

Design Parallel Linear PD Compensation by Fuzzy Sliding Compensator for Continuum Robot

Amin Jalali

Department of Maritime Electronic and Communication Engineering, College of Maritime Engineering, Chabahar University, Iran

E-mail: Max.Jalali@gmail.com

Farzin Piltan

Research & Development Lab., Electrical and Electronic Engineering Unit, Sanatkadehe Sabze Pasargad (SSP. Co). Shiraz, Iran

E-mail: Piltan_f@iranssp.com

Mohammadreza Hashemzadeh

Department of Electrical Engineering, Fasa Branch, Islamic Azad University, Fars, Iran

E-mail: h.mohammadreza33@yahoo.com

Fatemeh BibakVaravi

Shiraz Islamic Azad University, Fars, Iran

E-mail: Fatima_bibak@yahoo.com

Hossein Hashemzadeh

Department of Information and Technology, Fars Science & Research branch, Islamic Azad University, Iran

E-mail: hossein.hshm@gmail.com

Abstract— In this paper, a linear proportional derivative (PD) controller is designed for highly nonlinear and uncertain system by using robust factorization approach. To evaluate a linear PD methodology two type methodologies are introduced; sliding mode controller and fuzzy logic methodology. This research aims to design a new methodology to fix the position in continuum robot manipulator. PD method is a linear methodology which can be used for highly nonlinear system's (e.g., continuum robot manipulator). To estimate this method, new parallel fuzzy sliding mode controller (PD.FSMC) is used. This estimator can estimate most of nonlinearity terms of dynamic parameters to achieve the best performance. The asymptotic stability of fuzzy PD control with first-order sliding mode compensation in the parallel structure is proven. For the parallel structure, the finite time convergence with a super-twisting second-order sliding-mode is guaranteed.

Index Terms— Fuzzy Logic Compensator, Continuum Robot Manipulator, Sliding Mode Control, PD Control Methodology

I. Introduction

Continuum robots represent a class of robots that have a biologically inspired form characterized by flexible backbones and high degrees-of-freedom structures [1]. The idea of creating “trunk and tentacle” robots, (in recent years termed continuum robots [1]), is not new [2]. Inspired by the bodies of animals such as snakes [3], the arms of octopi [4], and the trunks of elephants [5], [6], researchers have been building prototypes for many years. A key motivation in this research has been to reproduce in robots some of the special qualities of the biological counterparts. This includes the ability to “slither” into tight and congested spaces and (of particular interest in this work) the ability to grasp and manipulate a wide range of objects, via the use of “whole arm manipulation” i.e. wrapping their bodies around objects, conforming to their shape profiles. Hence, these robots have potential applications in whole arm grasping and manipulation in unstructured environments such as rescue operations. Theoretically, the compliant nature of a continuum robot provides infinite degrees of freedom to these devices. However, there is a limitation set by the practical inability to incorporate infinite actuators in the device. Most of these robots are consequently under actuated (in terms of numbers of independent actuators) with respect to their anticipated tasks. In other words they must achieve

a wide range of configurations with relatively few control inputs. This is partly due to the desire to keep the body structures (which, unlike in conventional rigid-link manipulators or fingers, are required to directly contact the environment) “clean and soft”, but also to exploit the extra control authority available due to the continuum contact conditions with a minimum number of actuators. For example, the Octarm VI continuum manipulator, discussed frequently in this paper, has nine independent actuated degrees-of-freedom with only three sections. Continuum manipulators differ fundamentally from rigid-link and hyper-redundant robots by having an unconventional structure that lacks links and joints. Hence, standard techniques like the Denavit-Hartenberg (D-H) algorithm cannot be directly applied for developing continuum arm kinematics. Moreover, the design of each continuum arm varies with respect to the flexible backbone present in the system, the positioning, type and number of actuators. The constraints imposed by these factors make the set of reachable configurations and nature of movements unique to every continuum robot. This makes it difficult to formulate generalized kinematic or dynamic models for continuum robot hardware. Chirikjian and Burdick were the first to introduce a method for modeling the kinematics of a continuum structure by representing the curve-shaping function using modal functions [6]. Mochiyama used the Serret-Frenet formulae to develop kinematics of hyper-degrees of freedom continuum manipulators [5]. For details on the previously developed and more manipulator-specific kinematics of the Rice/Clemson “Elephant trunk” manipulator, see [1-2], [5]. For the Air Octor and Octarm continuum robots, more general forward and inverse kinematics have been developed by incorporating the transformations of each section of the manipulator (using D-H parameters of an equivalent virtual rigid link robot) and expressing those in terms of the continuum manipulator section parameters [4]. The net result of the work in [3-6] is the establishment of a general set of kinematic algorithms for continuum robots. Thus, the kinematics (i.e. geometry based modeling) of a quite general set of prototypes of continuum manipulators has been developed and basic control strategies now exist based on these. The development of analytical models to analyze continuum arm dynamics (i.e. physics based models involving forces in addition to geometry) is an active, ongoing research topic in this field. From a practical perspective, the modeling approaches currently available in the literature prove to be very complicated and a dynamic model which could be conveniently implemented in an actual device’s real-time controller has not been developed yet. The absence of a computationally tractable dynamic model for these robots also prevents the study of interaction of external forces and the impact of collisions on these continuum structures. This impedes the study and ultimate usage of continuum robots in various practical applications like grasping and manipulation, where impulsive dynamics [1, 4] are important factors. Although continuum

robotics is an interesting subclass of robotics with promising applications for the future, from the current state of the literature, this field is still in its stages of inception.

Controller is used to sense information from linear or nonlinear system (e.g., continuum robot manipulator) to improve the systems performance [24-33]. The main targets in the design control systems are stability, good disturbance rejection, and small tracking error [5]. Linear control methodology (e.g., Proportional-Derivative (PD) controller, Proportional-Integral (PI) controller or Proportional-Integral-Derivative (PID) controller) is used to control of many nonlinear systems, but when these systems have uncertainty in dynamic models this technique has limitations. To solve this challenge nonlinear robust methodology (e.g., sliding mode controller, computed torque controller, backstepping controller and Lyapunov based methodology) is introduced. In some applications of nonlinear systems dynamic parameters are unknown or the environment is unstructured, therefore strong mathematical tools used in new control methodologies to design nonlinear robust controller with an acceptable performance (e.g., minimum error, good trajectory, disturbance rejection) [34-38]. Most of robust nonlinear controllers are work based on nonlinear dynamic equivalent part and design this controller based on these formulation is difficult. To reduce the above challenges, the nonlinear robust controller is used to estimate of continuum robot manipulator.

Fuzzy-logic aims to provide an approximate but effective means of describing the behavior of systems that are not easy to describe precisely, and which are complex or ill-defined [7-11, 22]. It is based on the assumption that, in contrast to Boolean logic, a statement can be partially true (or false) [12-21, 23-33]. For example, the expression (I live near SSP.Co) where the fuzzy value (near) applied to the fuzzy variable (distance), in addition to being imprecise, is subject to interpretation. The essence of fuzzy control is to build a model of human expert who is capable of controlling the plant without thinking in terms of its mathematical model. As opposed to conventional control approaches where the focus is on constructing a controller described by differential equations, in fuzzy control the focus is on gaining an intuitive understanding (heuristic data) of how to best control the process [28], and then load this data into the control system [34-35].

Sliding mode control (SMC) is obtained by means of injecting a nonlinear discontinuous term. This discontinuous term is the one which enables the system to reject disturbances and also some classes of mismatches between the actual system and the model used for design [12, 36-44]. These standard SMCs are robust with respect to internal and external perturbations, but they are restricted to the case in which the output relative degree is one. Besides, the high frequency switching that produces the sliding mode may cause chattering effect. The tracking error of

SMC converges to zero if its gain is bigger than the upper bound of the unknown nonlinear function. Boundary layer SMC can assure no chattering happens when tracking error is less than ϵ ; but the tracking error converges to ϵ ; it is not asymptotically stable [13]. A new generation of SMC using second-order sliding-mode has been recently developed by [15] and [16]. This higher order SMC preserves the features of the first order SMC and improves it in eliminating the chattering and fast convergence [45-47].

Normal combinations of PD control with fuzzy logic (PD+FL) and sliding mode (PD+SMC) are to apply these three controllers at the same time [17], while FLC compensates the control error, SMC reduces the remain error of fuzzy PD such that the final tracking error is asymptotically stable [18]. The chattering is eliminate, because PD+SMC and PD+FL work parallel. In this paper, the asymptotic stability of PD control with parallel fuzzy logic and the first-order sliding mode compensation is proposed (PD+SMC+FL). The fuzzy PD is used to approximate the nonlinear plant. A dead one algorithm is applied for the fuzzy PD control. After the regulation error enter converges to the dead-zone, a super-twisting second-order sliding-mode is used to guarantee finite time convergence of the whole control (PD+FL+SMC). By means of a Lyapunov approach, we prove that this type of control can ensure finite time convergence and less chattering than SMC and SMC+FL [33-47].

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

II. Theory

2.1 Continuum Robot Manipulator's Dynamic:

The Continuum section analytical model developed here consists of three modules stacked together in series. In general, the model will be a more precise replication of the behavior of a continuum arm with a greater of modules included in series. However, we will show that three modules effectively represent the dynamic behavior of the hardware, so more complex models are not motivated. Thus, the constant curvature bend exhibited by the section is incorporated inherently within the model. The mass of the arm is modeled as being concentrated at three points whose co-ordinates referenced with respect to (see Figure 1);

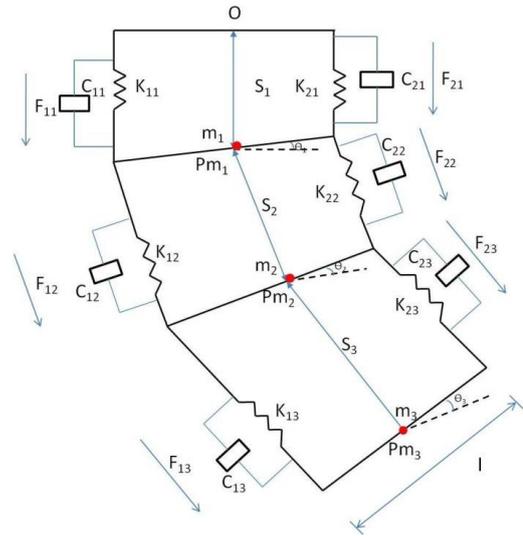


Fig. 1: Assumed structure for analytical model of a section of a continuum arm

Where;

l - Length of the rigid rod connecting the two struts, constant throughout the structure

$k_{1,i}$, $i = 1,2,3$ - Spring constant of actuator 1 at module i

$k_{2,i}$, $i = 1,2,3$ - Spring constant of actuator 2 at module i

$C_{1,i}$, $i = 1,2,3$ - Damping coefficient of actuator 1 at module i

$C_{2,i}$, $i = 1,2,3$ - Damping coefficient of actuator 2 at module i

m_i , $i = 1,2,3$ - Mass in each module

I_i , $i = 1,2,3$ - Moment of inertia of the rigid rod in each module.

A global inertial frame (N) located at the base of the arm are given below

$${}^N_{m1}P = S_1 \cdot \hat{n}_3 \quad (1)$$

$${}^N_{m2}P = S_2 \cdot \sin\theta_1 \hat{n}_1 + (S_1 + S_2 \cos\theta_1) \cdot \hat{n}_3 \quad (2)$$

$${}^N_{m3}P = (S_2 \cdot \sin\theta_1 + S_3 \cdot \sin(\theta_1 + \theta_2)) \hat{n}_1 + (S_1 + S_2 \cos\theta_1 + S_3 \cdot \cos(\theta_1 + \theta_2)) \cdot \hat{n}_3 \quad (3)$$

The position vector of each mass is initially defined in a frame local to the module in which it is present. These local frames are located at the base of each module and oriented along the direction of variation of coordinate 's' of that module. The positioning of each of these masses is at the centre of mass of the rigid rods connecting the two actuators. Differentiating the position vectors we obtain the linear velocities of the

masses. The kinetic energy (T) of the system comprises the sum of linear kinetic energy terms (constructed using the above velocities) and rotational kinetic energy terms due to rotation of the rigid rod connecting the two actuators, and is given below as

$$\begin{aligned}
 T = & (0.5)m_1\dot{s}_1^2 + (0.5)m_2 \left((\dot{s}_2 \sin\theta_1 + s_2 \cos\theta_1 \dot{\theta}_1)^2 + (\dot{s}_1 + \dot{s}_2 \cos\theta_1 - s_2 \sin\theta_1 \dot{\theta}_1)^2 \right) + (0.5)m_3 \left((\dot{s}_2 \sin\theta_1 + s_2 \cos\theta_1 \dot{\theta}_1 + \dot{s}_3 \sin(\theta_1 + \theta_2) + s_3 \cos(\theta_1 + \theta_2) \dot{\theta}_1 + s_3 \cos(\theta_1 + \theta_2) \dot{\theta}_2)^2 + (\dot{s}_1 + \dot{s}_2 \cos\theta_1 - s_2 \sin\theta_1 \dot{\theta}_1 + \dot{s}_3 \cos(\theta_1 + \theta_2) - s_3 \sin(\theta_1 + \theta_2) \dot{\theta}_1 - s_3 \sin(\theta_1 + \theta_2) \dot{\theta}_2)^2 \right) + (0.5)I_1 \dot{\theta}_1^2 + (0.5)I_2 (\dot{\theta}_1^2 + \dot{\theta}_2^2) + (0.5)I_3 (\dot{\theta}_1^2 + \dot{\theta}_2^2 + \dot{\theta}_3^2). \quad (4)
 \end{aligned}$$

The potential energy (P) of the system comprises the sum of the gravitational potential energy and the spring potential energy. A small angle assumption is made throughout the derivation. This allows us to directly express the displacement of springs and the velocities associated with dampers in terms of system generalized coordinates.

$$\begin{aligned}
 P = & -m_1 g s_1 - m_2 g (s_1 + s_2 \cos\theta_1) - m_3 g (s_1 + s_2 \cos\theta_1 + s_3 \cos(\theta_1 + \theta_1)) + (0.5)k_{11} (s_1 + (1/2)\theta_1 - s_{01})^2 + (0.5)k_{21} (s_1 + (1/2)\theta_1 - s_{01})^2 + (0.5)k_{12} (s_2 + (1/2)\theta_2 - s_{02})^2 + (0.5)k_{22} (s_2 + (1/2)\theta_2 - s_{02})^2 + (0.5)k_{13} (s_3 + (1/2)\theta_3 - s_{03})^2 + (0.5)k_{23} (s_3 + (1/2)\theta_3 - s_{03})^2 \quad (5)
 \end{aligned}$$

where, s_{01}, s_{02}, s_{03} are the initial values of s_1, s_2, s_3 respectively.

Due to viscous damping in the system, Rayleigh's dissipation function [6] is used to give damping energy

$$\begin{aligned}
 D = & (0.5)c_{11} (\dot{s}_1 + (1/2)\dot{\theta}_1)^2 + (0.5)c_{21} (\dot{s}_1 + (1/2)\dot{\theta}_1)^2 + (0.5)c_{12} (\dot{s}_2 + (1/2)\dot{\theta}_2)^2 + (0.5)c_{22} (\dot{s}_2 + (1/2)\dot{\theta}_2)^2 + (0.5)c_{13} (\dot{s}_3 + (1/2)\dot{\theta}_3)^2 + (0.5)c_{23} (\dot{s}_3 + (1/2)\dot{\theta}_3)^2. \quad (6)
 \end{aligned}$$

$F_{coeff} =$

$$\begin{bmatrix} 1 & 1 & \cos(\theta_1) & \cos(\theta_1) & \cos(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \\ 0 & 0 & 1 & 1 & \cos(\theta_2) & \cos(\theta_2) \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1/2 & -1/2 & 1/2 & -1/2 & 1/2 + s_2 \sin(\theta_2) & -1/2 + s_2 \sin(\theta_2) \\ 0 & 0 & 1/2 & -1/2 & 1/2 & -1/2 \\ 0 & 0 & 0 & 0 & 1/2 & -1/2 \end{bmatrix} \quad (14)$$

The generalized forces in the system corresponding to the generalized co-ordinates are expressed as appropriately weighted combinations of the input forces.

$$Q_{s_1} = F_{11} + F_{21} + (F_{12} + F_{22})\cos\theta_1 + (F_{13} + F_{23})\cos(\theta_1 + \theta_2) \quad (7)$$

$$Q_{s_2} = F_{12} + F_{22} + (F_{13} + F_{23})\cos(\theta_2) \quad (8)$$

$$Q_{s_3} = F_{13} + F_{23} \quad (9)$$

$$Q_{\theta_1} = (1/2)(F_{11} - F_{21}) + (1/2)(F_{12} - F_{22}) + (1/2)(F_{13} - F_{23}) + s_2 \sin\theta_2 (F_{13} + F_{23}) \quad (10)$$

$$Q_{\theta_2} = (1/2)(F_{12} - F_{22}) + (1/2)(F_{13} - F_{23}) \quad (11)$$

$$Q_{\theta_3} = (1/2)(F_{13} - F_{23}) \quad (12)$$

It can be evinced from the force expressions that the total input forces acting on each module can be resolved into an additive component along the direction of extension and a subtractive component that results in a torque. For the first module, there is an additional torque produced by forces in the third module.

The model resulting from the application of Lagrange's equations of motion obtained for this system can be represented in the form

$$F_{coeff} \underline{\tau} = D(\underline{q}) \underline{\ddot{q}} + C(\underline{q}) \underline{\dot{q}} + G(\underline{q}) \quad (13)$$

where τ is a vector of input forces and q is a vector of generalized co-ordinates. The force coefficient matrix F_{coeff} transforms the input forces to the generalized forces and torques in the system. The inertia matrix, D is composed of four block matrices. The block matrices that correspond to pure linear accelerations and pure angular accelerations in the system (on the top left and on the bottom right) are symmetric. The matrix C contains coefficients of the first order derivatives of the generalized co-ordinates. Since the system is nonlinear, many elements of C contain first order derivatives of the generalized co-ordinates. The remaining terms in the dynamic equations resulting from gravitational potential energies and spring energies are collected in the matrix G . The coefficient matrices of the dynamic equations are given below,

$$D(\mathbf{q}) = \begin{bmatrix} m_1 + m_2 + m_3 & m_2 \cos(\theta_1) + m_3 \cos(\theta_1) & m_3 \cos(\theta_1 + \theta_2) & -m_2 s_2 \sin(\theta_1) - m_3 s_2 \sin(\theta_1) - m_3 s_3 \sin(\theta_1 + \theta_2) & -m_3 s_3 \sin(\theta_1 + \theta_2) & 0 \\ m_2 \cos(\theta_1) + m_3 \cos(\theta_1) & m_2 + m_3 & m_3 \cos(\theta_2) & -m_3 s_3 \sin(\theta_2) & -m_3 s_3 \sin(\theta_2) & 0 \\ m_3 \cos(\theta_1 + \theta_2) & m_3 \cos(\theta_2) & m_3 & m_3 s_3 \sin(\theta_2) & 0 & 0 \\ -m_2 s_2 \sin(\theta_1) - m_3 s_2 \sin(\theta_1) - m_3 s_3 \sin(\theta_1 + \theta_2) & -m_3 s_3 \sin(\theta_2) & m_3 s_2 \sin(\theta_2) & m_2 s_2^2 + I_1 + I_2 + I_3 + m_3 s_2^2 + m_3 s_3^2 + 2m_3 s_3 \cos(\theta_2) s_2 & I_2 + m_3 s_3^2 + I_3 + m_3 s_3 \cos(\theta_2) s_2 & I_3 \\ -m_3 s_3 \sin(\theta_1 + \theta_2) & -m_3 s_3 \sin(\theta_2) & 0 & I_2 + m_3 s_3^2 + I_3 + m_3 s_3 \cos(\theta_2) s_2 I & I_2 + m_3 s_3^2 + I_3 & I_3 \\ 0 & 0 & 0 & I_3 & I_3 & I_3 \end{bmatrix} \quad (15)$$

$$\mathbf{c}(\mathbf{q}) = \begin{bmatrix} c_{11} + c_{21} & -2m_2 \sin(\theta_1) \dot{\theta}_1 - 2m_3 \sin(\theta_1) \dot{\theta}_1 & -2m_3 \sin(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2) & -m_2 s_2 \cos(\theta_1) (\dot{\theta}_1) + (1/2)(c_{11} + c_{21}) - m_3 s_2 \cos(\theta_1) (\dot{\theta}_1) - m_3 s_3 \cos(\theta_1 + \theta_2) (\dot{\theta}_1) & -m_3 s_3 \sin(\theta_1 + \theta_2) & 0 \\ 0 & c_{12} + c_{22} & -2m_3 \sin(\theta_2) (\dot{\theta}_1 + \dot{\theta}_2) & -m_3 s_3 (\dot{\theta}_1) + (1/2)(c_{12} + c_{22}) - m_3 s_2 (\dot{\theta}_1) - m_3 s_3 \cos(\theta_2) (\dot{\theta}_1) & -2m_3 s_3 \cos(\theta_2) (\dot{\theta}_1) - m_3 s_3 \cos(\theta_2) (\dot{\theta}_2) & 0 \\ 0 & 2m_3 \sin(\theta_2) (\dot{\theta}_1) & c_{13} + c_{23} & -m_3 s_3 s_2 \cos(\theta_2) (\dot{\theta}_1) - m_3 s_3 (\dot{\theta}_1) & -2m_3 s_3 (\dot{\theta}_1) - m_3 s_3 (\dot{\theta}_2) & (1/2)(c_{13} + c_{23}) \\ (1/2)(c_{11} + c_{21}) & 2m_3 s_3 \cos(\theta_2) (\dot{\theta}_1) - 2m_3 s_2 (\dot{\theta}_1) + 2m_2 s_2 (\dot{\theta}_1) & 2m_3 s_3 (\dot{\theta}_1 + \dot{\theta}_2) - 2m_3 s_2 \cos(\theta_2) (\dot{\theta}_1 + \dot{\theta}_2) & 2m_3 s_3 s_2 \sin(\theta_2) (\dot{\theta}_2) + (1^2/4)(c_{11} + c_{21}) & m_3 s_3 s_2 \sin(\theta_2) (\dot{\theta}_2) & 0 \\ 0 & (1/2)(c_{12} + c_{22}) + 2m_3 s_3 \cos(\theta_2) (\dot{\theta}_1) & 2m_3 s_3 (\dot{\theta}_1 + \dot{\theta}_2) & m_3 s_3 s_2 \sin(\theta_2) (\dot{\theta}_1) & (1^2/4)(c_{12} + c_{22}) & 0 \\ 0 & 0 & (1/2)(c_{13} - c_{23}) & 0 & 0 & (1^2/4)(c_{13} + c_{23}) \end{bmatrix} \quad (16)$$

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} -m_1 g - m_2 g + k_{11}(s_1 + (1/2)\theta_1 - s_{01}) + k_{21}(s_1 - (1/2)\theta_1 - s_{01}) - m_3 g \\ -m_2 g \cos(\theta_1) + k_{12}(s_2 + (1/2)\theta_2 - s_{02}) + k_{22}(s_2 - (1/2)\theta_2 - s_{02}) - m_3 g \cos(\theta_1) \\ -m_3 g \cos(\theta_1 + \theta_2) + k_{13}(s_3 + (1/2)\theta_3 - s_{03}) + k_{23}(s_3 - (1/2)\theta_3 - s_{03}) \\ m_2 s_2 g \sin(\theta_1) + m_3 s_3 g \sin(\theta_1 + \theta_2) + m_3 s_2 g \sin(\theta_1) + k_{11}(s_1 + (1/2)\theta_1 - s_{01})(1/2) + k_{21}(s_1 - (1/2)\theta_1 - s_{01})(-1/2) \\ m_3 s_3 g \sin(\theta_1 + \theta_2) + k_{12}(s_2 + (1/2)\theta_2 - s_{02})(1/2) + k_{22}(s_2 - (1/2)\theta_2 - s_{02})(-1/2) \\ k_{13}(s_3 + (1/2)\theta_3 - s_{03})(1/2) + k_{23}(s_3 - (1/2)\theta_3 - s_{03})(-1/2) \end{bmatrix} \quad (17)$$

These dynamic formulations are very attractive from a control point of view.

2.2 Linear PD Main Controller

Design of a linear methodology to control of continuum robot manipulator was very straight forward. Since there was an output from the torque model, this means that there would be two inputs into the PD controller. Similarly, the outputs of the controller result from the two control inputs of the torque signal. In a

typical PD method, the controller corrects the error between the desired input value and the measured value. Since the actual position is the measured signal. Figure 1 is shown linear PD methodology, applied to continuum robot manipulator [22-38].

$$e(t) = \theta_a(t) - \theta_d(t) \tag{18}$$

$$U_{PD} = K_{p_a} e + K_{v_a} \dot{e} \tag{19}$$

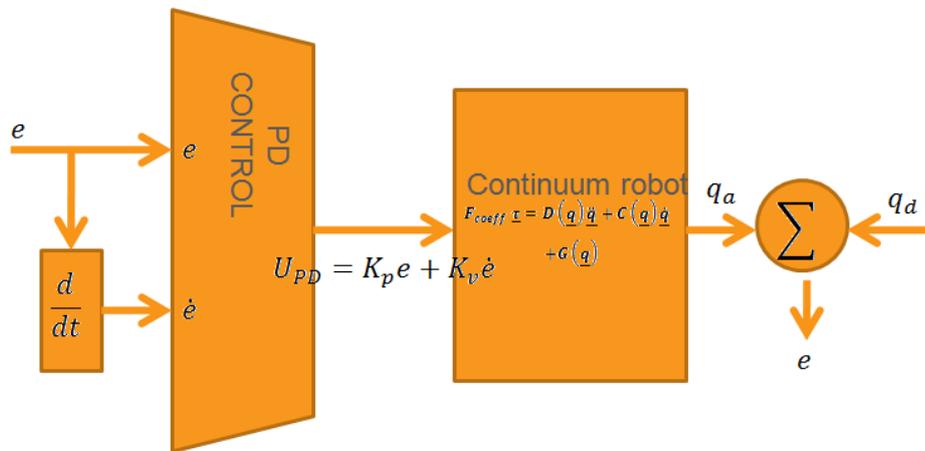


Fig. 1: Block diagram of linear PD method

The model-free control strategy is based on the assumption that the joints of the manipulators are all independent and the system can be decoupled into a group of single-axis control systems [18-23]. Therefore, the kinematic control method always results in a group of individual controllers, each for an active joint of the manipulator. With the independent joint assumption, no a priori knowledge of robot manipulator dynamics is needed in the kinematic controller design, so the complex computation of its dynamics can be avoided and the controller design can be greatly simplified. This is suitable for real-time control applications when powerful processors, which can execute complex algorithms rapidly, are not accessible. However, since joints coupling is neglected, control performance degrades as operating speed increases and a manipulator controlled in this way is only appropriate for relatively slow motion [44, 46]. The fast motion requirement results in even higher dynamic coupling between the various robot joints, which cannot be compensated for by a standard robot controller such as PD [47], and hence model-based control becomes the alternative.

2.3 Variable Structure Controller

Consider a nonlinear single input dynamic system is defined by [23-34]:

$$\mathbf{x}^{(n)} = \mathbf{f}(\bar{\mathbf{x}}) + \mathbf{b}(\bar{\mathbf{x}})\mathbf{u} \tag{20}$$

Where \mathbf{u} is the vector of control input, $\mathbf{x}^{(n)}$ is the n^{th} derivation of \mathbf{x} , $\mathbf{x} = [\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}, \dots, \mathbf{x}^{(n-1)}]^T$ is the state vector, $\mathbf{f}(\mathbf{x})$ is unknown or uncertainty, and $\mathbf{b}(\mathbf{x})$ is of known *sign* function. The main goal to design this controller is train to the desired state; $\mathbf{x}_d = [\mathbf{x}_d, \dot{\mathbf{x}}_d, \ddot{\mathbf{x}}_d, \dots, \mathbf{x}_d^{(n-1)}]^T$, and tracking error vector is defined by [20-34]:

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d = [\tilde{\mathbf{x}}, \dots, \tilde{\mathbf{x}}^{(n-1)}]^T \tag{21}$$

A time-varying sliding surface $\mathbf{s}(\mathbf{x}, \mathbf{t})$ in the state space \mathbf{R}^n is given by [33]:

$$\mathbf{s}(\mathbf{x}, \mathbf{t}) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{\mathbf{x}} = \mathbf{0} \tag{22}$$

where λ is the positive constant. To further penalize tracking error, integral part can be used in sliding surface part as follows [24-35]:

$$\mathbf{s}(\mathbf{x}, \mathbf{t}) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \left(\int_0^t \tilde{\mathbf{x}} dt\right) = \mathbf{0} \tag{23}$$

The main target in this methodology is kept the sliding surface slope $\mathbf{s}(\mathbf{x}, \mathbf{t})$ near to the zero. Therefore,

one of the common strategies is to find input U outside of $s(x, t)$ [24-33].

$$\frac{1}{2} \frac{d}{dt} s^2(x, t) \leq -\zeta |s(x, t)| \quad (24)$$

where ζ is positive constant.

$$\text{If } S(0) > 0 \rightarrow \frac{d}{dt} S(t) \leq -\zeta \quad (25)$$

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

$$\int_{t=0}^{t=t_{reach}} \frac{d}{dt} S(t) \leq - \int_{t=0}^{t=t_{reach}} \eta \rightarrow S(t_{reach}) - S(0) \leq -\zeta(t_{reach} - 0) \quad (26)$$

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

$$0 - S(0) \leq -\eta(t_{reach}) \rightarrow t_{reach} \leq \frac{S(0)}{\zeta} \quad (27)$$

and

$$\text{if } S(0) < 0 \rightarrow 0 - S(0) \leq -\eta(t_{reach}) \rightarrow S(0) \leq \frac{-\zeta(t_{reach})}{|S(0)|} \rightarrow t_{reach} \leq \frac{|S(0)|}{\eta} \quad (28)$$

Equation (28) guarantees time to reach the sliding surface is smaller than $\frac{|S(0)|}{\zeta}$ since the trajectories are outside of $S(t)$.

$$\text{if } S_{t_{reach}} = S(0) \rightarrow \text{error}(x - x_d) = 0 \quad (29)$$

suppose S is defined as

$$s(x, t) = \left(\frac{d}{dt} + \lambda\right) \tilde{x} = (\dot{x} - \dot{x}_d) + \lambda(x - x_d) \quad (30)$$

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

$$\dot{S} = (\ddot{x} - \ddot{x}_d) + \lambda(\dot{x} - \dot{x}_d) \quad (31)$$

suppose the second ordersystemis defined as;

$$\ddot{x} = f + u \rightarrow \dot{S} = f + U - \ddot{x}_d + \lambda(\dot{x} - \dot{x}_d) \quad (32)$$

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

$$\hat{U} = -\hat{f} + \ddot{x}_d - \lambda(\dot{x} - \dot{x}_d) \quad (33)$$

A simple solution to get the sliding condition when the dynamic parameters have uncertainty is the switching control law [52-53]:

$$U_{dis} = \hat{U} - K(\tilde{x}, t) \cdot \text{sgn}(s) \quad (34)$$

where the switching function $\text{sgn}(S)$ is defined as [1, 6]

$$\text{sgn}(s) = \begin{cases} 1 & s > 0 \\ -1 & s < 0 \\ 0 & s = 0 \end{cases} \quad (35)$$

and the $K(\tilde{x}, t)$ is the positive constant. Suppose by (24) the following equation can be written as,

$$\begin{aligned} \frac{1}{2} \frac{d}{dt} s^2(x, t) &= S \cdot \dot{S} \\ &= [f - \hat{f} - K \text{sgn}(s)] \cdot S \\ &= (f - \hat{f}) \cdot S - K|S| \end{aligned} \quad (36)$$

and if the equation (28) instead of (27) the sliding surface can be calculated as

$$\begin{aligned} s(x, t) &= \left(\frac{d}{dt} + \lambda\right)^2 \left(\int_0^t \tilde{x} dt\right) \\ &= (\ddot{x} - \ddot{x}_d) + 2\lambda(\dot{x} - \dot{x}_d) - \lambda^2(x - x_d) \end{aligned} \quad (37)$$

in this method the approximation of U is computed as [6]

$$\hat{U} = -\hat{f} + \ddot{x}_d - 2\lambda(\dot{x} - \dot{x}_d) + \lambda^2(x - x_d) \quad (38)$$

Based on above discussion, the variable structure control law for a multi degrees of freedom robot manipulator is written as [1, 6]:

$$U = U_{eq} + U_{dis} \quad (39)$$

Where, the model-based component U_{eq} is the nominal dynamics of systems calculated as follows [1]:

$$U_{eq} = [D^{-1}(C + G) + \dot{S}]D \quad (40)$$

and U_{dis} is computed as [1];

$$\tau_{dis} = K \cdot \text{sgn}(S) \quad (41)$$

By (41) and (40) the variable structure control of robot manipulator is calculated as;

$$\tau = [D^{-1}(C + G) + \dot{S}]D + K \cdot \text{sgn}(S) \quad (42)$$

where $S = \lambda e + \dot{e}$ in PD-SMC.

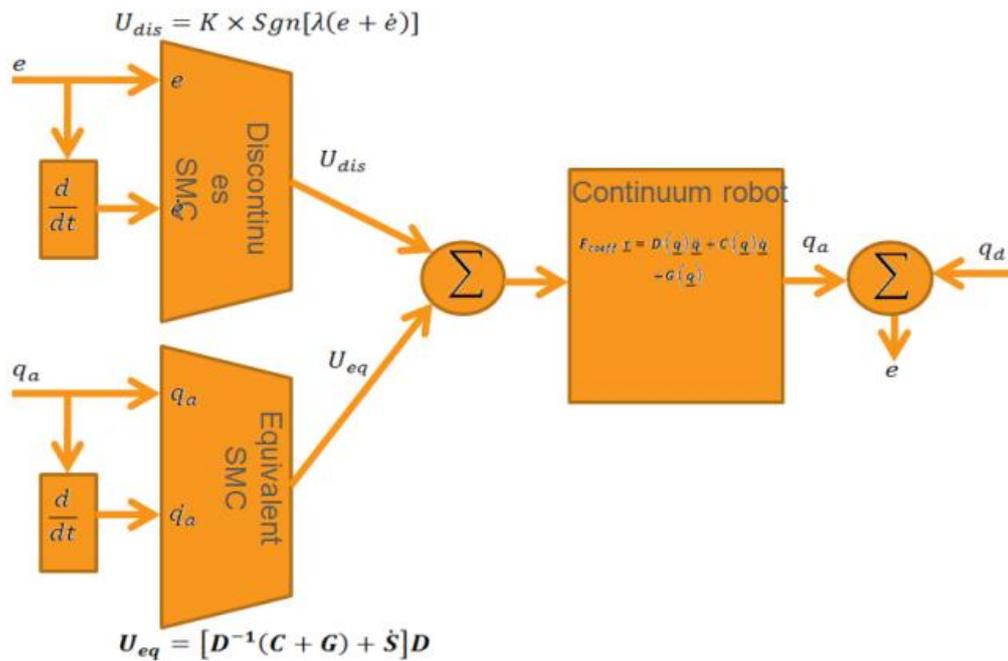


Fig. 2: Block diagram of nonlinear SMC compensator

2.4 Proof of Stability

The lyapunov formulation can be written as follows,

$$V = \frac{1}{2} S^T \cdot D \cdot S \tag{43}$$

The derivation of V can be determined as,

$$\dot{V} = \frac{1}{2} S^T \cdot \dot{D} \cdot S + S^T D \dot{S} \tag{44}$$

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

$$D \dot{S} = -VS + D \dot{S} + C + G \tag{45}$$

It is assumed that

$$S^T (\dot{D} + C + G) S = 0 \tag{46}$$

by substituting (45) in (44)

$$\begin{aligned} \dot{V} &= \frac{1}{2} S^T \dot{D} S + CS + S^T (D \dot{S} + CS + G) \\ &= S^T (D \dot{S} + CS + G) \end{aligned} \tag{47}$$

Suppose the control input is written as follows

$$\begin{aligned} \hat{U} &= U_{\text{Nonlinear}} + \hat{U}_{\text{dis}} \\ &= [\hat{D}^{-1}(C + G) + \hat{S}] \hat{D} \\ &\quad + K \cdot \text{sgn}(S) + CS + G \end{aligned} \tag{48}$$

By replacing the equation (48) in (43)

$$\begin{aligned} \dot{V} &= S^T (D \dot{S} + C + G - \hat{D} \dot{S} - \hat{C} S + G \\ &\quad - K \text{sgn}(S)) \\ &= S^T (\tilde{D} \dot{S} + \tilde{C} S + G \\ &\quad - K \text{sgn}(S)) \end{aligned} \tag{49}$$

and

$$|\tilde{D} \dot{S} + \tilde{C} S + G| \leq |\tilde{D} \dot{S}| + |\tilde{C} S + G| \tag{50}$$

The Lemma equation in CONTINUUM robot arm system can be written as follows

$$\begin{aligned} K_u &= [|\tilde{D} \dot{S}| + |CS + G| + \eta]_i, i \\ &= 1, 2, 3, 4, \dots \end{aligned} \tag{51}$$

and finally;

$$\dot{V} \leq - \sum_{i=1}^n \eta_i |S_i| \tag{52}$$

2.5 Fuzzy Logic Methodology

Based on foundation of fuzzy logic methodology; fuzzy logic controller has played important rule to design nonlinear controller for nonlinear and uncertain systems [47]. However the application area for fuzzy control is really wide, the basic form for all command types of controllers consists of; Input fuzzification

(binary-to-fuzzy [B/F] conversion) Fuzzy rule base (knowledge base), Inference engine and Output defuzzification (fuzzy-to-binary [F/B] conversion). Figure 3 shows the fuzzy controller part.

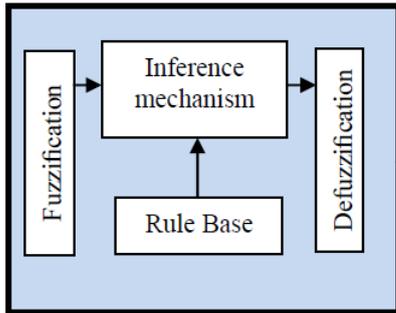


Fig. 3: Fuzzy Controller Part

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

use a singleton as a membership function of the rule consequent part. The following definition shows the Mamdani and Sugeno fuzzy rule base [22-33]

$$\begin{aligned} & \text{if } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z \text{ is } C \text{ 'mamdani'} \\ & \text{if } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z \text{ is } f(x,y) \text{ 'sugeno'} \end{aligned} \quad (53)$$

When x and y have crisp values fuzzification calculates the membership degrees for antecedent part. Rule evaluation focuses on fuzzy operation (*AND/OR*) in the antecedent of the fuzzy rules. The aggregation is used to calculate the output fuzzy set and several methodologies can be used in fuzzy logic controller aggregation, namely, Max-Min aggregation, Sum-Min aggregation, Max-bounded product, Max-drastic product, Max-bounded sum, Max-algebraic sum and Min-max. Defuzzification is the last step in the fuzzy inference system which it is used to transform fuzzy set to crisp set. Consequently defuzzification's input is the aggregate output and the defuzzification's output is a crisp number. Centre of gravity method (*COG*) and Centre of area method (*COA*) are two most common defuzzification methods. Figure 4 shows fuzzy sliding mode compensator.

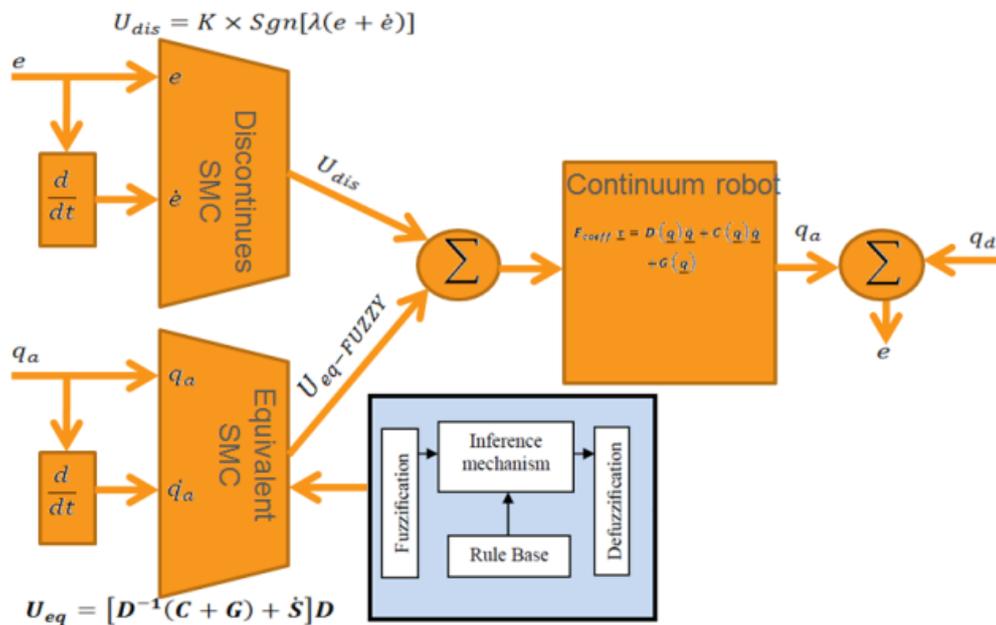


Fig. 4: Fuzzy logic estimator SMC

III. Methodology

Based on the dynamic formulation of continuum robot manipulator, (3), and the industrial PD law (5) in this paper we discuss about regulation problem, the desired position is constant, i.e., $\dot{q}_d = 0$. In most robot

manipulator control, desired joint positions are generated by the trajectory planning. The objective of robot control is to design the input torque in (1) such that the tracking error

$$e = q_d - q_a \quad (54)$$

When the dynamic parameters of robot formulation known, the PD control formulation (25) should include a compensator as

$$\tau = -k_p e - k_d \dot{e} + (G + F) \tag{55}$$

Where G is gravity and F is apposite definite diagonal matrix friction term (coulomb friction). If we use a Lyapunov function candidate as

$$V_{pd} = \frac{1}{2} \dot{q}^T M \dot{q} + \frac{1}{2} e^T k_p e \tag{56}$$

$$\dot{V}_{pd} = -\dot{q}^T k_d \dot{q} \leq 0 \tag{57}$$

Based on above discussion $\dot{q} = 0$ and $e = 0$ are only initial conditions in $\Omega = \{[\dot{q}, e] : \dot{V} = 0\}$, for which $[\dot{q}, e] \in \Omega$ for all $t \leq 0$. By the LaSalle's invariance principle, $e \rightarrow 0$ and $\dot{e} \rightarrow 0$. When G and F in (25) are unknown, a fuzzy logic can be used to approximate them as

$$f(x) = \sum_{l=1}^M \theta^l \varepsilon^l(x) = \theta^T \varepsilon(x) \tag{58}$$

Where $\theta = (\theta^1, \dots, \theta^M)^T$, $\varepsilon(x) = (\varepsilon^1(x), \dots, \varepsilon^M(x))^T$, and $\varepsilon^l(x) = \frac{\mu_{A_l^i}(x_i)}{\sum_{i=1}^n \mu_{A_l^i}(x_i)}$. $\theta^1, \dots, \theta^M$ are adjustable parameters in (58). $\mu_{A_1^1}(x_1), \dots, \mu_{A_n^m}(x_n)$ are given membership functions whose parameters will not change over time.

The second type of fuzzy systems is given by

$$f(x) = \frac{\sum_{l=1}^M \theta^l \left[\prod_{i=1}^n \exp\left(-\left(\frac{x_i - \alpha_i^l}{\delta_i^l}\right)^2\right) \right]}{\sum_{l=1}^M \left[\prod_{i=1}^n \exp\left(-\left(\frac{x_i - \alpha_i^l}{\delta_i^l}\right)^2\right) \right]} \tag{59}$$

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

$$f^\wedge(x|\theta) = \theta^T \varepsilon(x) \tag{60}$$

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

$$\theta^* = \arg \min_{\theta \in \Omega} \left[\sup_{x \in U} |f^\wedge(x|\theta) - g(x)| \right] \tag{61}$$

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 (58) to estimate the nonlinear system (26) the fuzzy formulation can be written:

$$f(x|\theta) = \frac{\theta^T \varepsilon(x)}{\sum_{l=1}^n \theta^l [\mu_{A_l^i}(x)]} = \frac{\sum_{l=1}^n \theta^l [\mu_{A_l^i}(x)]}{\sum_{l=1}^n [\mu_{A_l^i}(x)]} \tag{62}$$

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

$$f(x) = \Theta^T \varepsilon(x) \tag{63}$$

Where

$$\Theta^T = (\theta_1, \dots, \theta_m)^T = \begin{bmatrix} \theta_1^1, \theta_1^2, \dots, \theta_1^M \\ \theta_2^1, \theta_2^2, \dots, \theta_2^M \\ \vdots \\ \theta_m^1, \theta_m^2, \dots, \theta_m^M \end{bmatrix} \tag{64}$$

$\varepsilon(x) = (\varepsilon^1(x), \dots, \varepsilon^M(x))^T$, $\varepsilon^1(x) = \prod_{i=1}^n \mu_{A_i^1}(x_i) / \sum_{i=1}^M (\prod_{i=1}^n \mu_{A_i^i}(x_i))$, and $\mu_{A_i^i}(x_i)$ is defined in (62). To reduce the number of fuzzy rules, we divide the fuzzy system in to three parts:

$$F^1(q, \dot{q}) = \Theta^1{}^T \varepsilon(q, \dot{q}) = \left[\theta_1^1{}^T \varepsilon(q, \dot{q}), \dots, \theta_m^1{}^T \varepsilon(q, \dot{q}) \right]^T \tag{65}$$

$$F^2(q, \ddot{q}_r) = \Theta^2{}^T \varepsilon(q, \ddot{q}_r) = \left[\theta_1^2{}^T \varepsilon(q, \ddot{q}_r), \dots, \theta_m^2{}^T \varepsilon(q, \ddot{q}_r) \right]^T \tag{66}$$

$$F^3(q, \ddot{q}) = \Theta^3{}^T \varepsilon(q, \ddot{q}) = \left[\theta_1^3{}^T \varepsilon(q, \ddot{q}), \dots, \theta_m^3{}^T \varepsilon(q, \ddot{q}) \right]^T \tag{67}$$

The control security input is given by

$$\tau = D \ddot{q}_r + C(q) \dot{q}^2 + g(q) + F^1(q, \dot{q}) + F^2(q, \ddot{q}_r) + F^3(q, \ddot{q}) - K_p e - K_v \dot{e} \tag{68}$$

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

Based on sliding mode formulation (42) and PD linear methodology (19);

$$S_{New} = (\dot{e} + \lambda e) \tag{69}$$

And U_{switch} is obtained by

$$U_{switch} = K(\vec{x}, t) \cdot \text{sgn}(S_{New}) = K(\vec{x}, t) \cdot \text{sgn}(K(\dot{e} + \lambda e)) \tag{70}$$

The Lyapunov function in this design is defined as

$$V = \frac{1}{2} S^T M S + \frac{1}{2} \sum_{j=1}^M \frac{1}{\gamma_{sj}} \phi^T \cdot \phi_j \quad (71)$$

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

$$S^T \dot{D} S + \frac{1}{2} S^T \dot{D} S = S^T (D \dot{S} + V S) \quad (72)$$

If the dynamic formulation of robot manipulator defined by

$$\tau = D(q) \ddot{q} + V(q, \dot{q}) \dot{q} + G(q) \quad (73)$$

The controller formulation is defined by

$$\tau = \hat{D} \ddot{q}_r + \hat{V} \dot{q}_r + \hat{G} - \lambda S - K \quad (74)$$

According to (72) and (73)

$$D(q) \ddot{q} + V(q, \dot{q}) \dot{q} + G(q) = \hat{D} \ddot{q}_r + \hat{V} \dot{q}_r + \hat{G} - \lambda S - K \quad (75)$$

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

$$D \dot{S} + (V + \lambda) S = \Delta f - K \quad (76)$$

$$D \dot{S} = \Delta f - K - V S - \lambda S$$

The derivation of V is defined

$$\dot{V} = S^T D \dot{S} + \frac{1}{2} S^T \dot{D} S + \sum_{j=1}^M \frac{1}{\gamma_{sj}} \phi^T \cdot \dot{\phi}_j \quad (77)$$

$$\dot{V} = S^T (D \dot{S} + V S) + \sum_{j=1}^M \frac{1}{\gamma_{sj}} \phi^T \cdot \dot{\phi}_j$$

Based on (75) and (76)

$$\dot{V} = S^T (\Delta f - K - V S - \lambda S + V S) + \sum_{j=1}^M \frac{1}{\gamma_{sj}} \phi^T \cdot \dot{\phi}_j \quad (78)$$

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

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

Suppose K_j is defined as follows

$$K_j = \frac{\sum_{l=1}^M \theta_j^l [\mu_{A_j}(S_j)]}{\sum_{l=1}^M \mu_{A_j}(S_j)} = \theta_j^T \zeta_j(S_j) \quad (79)$$

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

$$\zeta_j^1(S_j) = \frac{\mu_{(A_j)^1}(S_j)}{\sum_i \mu_{(A_j)^i}(S_j)} \quad (80)$$

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

The fuzzy system is defined as

$$f(x) = \tau_{fuzzy} = \sum_{l=1}^M \theta^T \zeta(x) = \psi(e, \dot{e}) \quad (81)$$

where $\theta = (\theta^1, \theta^2, \theta^3, \dots, \theta^M)$ is adjustable parameter in (79)

According to (76), (77) and (79);

$$\dot{V} = \sum_{j=1}^M [S_j (\Delta f_j - \theta^T \zeta_j(S_j))] - S^T \lambda S + \sum_{j=1}^M \frac{1}{\gamma_{sj}} \phi^T \cdot \dot{\phi}_j \quad (82)$$

Based on $\phi = \theta^* - \theta \rightarrow \theta = \theta^* - \phi$

$$\dot{V} = \sum_{j=1}^M [S_j (\Delta f_j - \theta^{*T} \zeta_j(S_j) + \phi^T \zeta_j(S_j))] - S^T \lambda S + \sum_{j=1}^M \frac{1}{\gamma_{sj}} \phi^T \cdot \dot{\phi}_j \quad (83)$$

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

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

$$\dot{V} = \sum_{j=1}^m [S_j \Delta f_j - ((\theta_j^*)^T \zeta_j(S_j))] - S^T \lambda S \quad (84)$$

The minimum error is defined by

$$e_{mj} = \Delta f_j - ((\theta_j^*)^T \zeta_j(S_j)) \quad (85)$$

Therefore \dot{V} is computed as

$$\begin{aligned} \dot{V} &= \sum_{j=1}^m [S_j e_{mj}] - S^T \lambda S \\ &\leq \sum_{j=1}^m |S_j| |e_{mj}| - S^T \lambda S \\ &= \sum_{j=1}^m |S_j| |e_{mj}| - \lambda_j S_j^2 \end{aligned} \quad (86)$$

$$= \sum_{j=1}^m |S_j| (|e_{mj}| - \lambda_j S_j) \quad (87)$$

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

$$\sup_{x \in U} |f(x) - g(x)| < \varepsilon \quad (88)$$

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

$$\text{if } \lambda_j = \alpha \text{ that } \alpha |S_j| > e_{mj} \text{ (} S_j \neq 0 \text{)} \text{ then } \dot{V} < 0 \text{ for (} S_j \neq 0 \text{)} \quad (89)$$

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

$$U = (U_{fuzzy} + U_{sliding}) \times U_{PD} \quad (90)$$

$$U = (U_{fuzzy} + U_{switch}) \times U_{PD} = [D^{-1}(B + C + G) + \dot{S}]D + K \cdot \text{sgn}(S) + \quad (91)$$

$$\frac{\sum_{l=1}^M \theta^l \left[\prod_{i=1}^n \exp\left(-\left(\frac{x_i - \alpha_i}{\delta_i^l}\right)^2\right) \right]}{\sum_{l=1}^M \left[\prod_{i=1}^n \exp\left(-\left(\frac{x_i - \alpha_i}{\delta_i^l}\right)^2\right) \right]} \times (K_p e_1 + K_v \dot{e}_1)$$

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

$$G + F + f = \frac{\sum_{l=1}^M \theta^l \left[\prod_{i=1}^n \exp\left(-\left(\frac{x_i - \alpha_i}{\delta_i^l}\right)^2\right) \right]}{\sum_{l=1}^M \left[\prod_{i=1}^n \exp\left(-\left(\frac{x_i - \alpha_i}{\delta_i^l}\right)^2\right) \right]} \quad (92)$$

It is usually is smaller than $G + F + f$; and the upper bound of it is easy to be estimated. Figure 5 shows the PD controller with serial PD fuzzy sliding mode controller.

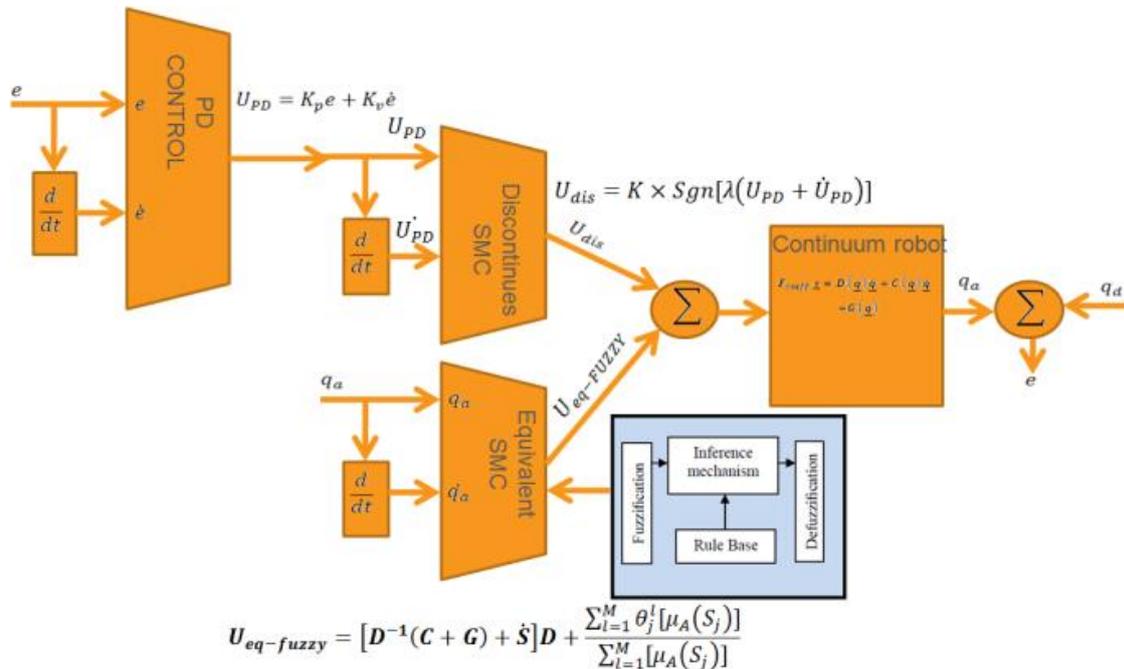


Fig. 5: PD with fuzzy SMC Compensator

IV. Results and Discussion

Linear PD controller (PD) and proposed PD with SISO fuzzy sliding mode estimator were tested to sinus response trajectory. The simulation was implemented in MATLAB/SIMULINK environment. Links trajectory and disturbance rejection 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.

Trajectory: Figure 6 shows the links trajectory in PD controller and proposed PD with SISO fuzzy variable structure estimator without disturbance for sinus trajectory in general and zoom scaling because all 3 joints have the same response so, all links are shown in a graph.

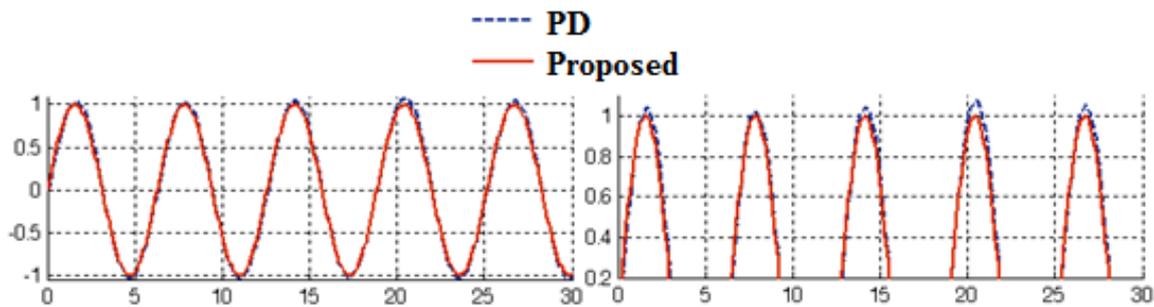


Fig. 6: PD controller Vs. Proposed method

By comparing sinus response, Figure 6, in PD controller and PD with FSMC estimator, we can see the the proposed controller's overshoot (**0%**) is lower than PD's (**7.8%**).

Disturbance rejection: Figure 7 is indicated the power disturbance removal in PD controller and PD with SISO

FSMC estimation. Besides a band limited white noise with predefined of 40% the power of input signal is applied to the sinus PD controller and proposed method; it found slight oscillations in PD trajectory responses. All 3 joints are shown in one graph.

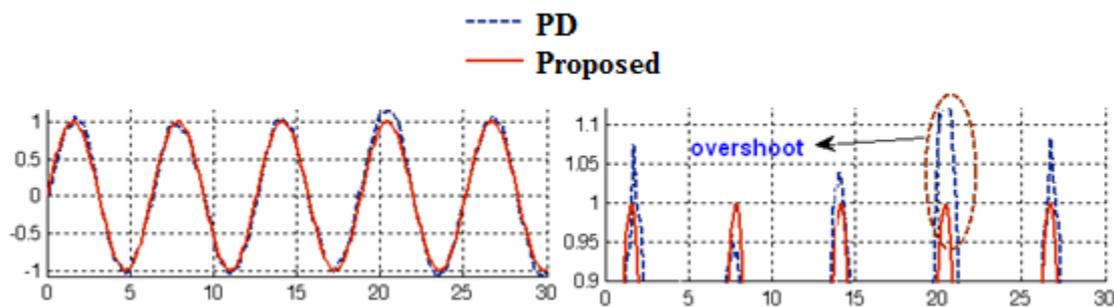


Fig. 7: PD Vs. Proposed method: Robot arm with external disturbance

Among above graph, relating to sinus trajectory following with external disturbance, PD controller has slightly fluctuations. By comparing overshoot; proposed controller's overshoot (**1.3%**) is lower than PD's (**22%**).

improved by using the advantages of sliding mode methodology and artificial intelligence compensate while the disadvantages removed by added each method to previous method.

V. Conclusion

Refer to the research, a Lyapunov based SISO fuzzy sliding mode estimator PD controller design and application to continuum robot arm has proposed in order to design high performance nonlinear estimator in the presence of highly uncertainties and external disturbances. Regarding to the positive points in linear PD controller, sliding mode controller and fuzzy inference system it is found that the fuzzy logic laws derived in the Lyapunov sense. 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 based on artificial intelligence. The second target in this work is compensating the model uncertainty by SISO fuzzy inference system, in the case of continuum robot arm. In finally part fuzzy sliding mode methodology with minimum rule base is used to compensate and adjusted the PD controller. In this case the performance is

References:

- [1] T. R. Kurfess, *Robotics and automation handbook*: CRC, 2005.
- [2] J. J. E. Slotine and W. Li, *Applied nonlinear control* vol. 461: Prentice hall Englewood Cliffs, NJ, 1991.
- [3] K. Ogata, *Modern control engineering*: Prentice Hall, 2009.
- [4] J. J. D'Azzo, C. H. Houpis and S. N. Sheldon, *Linear control system analysis and design with MATLAB*: CRC, 2003.
- [5] B. Siciliano and O. Khatib, *Springer handbook of robotics*: Springer-Verlag New York Inc, 2008.
- [6] F. T. Cheng, T. L. Hour, Y. Y. Sun and T. H. Chen, "Study and resolution of singularities for a 6-DOF PUMA manipulator," *Systems, Man, and*

- Cybernetics, Part B: Cybernetics, IEEE Transactions on, No. 2, vol. 27, pp. 332-343, 2002.*
- [7] M. W. Spong and M. Vidyasagar, *Robot dynamics and control: Wiley-India, 2009.*
- [8] Farzin Piltan, S. Emamzadeh, Z. Hivand, F. Shahriyari & Mina Mirazaei . " PUMA-560 Robot Manipulator Position Sliding Mode Control Methods Using MATLAB/SIMULINK and Their Integration into Graduate/Undergraduate Nonlinear Control, Robotics and MATLAB Courses" *International Journal of Robotics and Automation, 3(3): 2012.*
- [9] Farzin Piltan, J. Meigolinedjad, S. Mehrara, S. Rahmdel. " Evaluation Performance of 2nd Order Nonlinear System: Baseline Control Tunable Gain Sliding Mode Methodology" *International Journal of Robotics and Automation, 3(3): 2012.*
- [10] Farzin Piltan , N. Sulaiman, Zahra Tajpaykar, Payman Ferdosali, Mehdi Rashidi, "Design Artificial Nonlinear Robust Controller Based on CTLC and FSMC with Tunable Gain," *International Journal of Robotic and Automation, 2 (3): 205-220, 2011.*
- [11] Farzin Piltan, A. R. Salehi and Nasri B Sulaiman., " Design artificial robust control of second order system based on adaptive fuzzy gain scheduling," *world applied science journal (WASJ), 13 (5): 1085-1092, 2011.*
- [12] Farzin Piltan, N. Sulaiman, Atefeh Gavahian, Samira Soltani, Samaneh Roosta, "Design Mathematical Tunable Gain PID-Like Sliding Mode Fuzzy Controller with Minimum Rule Base," *International Journal of Robotic and Automation, 2 (3): 146-156, 2011.*
- [13] Farzin Piltan , A. Zare, Nasri B. Sulaiman, M. H. Marhaban and R. Ramli , "A Model Free Robust Sliding Surface Slope Adjustment in Sliding Mode Control for Robot Manipulator," *World Applied Science Journal, 12 (12): 2330-2336, 2011.*
- [14] Farzin Piltan , A. H. Aryanfar, Nasri B. Sulaiman, M. H. Marhaban and R. Ramli "Design Adaptive Fuzzy Robust Controllers for Robot Manipulator," *World Applied Science Journal, 12 (12): 2317-2329, 2011.*
- [15] Farzin Piltan, N. Sulaiman , Arash Zargari, Mohammad Keshavarz, Ali Badri , "Design PID-Like Fuzzy Controller With Minimum Rule Base and Mathematical Proposed On-line Tunable Gain: Applied to Robot Manipulator," *International Journal of Artificial intelligence and expert system, 2 (4):184-195, 2011.*
- [16] Farzin Piltan, Nasri Sulaiman, M. H. Marhaban and R. Ramli, "Design On-Line Tunable Gain Artificial Nonlinear Controller," *Journal of Advances In Computer Research, 2 (4): 75-83, 2011.*
- [17] Farzin Piltan, N. Sulaiman, Payman Ferdosali, Iraj Assadi Talooki, "Design Model Free Fuzzy Sliding Mode Control: Applied to Internal Combustion Engine," *International Journal of Engineering, 5 (4):302-312, 2011.*
- [18] Farzin Piltan, N. Sulaiman, Samaneh Roosta, M.H. Marhaban, R. Ramli, "Design a New Sliding Mode Adaptive Hybrid Fuzzy Controller," *Journal of Advanced Science & Engineering Research , 1 (1): 115-123, 2011.*
- [19] Farzin Piltan, Atefeh Gavahian, N. Sulaiman, M.H. Marhaban, R. Ramli, "Novel Sliding Mode Controller for robot manipulator using FPGA," *Journal of Advanced Science & Engineering Research, 1 (1): 1-22, 2011.*
- [21] Farzin Piltan, N. Sulaiman, Iraj Asadi Talooki, Payman Ferdosali, "Control of IC Engine: Design a Novel MIMO Fuzzy Backstepping Adaptive Based Fuzzy Estimator Variable Structure Control," *International Journal of Robotics and Automation, 2 (5):360-380, 2011.*
- [22] Farzin Piltan, N. Sulaiman, Payman Ferdosali, Mehdi Rashidi, Zahra Tajpeikar, "Adaptive MIMO Fuzzy Compensate Fuzzy Sliding Mode Algorithm: Applied to Second Order Nonlinear System," *International Journal of Engineering, 5 (5): 380-398, 2011.*
- [23] Farzin Piltan, N. Sulaiman, Hajar Nasiri, Sadeq Allahdadi, Mohammad A. Bairami, "Novel Robot Manipulator Adaptive Artificial Control: Design a Novel SISO Adaptive Fuzzy Sliding Algorithm Inverse Dynamic Like Method," *International Journal of Engineering, 5 (5): 399-418, 2011.*
- [24] Piltan, F., et al. "Design sliding mode controller for robot manipulator with artificial tunable gain". *Canadian Journal of pure and applied science, 5 (2), 1573-1579, 2011.*
- [25] Farzin Piltan, N. Sulaiman, Sadeq Allahdadi, Mohammadali Dialame, Abbas Zare, "Position Control of Robot Manipulator: Design a Novel SISO Adaptive Sliding Mode Fuzzy PD Fuzzy Sliding Mode Control," *International Journal of Artificial intelligence and Expert System, 2 (5):208-228, 2011.*
- [26] Farzin Piltan, SH. Tayebi HAGHIGHI, N. Sulaiman, Iman Nazari, Sobhan Siamak, "Artificial Control of PUMA Robot Manipulator: A-Review of Fuzzy Inference Engine And Application to Classical Controller ," *International Journal of Robotics and Automation, 2 (5):401-425, 2011.*
- [27] Farzin Piltan, N. Sulaiman, Abbas Zare, Sadeq Allahdadi, Mohammadali Dialame, "Design

- Adaptive Fuzzy Inference Sliding Mode Algorithm: Applied to Robot Arm," *International Journal of Robotics and Automation*, 2 (5): 283-297, 2011.
- [28] Farzin Piltan, Amin Jalali, N. Sulaiman, Atefeh Gavahian, Sobhan Siamak, "Novel Artificial Control of Nonlinear Uncertain System: Design a Novel Modified PSO SISO Lyapunov Based Fuzzy Sliding Mode Algorithm," *International Journal of Robotics and Automation*, 2 (5): 298-316, 2011.
- [29] Farzin Piltan, N. Sulaiman, Amin Jalali, Koorosh Aslansefat, "Evolutionary Design of Mathematical tunable FPGA Based MIMO Fuzzy Estimator Sliding Mode Based Lyapunov Algorithm: Applied to Robot Manipulator," *International Journal of Robotics and Automation*, 2 (5):317-343, 2011.
- [30] Farzin Piltan, N. Sulaiman, Samaneh Roosta, Atefeh Gavahian, Samira Soltani, "Evolutionary Design of Backstepping Artificial Sliding Mode Based Position Algorithm: Applied to Robot Manipulator," *International Journal of Engineering*, 5 (5):419-434, 2011.
- [31] Farzin Piltan, N. Sulaiman, S.Soltani, M. H. Marhaban & R. Ramli, "An Adaptive sliding surface slope adjustment in PD Sliding Mode Fuzzy Control for Robot Manipulator," *International Journal of Control and Automation*, 4 (3): 65-76, 2011.
- [32] Farzin Piltan, N. Sulaiman, Mehdi Rashidi, Zahra Tajpaikar, Payman Ferdosali, "Design and Implementation of Sliding Mode Algorithm: Applied to Robot Manipulator-A Review," *International Journal of Robotics and Automation*, 2 (5):265-282, 2011.
- [33] Farzin Piltan, N. Sulaiman, Amin Jalali, Sobhan Siamak, and Iman Nazari, "Control of Robot Manipulator: Design a Novel Tuning MIMO Fuzzy Backstepping Adaptive Based Fuzzy Estimator Variable Structure Control," *International Journal of Control and Automation*, 4 (4):91-110, 2011.
- [34] Farzin Piltan, N. Sulaiman, Atefeh Gavahian, Samaneh Roosta, Samira Soltani, "On line Tuning Premise and Consequence FIS: Design Fuzzy Adaptive Fuzzy Sliding Mode Controller Based on Lyapunov Theory," *International Journal of Robotics and Automation*, 2 (5):381-400, 2011.
- [35] Farzin Piltan, N. Sulaiman, Samaneh Roosta, Atefeh Gavahian, Samira Soltani, "Artificial Chattering Free on-line Fuzzy Sliding Mode Algorithm for Uncertain System: Applied in Robot Manipulator," *International Journal of Engineering*, 5 (5):360-379, 2011.
- [36] Farzin Piltan, N. Sulaiman and I.AsadiTalooki, "Evolutionary Design on-line Sliding Fuzzy Gain Scheduling Sliding Mode Algorithm: Applied to Internal Combustion Engine," *International Journal of Engineering Science and Technology*, 3 (10):7301-7308, 2011.
- [37] Farzin Piltan, Nasri B Sulaiman, Iraj Asadi Talooki and Payman Ferdosali, "Designing On-Line Tunable Gain Fuzzy Sliding Mode Controller Using Sliding Mode Fuzzy Algorithm: Applied to Internal Combustion Engine," *world applied science journal (WASJ)*, 15 (3): 422-428, 2011.
- [38] B. K. Yoo and W. C. Ham, "Adaptive control of robot manipulator using fuzzy compensator," *Fuzzy Systems, IEEE Transactions on*, No. 2, vol. 8, pp. 186-199, 2002.
- [39] Y. S. Kung, C. S. Chen and G. S. Shu, "Design and Implementation of a Servo System for Robotic Manipulator," *CACS*, 2005.
- [40] Farzin Piltan, N. Sulaiman, M. H. Marhaban, Adel Nowzary, Mostafa Tohidian, "Design of FPGA based sliding mode controller for robot manipulator," *International Journal of Robotic and Automation*, 2 (3): 183-204, 2011.
- [41] Farzin Piltan, M. Mirzaie, F. Shahriyari, Iman Nazari & S. Emamzadeh, "Design Baseline Computed Torque Controller" *International Journal of Engineering*, 3(3): 2012.
- [42] Farzin Piltan, H. Rezaie, B. Boroomand, Arman Jahed, "Design robust back stepping online tuning feedback linearization control applied to IC engine," *International Journal of Advance Science and Technology*, 42: 183-204, 2012.
- [43] Farzin Piltan, I. Nazari, S. Siamak, P. Ferdosali, "Methodology of FPGA-based mathematical error-based tuning sliding mode controller" *International Journal of Control and Automation*, 5(1): 89-110, 2012.
- [44] Farzin Piltan, M. A. Dialame, A. Zare, A. Badri, "Design Novel Lookup table changed Auto Tuning FSMC: Applied to Robot Manipulator" *International Journal of Engineering*, 6(1): 25-40, 2012.
- [45] Farzin Piltan, B. Boroomand, A. Jahed, H. Rezaie, "Methodology of Mathematical Error-Based Tuning Sliding Mode Controller" *International Journal of Engineering*, 6(2): 96-112, 2012.
- [46] Farzin Piltan, F. Aghayari, M. R. Rashidian, M. Shamsodini, "A New Estimate Sliding Mode Fuzzy Controller for Robotic Manipulator" *International Journal of Robotics and Automation*, 3(1): 45-58, 2012.
- [47] Farzin Piltan, M. Keshavarz, A. Badri, A. Zargari, "Design novel nonlinear controller applied to robot manipulator: design new feedback linearization fuzzy controller with minimum rule

base tuning method" International Journal of Robotics and Automation, 3(1): 1-18, 2012.

How to cite this paper: Amin Jalali, Farzin Piltan, Mohammadreza Hashemzadeh, Fatemeh BibakVaravi, Hossein Hashemzadeh, "Design Parallel Linear PD Compensation by Fuzzy Sliding Compensator for Continuum Robot", International Journal of Information Technology and Computer Science(IJITCS), vol.5, no.12, pp.97-112, 2013. DOI: 10.5815/ijitcs.2013.12.12

Authors' Profiles



Amin Jalali is an Electrical/Electronic researcher of research and development company SSP. Co. In 2010 he is jointed the research and development company, SSP Co, Shiraz, Iran. He is the main author of more than 6 scientific papers in refereed journals. His main areas of research interests are Nonlinear Control and Automation, Artificial Intelligence Control, System Identification, Optimization, Fuzzy Theory, Neural Network, and Energy Systems.



Farzin Piltan was born on 1975, Shiraz, Iran. In 2004 he is jointed the research and development company, SSP Co, Shiraz, Iran. In addition to 7 textbooks, Farzin Piltan is the main author of more than 70 scientific papers in refereed journals and also is an editorial board of International Journal of Control and Automation (IJCA) and also is one of the reviewers of International Journal of Robotics and Automation (IJRA). His main areas of research interests are nonlinear control, artificial control system and applied to FPGA, robotics and artificial nonlinear control and IC engine modeling and control.



Mohammadreza Hashemzadeh is an electronic engineer researcher. His research activities deal with the Electronics, Information technology and Artificial Control System.



Fatemeh Bibak Varavi is a researcher in her universities. Her research activities deal with the biomedical engineering, artificial intelligence, robotics and computer Vision.



Hossein Hashemzadeh is a computer engineer researcher. His research activities deal with the Information Technology, Nano technology, and artificial Control System.