

Design Novel Soft Computing Backstepping Controller with Application to Nonlinear Dynamic Uncertain System

Amin Jalali

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

E-mail: Max.Jalali@gmail.com

Farzin Piltan

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

E-mail: Piltan_f@iranssp.com

Hossein Hashemzadeh

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

E-mail: hossein.hshm@gmail.com

Alireza Hasiri

Department of Control and Instrument Engineering, College of Electrical Engineering, Sarvestan Branch, Islamic Azad University, Fars, Iran

E-mail: alireza_hasiri2010@yahoo.com

Mohammadreza Hashemzadeh

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

E-mail: h.mohammadreza33@yahoo.com

Abstract— The increasing demand for multi-degree-of-freedom (DOF) continuum robot in presence of highly nonlinear dynamic parameters in a number of industries has motivated a flurry of research in the development of soft computing nonlinear methodology. This robot is capable of providing smooth and isotropic three-dimensional motion in each joint. Compared to conventional robotic manipulators that offer the same motion capabilities, the innovative spherical motor possesses several advantages. Not only can the spherical motor combine 3-DOF motion in a single joint, it has a large range of motion with no singularities in its workspace. This research contributes to the on-going research effort by exploring alternate methods for controlling the continuum robot manipulator. This research addresses two basic issues related to the control of a continuum robots; (1) a more accurate representation of the dynamic model of an existing prototype, and (2) the design of a robust feedback controller. The robust backstepping controller proposed in this research is used to further demonstrate the appealing features exhibited by the continuum robot. Robust feedback controller is used to position control of continuum robot in presence of uncertainties. Using

Lyapunov type stability arguments, a robust backstepping controller is designed to achieve this objective. The controller developed in this research is designed in two steps. Firstly, a robust stabilizing torque is designed for the nominal continuum robot dynamics derived using the constrained Lagrangian formulation. Next, the fuzzy logic methodology applied to it to solution uncertainty problem. The fuzzy model free problem is formulated to minimize the nonlinear formulation of continuum robot. The eventual stability of the controller depends on the torque generating capabilities of the continuum robots.

Index Terms— Backstepping Methodology, Fuzzy Logic Algorithm, Continuum Robot Manipulator, Uncertainty

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 underactuated (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 [6], [3]-[5] 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. physicsbased 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.

The designed controller not only demonstrates the appealing features exhibited by the spherical motor, but also demonstrates some of the nice features of backstepping-type controllers as well. The work done in this research contributes to bringing the continuum robot a step closer to practical industrial application. Controller is a device which can sense information from linear or nonlinear system (e.g., continuum robot) to improve the systems performance [7-9]. The main targets in designing control systems are stability, good disturbance rejection, and small tracking error[7-12]. Several continuum robot are controlled by linear methodologies (e.g., Proportional-Derivative (PD) controller, Proportional- Integral (PI) controller or Proportional- Integral-Derivative (PID) controller), but when robot works with various payloads and have uncertainty in dynamic models this technique has limitations. In some applications continuum robot are used in an unknown and unstructured environment, 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) [8-10].

Advanced control techniques such as integrator backstepping, feedback linearization, adaptive and robust have been applied to the control of numerous single-axis machines and robotic manipulators. Since available control techniques for continuum robot and robotic manipulators are so broad, the review in this research is restricted to some of the nonlinear control techniques for continuum robot. Most of the authors referenced on the nonlinear control techniques for

continuum robot also do a significant amount of work on the control of robotic manipulators. Kokotovic [13] published one of the pioneering works on the backstepping control technique and Qu et al. [14] extended this technique and developed a robust backstepping-type controller for a one-link robot with the motor dynamics taken into consideration. Carroll et al. [15] also extended the work of Kokotovic [16] to design an embedded computed torque and output feedback controller for permanent magnet brush de (BDC) motors. Hemati et al. [16] developed a robust feedback linearizing controller for a single-link robot actuated by a brushless de motor (BLOC). In [17], Carroll et al. also developed a robust tracking controller for a BLOC, which achieved globally bounded results for rotor position tracking error despite parametric uncertainties and additive bounded disturbances. In addition to DC machines, SR and PM stepper motors are also candidates for advanced nonlinear controllers. In [18], Die'-Spong et al. introduced a detailed nonlinear model and an electronic commutation strategy for the SR motor and applied a state feedback control algorithm which compensated for all the nonlinearities of the system. The work in [18] was then generalized to a direct-drive manipulator with SR actuation by Taylor et al. [19]. Carroll et al. [20] also used a backstepping technique to develop an adaptive tracking controller for the SR motor. Bodson [21] developed a model-based control law for the PM stepper motor using an exact linearization methodology while considering practical issues such as voltage saturation. Even though some of the above control techniques are not of the backstepping type controller, the backstepping-type controller developed in this thesis was somewhat inspired by them.

This paper is organized as follows; section 2, is served as an introduction to the feedback backstepping controller formulation algorithm and its application to control of continuum robot and dynamic of continuum robot. Part 3, introduces and describes the methodology. Section 4 presents the simulation results and discussion of this algorithm applied to a continuum robot and the final section is describing the conclusion.

II. Theory

2.1 Dynamic Formulation of Continuum Robot

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);

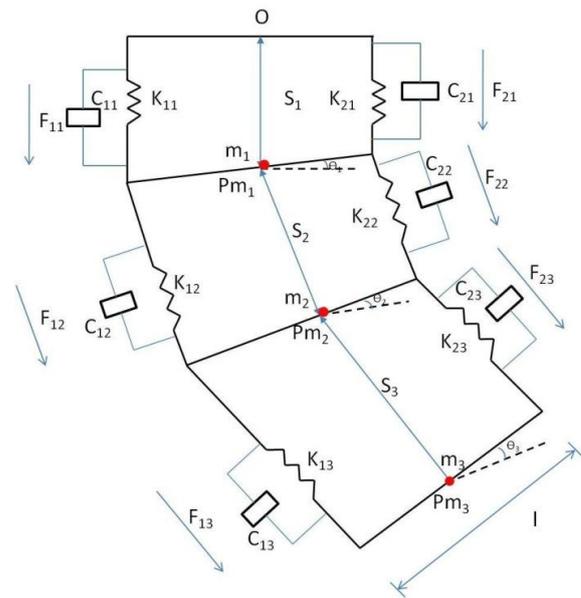


Fig. 1: Assumed structure for analytical model of a section of a continuum arm

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

$${}^N_1\mathbf{P} = S_1 \cdot \widehat{\mathbf{n}}_3 \quad (1)$$

$${}^N_2\mathbf{P} = S_2 \cdot \sin\theta_1 \widehat{\mathbf{n}}_1 + (S_1 + S_2 \cos\theta_1) \cdot \widehat{\mathbf{n}}_3 \quad (2)$$

$${}^N_3\mathbf{P} = (S_2 \cdot \sin\theta_1 + S_3 \cdot \sin(\theta_1 + \theta_2)) \widehat{\mathbf{n}}_1 + (S_1 + S_2 \cos\theta_1 + S_3 \cdot \cos(\theta_1 + \theta_2)) \cdot \widehat{\mathbf{n}}_3 \quad (3)$$

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

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

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.

$$P = -m_1gs_1 - m_2g(s_1 + s_2\cos\theta_1) - m_3g(s_1 + s_2\cos\theta_1 + s_3\cos(\theta_1 + \theta_2)) + (0.5)k_{11}(s_1 + (1/2)\theta_1 - s_{01})^2 + (0.5)k_{21}(s_1 + (1/2)\theta_1 - s_{01})^2 + (0.5)k_{12}(s_2 + (1/2)\theta_2 - s_{02})^2 + (0.5)k_{22}(s_2 + (1/2)\theta_2 - s_{02})^2 + (0.5)k_{13}(s_3 + (1/2)\theta_3 - s_{03})^2 + (0.5)k_{23}(s_3 + (1/2)\theta_3 - s_{03})^2$$

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

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

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

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

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

$$Q_{s_3} = F_{13} + F_{23}$$

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

$$Q_{\theta_1} = (1/2)(F_{12} - F_{22}) + (1/2)(F_{13} - F_{23})$$

$$Q_{\theta_1} = (1/2)(F_{13} - F_{23})$$

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

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

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,

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

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

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

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

2.2 Backstepping Controller

The continuum robot dynamics in Equations (12) and (13) have the appropriate structure for the so-called backstepping controller design method. The steps in the backstepping controller design method are outlined in Figure 2. With the position error defined as $Z_1 = X_d - X_a$, all joints will track the desired state X_d if the error dynamics are given as follows:

$$(\dot{Z}_1 + [K_p]Z_1) = 0 \tag{18}$$

where $[K_p]$ is a positive definite gain matrix. The error dynamics in Equation (18) can be rewritten as:

$$X_2 = \dot{X}_d + [K_p]Z_1 \tag{19}$$

Substitution of Equation (19) into Equation (13) makes the position error dynamics go to zero. Since the state vector x_2 is not a control variable, Equation (19) cannot be directly substituted into Equation (13). The expression in Equation (19) is therefore defined as a fictitious control input and is labeled in Figure 2 and expressed below as X_{2d} .

$$X_{2d} = \dot{X}_{1d} + [K_p](X_d - X_a) \tag{20}$$

The fictitious control input in Equation (20) is selected as the specified velocity trajectory and hence the velocity error can be defined as $Z_2 = X_{2d} - X_{2a}$. With the following dynamics

$$(\dot{Z}_2 + [K_p]Z_2) = 0 \tag{21}$$

The joint position error will approach zero asymptotically, which will lead to the eventual asymptotic convergence of the joint position error. The error dynamics in equation (21) can be rewritten as:

$$X_2 = \dot{X}_d + [K_p]Z_2 \tag{22}$$

Substitution of Equation (21) into Equation (13) leads to the following expression as the desired stabilizing torque:

$$\tau = [H](\dot{X}_{2d} + [K_p]Z_2) + C(X_1, X_2) \tag{23}$$

The desired torque control input is a nonlinear compensator since it depends on the dynamics of the spherical motor. The time derivative of desired velocity vector is calculated using Equation (21). In terms of the desired state trajectory, and its time derivatives and the position and velocity state variables, the desired torque can be rewritten in following form:

$$\tau = [H]y + C(X_1, X_2) \tag{24}$$

Where

$$y = \ddot{X}_{1d} + ([K_p] + [K_d])(\dot{X}_{1d} - \dot{X}_1) + ([K_p][K_d]X_d - X_a) \tag{25}$$

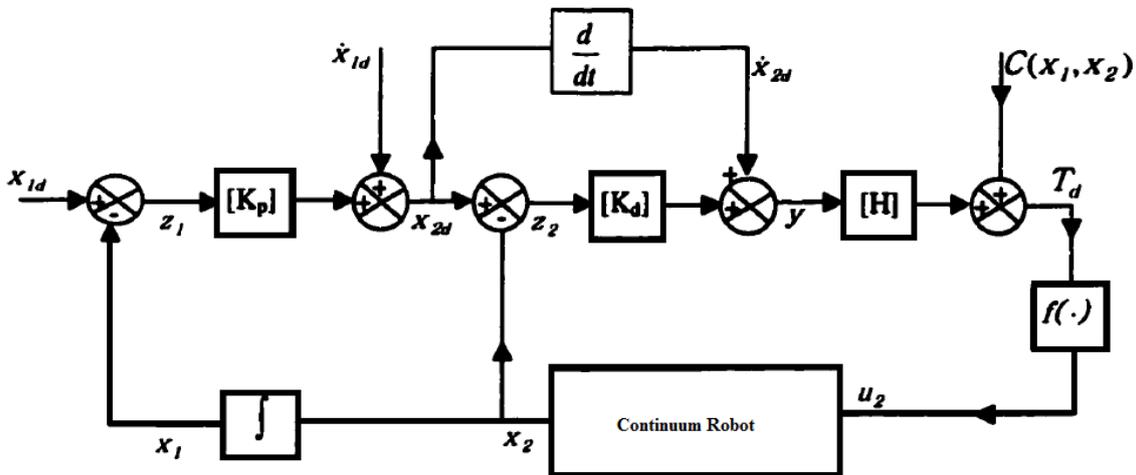


Fig. 2: Backstepping controller Block Diagram

The backstepping controller developed above is very similar to inverse dynamics control algorithm developed for robotic manipulators. The backstepping controller is ideal from a control point of view as the nonlinear dynamics of the continuum robot are cancelled and replaced by linear subsystems. The drawback of the backstepping controller is that it

requires perfect cancellation of the nonlinear continuum robot dynamics. Accurate real time representations of the robot dynamics are difficult due to uncertainties in the system dynamics resulting from imperfect knowledge of the robot mechanical parameters; existence of unmodeled dynamics and dynamic uncertainties due to payloads. The requirement for

perfect dynamic cancellation raises sensitivity and robustness issues that are addressed in the design of a robust backstepping controller. Another drawback of the backstepping controller is felt during real-time implementation of the control algorithm. Implementation of the backstepping controller requires the computation of the exact robot dynamics at each sampling time. This computational burden has an effect on the performance of the control algorithm and imposes constraints on the hardware/software architecture of the control system. By only computing the dominant parts of the robot dynamics, this computational burden can be reduced. These drawbacks of the backstepping controller makes it necessary to consider control algorithms that compensate for both model uncertainties and for approximations made during the on-line computation of robot dynamics. The next section provides robust modifications of the backstepping controller described in this section.

2.3 Robust Backstepping Control

When there are uncertainties in the spherical motor dynamics due to modeling inaccuracy and computational relaxation, robust controllers are ideal for ensuring system stability. When the system dynamics are completely known, the required torque control vector for the control of the spherical motor is given by Equations (24) and (25).

In the presence of modeling uncertainties, a reasonable approximation of the torque control input vector is given by

$$\mathbf{T}_d = [\hat{\mathbf{H}}]\mathbf{y} + \hat{\mathbf{C}} \quad (26)$$

where $[\hat{\mathbf{H}}]$ and $\hat{\mathbf{C}}$ are estimates of the inertia and coriolis terms in the spherical motor dynamics; and \mathbf{y} is given by Equation (25). The uncertainty on the estimates are expressed as

$$[\tilde{\mathbf{H}}] = [\hat{\mathbf{H}}] - [\mathbf{H}] \quad (27)$$

$$[\tilde{\mathbf{C}}] = [\hat{\mathbf{C}}] - [\mathbf{C}] \quad (28)$$

These uncertainties account for both imperfect modeling and intentional computational simplification. Application of the approximate control vector given by Equations (26) into Equation (10) leads to the following expression for the closed loop dynamics:

$$[\mathbf{H}]\dot{\mathbf{x}}_2 + \mathbf{C} = [\hat{\mathbf{H}}]\mathbf{y} + \hat{\mathbf{C}} \quad (29)$$

Since the inertia matrix $[\mathbf{H}]$ is symmetric and positive definite, the closed loop dynamics in Equation (29) can be rewritten as

$$\dot{\tilde{\mathbf{x}}}_1 = \mathbf{y} - \boldsymbol{\eta} \quad (30)$$

where

$$\boldsymbol{\eta} = ([\mathbf{I}] - [\mathbf{H}]^{-1}[\hat{\mathbf{H}}])\mathbf{y} - [\mathbf{H}]^{-1}\hat{\mathbf{C}} \quad (31)$$

Substitution of Equation (25) for \mathbf{y} in Equation (30) results in the following expression for the closed loop error dynamics.

$$\ddot{\tilde{\mathbf{x}}}_1 + ([\mathbf{K}_p] + [\mathbf{K}_d])\dot{\tilde{\mathbf{x}}}_1 + [\mathbf{K}_p][\mathbf{K}_d]\tilde{\mathbf{x}}_1 = \boldsymbol{\eta} \quad (32)$$

Defining a new error state vector,

$$\boldsymbol{\xi} = \begin{bmatrix} \tilde{\mathbf{x}}_1 \\ \dot{\tilde{\mathbf{x}}}_1 \end{bmatrix} \quad (33)$$

the error dynamics in Equation (32) can be expressed as

$$\dot{\boldsymbol{\xi}} = [\mathbf{F}]\boldsymbol{\xi} + [\mathbf{D}](\ddot{\mathbf{x}}_{1d} - \mathbf{y} + \boldsymbol{\eta}) \quad (34)$$

Where $[\mathbf{F}] = \begin{bmatrix} [\mathbf{0}] & [\mathbf{1}] \\ [\mathbf{0}] & [\mathbf{0}] \end{bmatrix}$ and $[\mathbf{D}] = \begin{bmatrix} [\mathbf{0}] \\ [\mathbf{1}] \end{bmatrix}$ are block matrices of dimensions $R^{6 \times 6}$ and $R^{6 \times 3}$ respectively. Since $\boldsymbol{\eta}$ is a nonlinear function of the position and velocity state vectors, the system error dynamics in the above equation are nonlinear and coupled. The backstepping controller developed in the previous section cannot guarantee system stability. The Lyapunov direct method is, however, used to design an outer feedback loop on the error dynamics that compensates for the system uncertainty contributed by $\boldsymbol{\eta}$.

2.4 Fuzzy Inference Engine:

This section provides a review about foundation of fuzzy logic based on [15-20]. Supposed that U is the universe of discourse and x is the element of U , therefore, a crisp set can be defined as a set which consists of different elements (x) will all or no membership in a set. A fuzzy set is a set that each element has a membership grade, therefore it can be written by the following definition;

$$\mathbf{A} = \{x, \mu_A(x) | x \in X\}; \mathbf{A} \in U \quad (35)$$

Where an element of universe of discourse is x , μ_A is the membership function (MF) of fuzzy set. The membership function ($\mu_A(x)$) of fuzzy set A must have a value between zero and one. If the membership function $\mu_A(x)$ value equal to zero or one, this set change to a crisp set but if it has a value between zero and one, it is a fuzzy set. Defining membership function for fuzzy sets has divided into two main groups; namely;

numerical and functional method, which in numerical method each number has different degrees of membership function and functional method used standard functions in fuzzy sets. The membership function which is often used in practical applications includes triangular form, trapezoidal form, bell-shaped form, and Gaussian form.

Linguistic variable can open a wide area to use of fuzzy logic theory in many applications (e.g., control and system identification). In a natural artificial language all numbers replaced by words or sentences.

If – then Rule statements are used to formulate the condition statements in fuzzy logic. A single fuzzy *If – then* rule can be written by

$$\text{If } x \text{ is } A \text{ Then } y \text{ is } B \tag{36}$$

Where *A* and *B* are the Linguistic values that can be defined by fuzzy set, the *If – part* of the part of “*x is A*” is called the antecedent part and the *then – part* of the part of “*y is B*” is called the Consequent or Conclusion part. The antecedent of a fuzzy if-then rule can have multiple parts, which the following rules shows the multiple antecedent rules:

$$\text{if } e \text{ is } NB \text{ and } \dot{e} \text{ is } ML \text{ then } T \text{ is } LL \tag{37}$$

Where *e* is error, \dot{e} is change of error, *NB* is Negative Big, *ML* is Medium Left, *T* is torque and *LL* is Large Left. *If – then* rules have three parts, namely, fuzzify inputs, apply fuzzy operator and apply implication method which in fuzzify inputs the fuzzy statements in the antecedent replaced by the degree of membership, apply fuzzy operator used when the antecedent has multiple parts and replaced by single number between 0 to 1, this part is a degree of support for the fuzzy rule, and apply implication method used in consequent of fuzzy rule to replaced by the degree of membership. The fuzzy inference engine offers a mechanism for transferring the rule base in fuzzy set which it is divided into two most important methods, namely, Mamdani method and Sugeno method. Mamdani method is one of the common fuzzy inference systems and he designed one of the first fuzzy controllers to control of system engine. Mamdani’s fuzzy inference system is divided into four major steps: fuzzification, rule evaluation, aggregation of the rule outputs and defuzzification. Michio Sugeno uses a singleton as a membership function of the rule consequent part. The following definition shows the Mamdani and Sugeno fuzzy rule base

$$\begin{aligned} \text{Mamdani } F.R^1: & \text{ if } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z \text{ is } C \\ \text{Sugeno } F.R^1: & \text{ if } x \text{ is } A \text{ and } y \text{ is } B \text{ then } f(x, y) \text{ is } C \end{aligned} \tag{38}$$

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. Two most common methods that used in fuzzy logic controllers are Max-min aggregation and Sum-min aggregation. Max-min aggregation defined as below

$$\begin{aligned} \mu_U(x_k, y_k, U) &= \mu_{\cup_{i=1}^r FR^i}(x_k, y_k, U) \\ &= \max \left\{ \min_{i=1}^r [\mu_{R_{pq}}(x_k, y_k), \mu_{p_m}(U)] \right\} \end{aligned} \tag{39}$$

The Sum-min aggregation defined as below

$$\begin{aligned} \mu_U(x_k, y_k, U) &= \mu_{\cup_{i=1}^r FR^i}(x_k, y_k, U) \\ &= \sum \min_{i=1}^r [\mu_{R_{pq}}(x_k, y_k), \mu_{p_m}(U)] \end{aligned} \tag{40}$$

Where *r* is the number of fuzzy rules activated by x_k and y_k and also $\mu_{\cup_{i=1}^r FR^i}(x_k, y_k, U)$ is a fuzzy interpretation of *i – th* rule. 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, which *COG* method used the following equation to calculate the defuzzification

$$COG(x_k, y_k) = \frac{\sum_i U_i \sum_{j=1}^r \mu_u(x_k, y_k, U_i)}{\sum_i \sum_{j=1}^r \mu_u(x_k, y_k, U_i)} \tag{41}$$

And *COA* method used the following equation to calculate the defuzzification

$$COA(x_k, y_k) = \frac{\sum_i U_i \cdot \mu_u(x_k, y_k, U_i)}{\sum_i \mu_u(x_k, y_k, U_i)} \tag{42}$$

Where $COG(x_k, y_k)$ and $COA(x_k, y_k)$ illustrates the crisp value of defuzzification output, $U_i \in U$ is discrete element of an output of the fuzzy set, $\mu_u(x_k, y_k, U_i)$ is the fuzzy set membership function, and *r* is the number of fuzzy rules.

Based on foundation of fuzzy logic methodology; fuzzy logic controller has played important rule to design nonlinear controller for nonlinear and uncertain systems [20-32]. 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).

III. Methodology

This step is focused on the design Mamdani’s fuzzy backstepping controller with application to continuum robot manipulator. As mentioned above pure backstepping controller has nonlinear dynamic parameters limitation in presence of uncertainty and external disturbances. In order to solve this challenge this work applied Mamdani’s fuzzy inference engine estimator in backstepping controller. The backstepping method is based on mathematical formulation which this method is introduced new variables into it in form depending on the dynamic equation of continuum robot arm. This method is used as feedback linearization in order to solve nonlinearities in the system. In this research fuzzy methodology is used to estimate the part of nonlinearity term in backstepping controller. The backstepping controller for continuum robot is calculated by;

$$U_{B,S} = U_{eq_{B,S}} + M \cdot I \quad (43)$$

Where $U_{B,S}$ is backstepping output function, $U_{eq_{B,S}}$ is backstepping nonlinear equivalent function which can be written as (58) and I is backstepping control law which calculated by (59)

$$U_{eq_{B,S}} = [B + C + G] \quad (44)$$

We have

$$I = [\ddot{\theta} + K_1(K_1 - 1) \cdot e + (K_1 + K_2) \cdot \dot{e}] \quad (45)$$

Based on (11) and (58) the fuzzy backstepping filter is considered as

$$(B + C + G) = \sum_{l=1}^M \theta^l \zeta(x) - \lambda S - K \quad (46)$$

Based on (44) the formulation of fuzzy backstepping filter can be written as;

$$U = U_{eq_{B,S}fuzzy} + MI \quad (47)$$

Where $U_{eq_{B,S}fuzzy} = [(B + C + G)] + \sum_{l=1}^M \theta^l \zeta(x) + K$

Most robust control designs are based on the assumption that even though the uncertainty vector η is unknown, some information is available on its bound.

In designing the robust backstepping controller, a robust term that compensates for the uncertainty in the system dynamics is added to the control law in Equations (25) and (26). The robust control law, therefore, has the following form:

$$T_d = [\hat{H}] + \hat{C} \quad (48)$$

$$y = \ddot{\tilde{x}}_1 + ([K_p] + [K_d])\dot{\tilde{x}}_1 + [K_p][K_d]\tilde{x}_1 + w \quad (49)$$

The first three terms in Equation (45) ensures the stabilization of the error dynamics in the absence of system uncertainties. The proper choice of w ensures the stability of the system even in the presence of uncertainties. Figure 3 shows a block diagram detailing the steps in the robust fuzzy backstepping control of continuum robot.

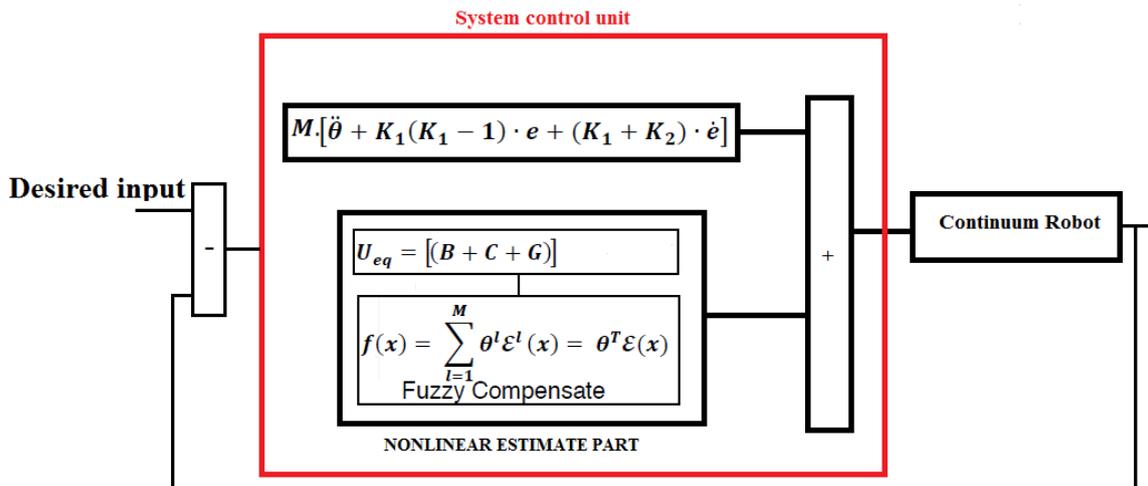


Fig. 3: Block diagram of fuzzy backstepping for continuum robot

Using the control law in Equation (45) and Equation (46), the system error dynamics in Equation (44) can be rewritten in the following form:

$$\dot{\xi} = [\tilde{F}]\xi + [D](\eta - w)$$

where

$$[\tilde{F}] = ([F] - [D][K]) = \begin{bmatrix} [0] & [I] \\ -[K_p][K_d] & -[K_p] - [K_d] \end{bmatrix} \quad (50)$$

$$K = [[K_p][K_d] \quad [K_p] + [K_d]]^T$$

When there are no uncertainties in the system dynamics, the robust term in the applied controller return is identically zero and the control law returns to the backstepping control developed in the previous section. Figure 2 shows a block diagram of the robust backstepping controller.

The stability considerations and the design for the robust controller are determined using Lyapunov methods. The following positive definite quadratic function is selected as a Lyapunov function:

$$V_1 = \xi^T [Q] \xi > 0 \quad \forall \xi \neq 0 \quad (51)$$

Where $[Q] \in \mathbf{R}^{6 \times 6}$ is a positive definite matrix. The time derivative of the Lyapunov function along the trajectories of the error dynamics results in the following equation:

$$\dot{V}_1 = \xi^T \left([\tilde{F}]^T [Q] + [Q][\tilde{F}] \right) \xi + 2\xi^T [Q][D](\eta - w) \quad (52)$$

Since, for a negative definite matrix $[\tilde{F}]$; a positive definite matrix $[Q]$; and a symmetric positive definite matrix $[P]$, the following identity holds;

$$[\tilde{F}]^T [Q] + [Q][\tilde{F}] = -[P] \quad (53)$$

The time derivative of the Lyapunov function in Equation (49) can therefore be rewritten as

$$\dot{V}_1 = -\xi^T [P] \xi + 2\xi^T [Q][D](\eta - w) \quad (54)$$

The time derivative of the Lyapunov function is strictly negative definite if the second term in the above equation is either negative or zero. The error dynamics converge to zero when the time derivative of the Lyapunov function in Equation (55) is negative definite. Selecting the robust control term w as

$$w = \frac{\rho}{\|z\|} z \quad \rho > 0 \quad (55)$$

Where $z = [D]^T [Q] \xi$ substituting it into Equation (49) results in the following expression for the time derivative of the Lyapunov function.

$$\dot{V}_1 = -\xi^T [P] \xi + 2z^T \left(\eta - \frac{\rho}{\|z\|} z \right) \quad (56)$$

According to the following inequality,

$$z^T (\eta - w) \leq \|z\| \|\eta\| - \rho \|z\| \quad (57)$$

$z^T \left(\eta - \frac{\rho}{\|z\|} z \right)$ in Equation (43) is always bounded from above by $\|z\|(\|\eta\| - \rho)$. Therefore, selecting ρ such that

$$\rho \geq \|\eta\| \quad \forall x_1 \dot{x}_1 \ddot{x}_{1d} \quad (58)$$

Results in the negative definiteness of the time derivative of the Lyapunov function along the dynamics of ξ .

With the definition of η given in Equation (31), the following bound can be put on the uncertainty term η .

$$\|\eta\| = \|I - [H]^{-1}[\hat{H}]\|(\|\dot{x}_{1d}\| + \|[K]\|\|\xi\| + \|[w]\|) + \|[H]^{-1}\|\|\tilde{\eta}\| \leq \kappa Q_M + \kappa\|[K]\|\|\xi\| + \kappa\rho + [H_M]\theta \quad (59)$$

Using the inequalities in Equations (58) and (59), ρ is solved to give the following relationship

$$\rho \geq \frac{1}{1-\kappa} (\kappa Q_M + \kappa\|[K]\|\|\xi\| + [H_M]\theta) \quad (60)$$

The choice of ρ from Equation (60) ensures that the time derivative of the Lyapunov function is negative definite. This results in the stability of the error dynamics.

$$\dot{V}_1 = -\xi^T [P] \xi + 2z^T \left(\eta - \frac{\rho}{\|z\|} z \right) < 0 \quad \forall \xi \neq 0 \quad (61)$$

IV. Results and Discussion

Robust fuzzy backstepping controller (FBSC) was tested to Step response trajectory. In this simulation, to control position of continuum robot the first, second, and third joints are moved from home to final position without and with external disturbance. The simulation was implemented in MATLAB/SIMULINK environment. 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 and applied to nonlinear dynamic of these controllers.

Tracking performances: Figure 4 shows tracking performance for FBSC and BSC without disturbance.

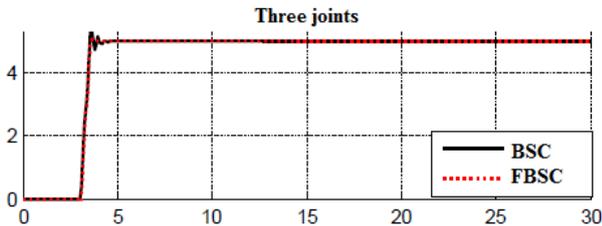


Fig. 5: FBSC and BSC for first, second and third joint trajectory

By comparing step response trajectory without disturbance in BSC and FBSC, it is found that the FBSC's overshoot (1.4%) is lower than BSC's (1.6%).

Disturbance rejection: Figure 5 has shown the power disturbance elimination in BSC and FBSC. The main target in this controller is disturbance rejection as well as the other responses. A band limited white noise with predefined of 40% the power of input signal is applied to BSC and FBSC. It found fairly fluctuations in trajectory responses. As mentioned earlier, BSC works very well when all parameters are known, or we have a limitation uncertainty in parameters.

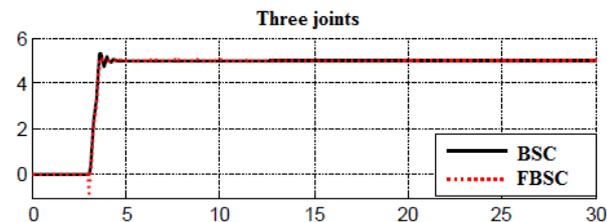


Fig. 5: FBSC and BSC for first, second and third joint trajectory with disturbance

Among above graph (5) relating to step trajectory following with external disturbance, BSC and FBSC have fairly fluctuations. By comparing some control parameters such as overshoot, rise time, steady state and RMS error it is computed that the FBSC's overshoot (1.6%) is lower than BSC's (3%). Both of them have about the same rise time; FBSC (0.5 sec) and BSC (0.49 sec).

V. Conclusion

Continuum robot is a nonlinear high degree serial robot. The dynamic parameters of this system are highly nonlinear. To control of this system robust backstepping methodology is introduced. Backstepping controller

(BSC) is an influential robust nonlinear controller to certain and partly uncertain systems. When all dynamic and physical parameters are known BSC works superbly; practically a large amount of systems have uncertainties and fuzzy model base BSC is used. FBSC is a mathematical model base method to off-line control of highly nonlinear systems such as continuum robot.

References

- [1] G. Robinson, and J. Davies, "Continuum robots – a state of the art," Proc. IEEE International Conference on Robotics and Automation, Detroit, MI, 1999, vol. 4, pp. 2849-2854.
- [2] I.D. Walker, D. Dawson, T. Flash, F. Grasso, R. Hanlon, B. Hochner, W.M. Kier, C. Pagano, C.D. Rahn, Q. Zhang, "Continuum Robot Arms Inspired by Cephalopods, Proceedings SPIE Conference on Unmanned Ground Vehicle Technology VII, Orlando, FL, pp 303-314, 2005.
- [3] K. Suzumori, S. Iikura, and H. Tanaka, "Development of Flexible Microactuator and it's Applications to Robotic Mechanisms", Proceedings IEEE International Conference on Robotics and Automation, Sacramento, California, pp. 1622-1627, 1991.
- [4] D. Trivedi, C.D. Rahn, W.M. Kier, and I.D. Walker, "Soft Robotics: Biological Inspiration, State of the Art, and Future Research", Applied Bionics and Biomechanics, 5(2), pp. 99-117, 2008.
- [5] W. McMahan, M. Pritts, V. Chitrakaran, D. Dienno, M. Grissom, B. Jones, M. Csencsits, C.D. Rahn, D. Dawson, and I.D. Walker, "Field Trials and Testing of "OCTARM" Continuum Robots", Proc. IEEE International Conference on Robotics and Automation, pp. 2336-2341, 2006.
- [6] W. McMahan, I.D. Walker, "Octopus-Inspired Grasp Synergies for Continuum Manipulators", Proc. IEEE International Conference on Robotics and Biomimetics, pp. 945- 950, 2009.
- [7] Farzin Piltan, M. Akbari, M. Piran , M. Bazregar. "Design Model Free Switching Gain Scheduling Baseline Controller with Application to Automotive Engine", International Journal of Information Technology and Computer Science, 01:65-73, 2013.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] Kokotovic, P., "The Joy of Feedback: Nonlinear and Adaptive," *IEEE Control System Magazine*, vol. 12, pp. 7-17, June 1992.
- [14] Qu, Z., and Dawson, D. M., "Lyapunov Direct Design of Robust Tracking Control for Classes of Cascaded Nonlinear Uncertain Systems without Matching Conditions," *Proceedings of the 31st IEEE Conference on Decision and Control*, Brighton, UK, 1991, vol. 3, pp. 2521-2526.
- [15] Carroll, J. J., and Dawson, D. M., "Integrator Backstepping Techniques for the Tracking Control of Permanent Magnet Brush DC Motors," *IEEE Transactions on Industry Applications*, Vol. 31, No.2, Mar.-Apr. 1995, pp. 248-255
- [16] Hemati, N., Thorp, J., and Leu, M., "Robust Nonlinear Control of Brushless de motors for direct-drive robotic applications," *IEEE Transactions on Industrial Electronics*, vol. 37, no. 6 pp. 460-468, Dec., 1990.
- [17] Carroll, J., and Dawson, D. M., "Robust tracking control of a brushless de motor with application to robotics," *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Atlanta, GA., pp. 94-99, May, 1993.
- [18] Die'-Spong, M., Marino, R., Peresada, S. M., and Taylor, D.O., "Feedback Linearizing Control of Switched Reluctance Motors," *IEEE Transactions on Automatic Control*, Vol. AC-32, No.5, pp. 371-379, May 1987.
- [19] Taylor, D., Die'-Spong, M., Marino, R., and Peresada, S., "A Feedback Linearizing Control for Direct-drive Robots with Switched Reluctance Motors," *Proceedings of the 25th IEEE Conference on Decision and Control*, pp. 388-396, Dec., 1996.
- [20] Carroll, J., Dawson, D. M., and Qu. Z., "Adaptive tracking control of a switched reluctance motor turning an inertial load," *Proceedings of the Amer. Control Conf.*, San Francisco, CA., pp 670-674, June 1993.
- [21] Bodson, M., Chiasson, R. N., and Rekowaki, R., "High-perfonnance Nonlinear Feedback Control of a Permanent Magnet Stepper Motor," *IEEE Transactions on Control System Technology*, Vol. 1, No.1, pp5-14, Mar., 1993.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] Farzin Piltan, B. Boroomand, A. Jahed and H. Rezaie. "Performance-Based Adaptive Gradient Descent Optimal Coefficient Fuzzy Sliding Mode

Methodology”, International Journal of Intelligent Systems and Applications, 11: 40-52 2012.

- [29] 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.
- [30] 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.
- [31] Farzin Piltan, A. Jahed, H. Rezaie and B. Boroomand. ” Methodology of Robust Linear On-line High Speed Tuning for Stable Sliding Mode Controller: Applied to Nonlinear System”, International Journal of Control and Automation, 5(3): 217-236, 2012.
- [32] Farzin Piltan, Mojdeh Piran, Mansour Bazregar, Mehdi Akbari " Design High Impact Fuzzy Baseline Variable Structure Methodology to Artificial Adjust Fuel Ratio", International Journal of Intelligent Systems and Applications(IJISA) , 5(2):59-70, 2013.

to FPGA, robotics and artificial nonlinear control and IC engine modeling and control.



Hossein Hashemzadeh is a computer engineer researcher. His research activities deal with the information technology, nano technology, artificial control system.



Alireza Hasiri is an electronic engineer researcher. His research activities deal with the Nonlinear Control, Sensors and Instrumentation, Artificial Intelligence.



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

Authos' 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 and he is also a manager of Rohan Abzar Javdan Mehr corporation. 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, optimization, fuzzy theory, neural network, energy systems, and computer vision.



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. He is editorial board of international journal of control and automation (IJCA), editorial board of International Journal of Intelligent System and Applications (IJISA), editorial board of IAES international journal of robotics and automation, editorial board of International Journal of Reconfigurable and Embedded Systems and reviewer of (CSC) international journal of robotics and automation. His main areas of research interests are nonlinear control, artificial control system and applied