and servo controller for robot manipulator," *Proc. IEEE Int. on Robotics and Biomimetics,* pp. 1163–1168, 2006.

[66] X. Shao*, et al.*, "A new motion control hardware architecture with FPGA-based IC design for robotic manipulators," 2006, pp. 3520-3525.

[67] Y. S. Kung*, et al.*, "Design and Implementation of a Servo System for Robotic Manipulator," ed: CACS, 2005.

[68] U. D. Meshram and R. Harkare, "FPGA Based Five Axis Robot Arm Controller."

[69] U. Meshram*, et al.*, "Robot arm controller using FPGA," 2009, pp. 8-11.

[70] Y. S. Kung and G. S. Shu, "Development of a FPGA-based motion control IC for robot arm," 2006, pp. 1397-1402.

[71] Z. A. Obaid*, et al.*, "Developed Method of FPGA-based Fuzzy Logic Controller Design with the Aid of Conventional PID Algorithm," *Australian Journal of Basic and Applied Sciences,* vol. 3, pp. 2724-2740, 2009.

[72] S. T. Karris, *Digital circuit analysis and design with Simulink modeling and introduction to CPLDs and FPGAs*: Orchard Pubns, 2007.

[73] K. D. Rogers, "ACCELERATION AND IMPLEMENTION OF A DSP PHASE-BASED FREQUENCY ESTIMATION ALGORITHM: MATLAB/SIMULINK TO FPGA VIA XILINX SYSTEM GENERATOR," Citeseer, 2004.

[74] F. J. Lin*, et al.*, "FPGA-based fuzzy sliding-mode control for a linear induction motor drive," 2005, pp. 1137-1148.
[75] R. R. Ramos*, et al.*, "A fixed-frequency quasi-sliding control algorithm: application to power inverters design by means of FPGA implementation," *Power Electronics, IEEE Transactions on,* vol. 18, pp. 344-355, 2003.
[76] F. J. Lin*, et al.*, "FPGA-based adaptive backstepping sliding-mode control for linear induction motor drive," *Power Electronics, IEEE Transactions on,* vol. 22, pp. 1222-1231, 2007.
[77] S. Lentijo*, et al.*, "FPGA based sliding mode control for high frequency power converters," 2004, pp. 3588-3592.
[78] B. Armstrong*, et al.*, "The explicit dynamic model and inertial parameters of the PUMA 560 arm," 2002, pp. 510-518.
[79] P. I. Corke and B. Armstrong-Helouvry, "A search for consensus among model parameters reported for the PUMA 560 robot," 2002, pp. 1608-1613.