Jurnal
Teknologi

Full Paper

SELF-TUNING PID CONTROLLER DESIGN USING FUZZY LOGIC FOR A SINGLE-LINK FLEXIBLE JOINT ROBOT MANIPULATOR

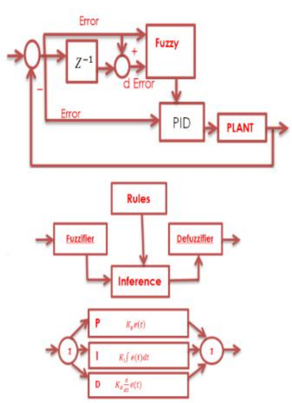
Ali Dehghani*, Hamed khodadadi

Department of Electrical Engineering, Khomeini shahr Branch, Islamic Azad University, Isfahan, Iran

Article history
Received
16 November 2015
Received in revised form
24 March 2016
Accepted
24 March 2016

*Corresponding author
ali.dehghani@iaukhsh.ac.ir

Graphical abstract



Abstract

Although flexible joint robots are widely used in the industry, they are not without problems. It is especially so in their joints, links and complex dynamic where the interaction between loops, non-linearity, and flexibility in the joints can be difficult. The purpose of the present paper is to improve the tracking performance of flexible joint robots. Therefore the physical relations of the system dynamics need to be used to determine a non-linear model for the flexible joint robot. This paper attempts to achieve the desired performance flexible joint robot based on Fuzzy Logic Self-Tuning PID controller. Generally, the classic PID controller is different from the newly introduced form of PID. In classic PID, the parameter values are calculated based on various methods such as Ziegler-Nichols, while in fuzzy logic self-tuning PID, they are obtained by intelligent methods such as fuzzy logic. After deriving the system model, this logic self-tuning PID controller is designed in two cases: using error and its derivative and employing error and its integral for the inputs. The simulation results indicate that the proposed controllers can improve the overall efficiency of the system.

Keywords: Single-link flexible-joint manipulator; PID; fuzzy logic self-tuning PID; nonlinear model

1.0 INTRODUCTION

Many modern control techniques for robotic system have been proposed in recent years [1, 2]. They use new approaches, considering the interaction between loops to determine the behavior of the system. Most of these controllers are designed on rigid base robots. However, using rigid base control methods for flexible robots may cause instability. Hence, to improve control approaches, the effect of flexibility must be considered. It is expected to improve the performance of the mechanical arms using low weight mechanical structures for flexible robots. The main purpose of the controller design is to place the position of the arms accurately, despite the existence of bending and vibrating [3-5]. Basically, flexible robots use less energy, make faster moves, reach farther and increase load capacity [5]. However, the flexible link makes technical problems, for instance the flexibility of the robot arms. Therefore, the exact control is more difficult. Probably, the first step to design and control the robot with flexible link is dedicated to [6], they considered that the flexibility of the link is established in only one direction like vertical axis. Hence, they designed a Linear Quadratic Gaussian (LQG) controller for controlling the robot position. In [1], the authors tried to control the position of single link flexible robot which rotates on horizontal axis. An optimal PD controller for a non-rigid robot with two links are designed in [7]. Its purpose was high speed positioning of the robot link. In [8], a complex controller with Linear Quadratic Regulator (LQR) and Fuzzy controller were presented. A nonlinear control strategy based on energy (with Lyapunov function) for a robot with two flexible links was presented by [9]. A robust control method via applying neural network has also been studied in [10]. Fuzzy logic based on PID controller in the presence of uncertainty condition was designed by [11]. Adaptive control, impedance control, and Model Predictive Control (MPC) are the other methods that were raised in the related literature. This paper goes on to introduce flexible joint robots with flexible link. Then, nonlinear robot modeling based on dynamic relations is derived. Robotic systems with flexible links follow the nonlinear ordinary differential equations. The intended structure for robot controlling is Fuzzy Logic Self-Tuning PID method. Accordingly, the first section of this paper after presenting the introduction turns to introduce the system model. In the next section, a classic PID controller is designed and evaluated. Focuses on the structure of self-tuning PID based on fuzzy logic comes in the next part. This controller is evaluated in two cases, utilizing the error integral and error derivate. We concentrate on the better step response by using Fuzzy Logic Self-Tuning PID. Finally, the simulation result and comparison will be illustrated and a conclusion is made.

2.0 SYSTEM MODELING

System modeling is based on Lagrange method and the equations of the system are based on conservation of energy. Initially, we obtain the kinetic energy and potential of all system components [11-13]. So:

$$L = K - P \tag{1}$$

Where K is Kinetic energy and P is Potential energy. Using Lagrange equations, it reads as:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\alpha}} - \frac{\partial L}{\partial \alpha} = 0 \tag{2}$$
$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = \tau$$

$$J_l\ddot{\theta} + J_l\ddot{\alpha} + K_s\alpha - \text{mgh}\,\text{Sin}(\theta+\alpha)\text{=}0 \tag{3}$$
$$(J_h + J_l)\ddot{\theta} + J_l\ddot{\alpha} + K_s\alpha\text{-mghSin}(\,\theta+\alpha)\text{ =r}$$

A relation between voltage and torque is presented as:

$$v = (J_h + (i\,R_m + K_m\,K_g\,\omega) \tag{4}$$
$$\text{i=}\frac{v}{R_m} - \frac{K_m\,K_g}{R_m}\omega$$

ω is the angular velocity of the motor, hence armature current is:

$$i = \frac{\tau}{K_m\,K_g}, \qquad \dot{\theta} = \omega \tag{5}$$

And the relation between motor torque and voltage is:

$$\tau = \frac{K_m K_g}{R_m}v - \frac{K_m^2 K_g^2}{R_m}\,\dot{\theta} \tag{6}$$

By choosing state variable as below:

$$x_1 = \theta \rightarrow \dot{x}_1 = \dot{\theta} \tag{7}$$
$$x_2 = \alpha \rightarrow \dot{x}_2 = \dot{\alpha}$$
$$x_3 = \dot{\theta} \rightarrow \dot{x}_3 = \ddot{\theta} = \dot{\omega}$$
$$x_4 = \dot{\alpha} \rightarrow \dot{x}_4 = \ddot{\alpha}$$

The system can be presented as [11, 12]:

$$\dot{x}_1 = x_3 \tag{8}$$
$$\dot{x}_2 = x_4$$
$$\dot{x}_3 = \frac{k_s}{j_h}x_2 + \frac{K_g K_m}{j_h R_m}v - \frac{{K_g}^2{K_m}^2}{j_h R_m}x_3$$
$$\dot{x}_4 = -\frac{k_s}{j_h}x_2 - \frac{K_g K_m}{j_h R_m}v + \frac{{K_g}^2{K_m}^2}{j_h R_m}x_3 - \frac{k_s}{j_l}x_2 + \frac{mgh}{j_l}sin(x_1 + x_2)$$

To drive a numerical model, the values of the system's parameters are replaced using Table 1.

Table 1 the value of system parameter [12]

Parameter	Symbol	Value
Load Inertia	J_l	0.0059kgm^2
Inertia of hub	j_h	0.0021kgm^2
Link Mass	M	0.403 [kg]
Height of C.M.	H	0.06 [m]
Spring Stiffness	K_s	1.61 [N/m]
Motor Const.	K_m	0.00767[N/rad/s]
Gear Ratio	K_g	70

After replacing the table's parameter values, the state space equation can be derived as:

$$\dot{x}_1 = x_3$$
$$\dot{x}_2 = x_4$$
$$\dot{x}_3 = 767.05x_2 - 52.795x_3 + 98.3v \tag{9}$$
$$\dot{x}_4 = -1040.1x_2 + 52.795x_3 + 40.2sin(x_1 + x_2) - 98.3v$$

3.0 DESIGN THE PID CONTROLLER

PID refers to Proportional_Integral_Derivative. This is a type of feedback controller whose the output, a Control Variable (CV), is generally based on the error between some user-defined set point (SP) and some measured process variable (PV) [14]. The classic PID controller is a very popular controller, used in almost all system. This type of compensator is based on the bellow formula that computes the control signal.

$$u = K_p e(t) + K_i\int e(t)dt + K_d\frac{d}{dt}e(t) \tag{10}$$
$$e(t) = r(t) - y(t) \tag{11}$$

The structure of the PID controller and its controller on the system is shown in the figure below:

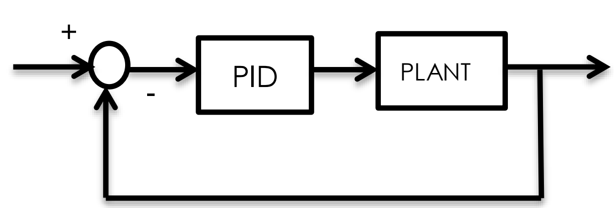


Figure1 Closed loop control schematic [15]

The performance of PID controller is shown in the schematic below:

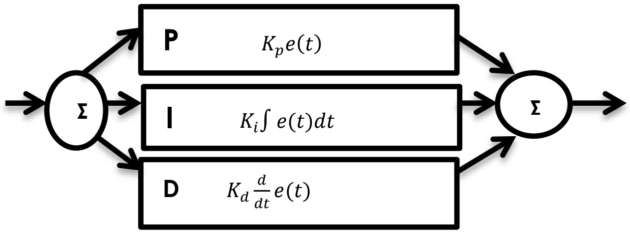


Figure 2 The structure of PID controller

Where $e(t)$ is error signal and K_p, K_i, K_d is a controller parameter that should be computed and tuned. K_i is integral term, K_d is derivative term and K_p is proportional term. These parameters are part of the control signal (*u (t)*) so there are some effects on the system's response (shown in Table 2).

Table 2 Effect of PID parameter on system response

	Rise Time	Overshoot	Settling Time	Steady State Error
K_p	Decrease	Increase	Small Increase	Decrease
K_i	Small Decrease	Decrease	Decrease	Small Change
K_d	Small Decrease	Increase	Increase	Large Decrease

In order to gain a proper response, the controller should be able to stabilize the system. After simulating we obtain Figure 3.

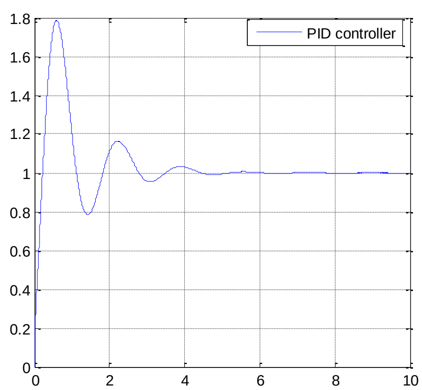


Figure 3 Step response of system by using PID controller

Table 3 Some transient parameters of PID response

	Overshoot	Settling time	Rise time
PID Controller	78%	6.50	0.20

The value of settling time is 6.50 seconds, Rise Time-and the time is 0.20 seconds and overshoot is %78. It is apparent that the step response of the PID controller is not optimal. Therefore, PID controller has the higher level of overshoot and long settling time. Hence, the PID controller is not an optimal controller.

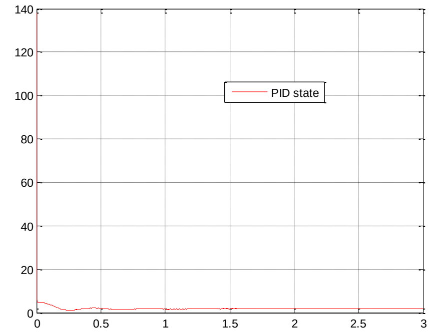


Figure 4 The control signal of the PID controller

4.0 SELF-TUNING PID CONTROLLER BASED ON FUZZY LOGIC

When the system condition is changed, the designed controller is not optimal. So the system response will be changed. One solution is to use self-tuning PID controller. There are many intelligent techniques to achieve a suitable and stable response. One of them is Fuzzy Logic which is a method set by Prof. Lotfi Zadeh. Fuzzy control is a control method based on fuzzy logic and it is based on human experience and strategy. Their principles are easy to understand and has been an active and fruitful research field since Mamdani and Assilian Pioneering work on fuzzy controller in 1974. The control process is shown below:

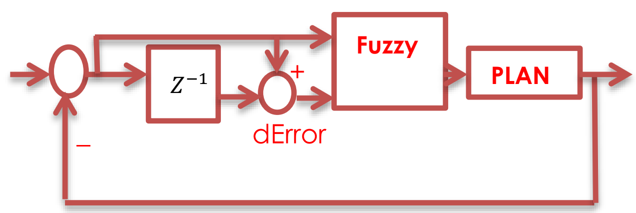


Figure 5 Fuzzy control process [15]

This controller has four important main parts which are fuzzification interface, knowledge base, inference mechanism and defuzzification. Fuzzification converts the inputs into suitable value The fuzzification subsystem; measures the values of input variables, performs a scale mapping that transfers the scope of values of input variables into corresponding universes of talks, performs the function of fuzzification that converts the input data into appropriate linguistic values which might be viewed as labels of fuzzy sets [16]. In the fuzzification process, a real scalar value changes into a fuzzy value The Defuzzification yield a non-fuzzy action from inference action. Defuzzification is taking the fuzzy outputs and converting them to a single or crisp output value. A rule base is a unit of making decision that simulates a human decision process from knowledge of control rules. In a fuzzy logic, a rule base is constructed to control the output variable. A fuzzy rule is a simple IF_THEN rule with a condition and conclusion. The inference mechanism uses fuzzy input variable to evaluate the control rule in that store in fuzzy rule base. In the first step of the design, we need to design the range of IMF (Input Membership Function) and OMF (Output Membership Function).

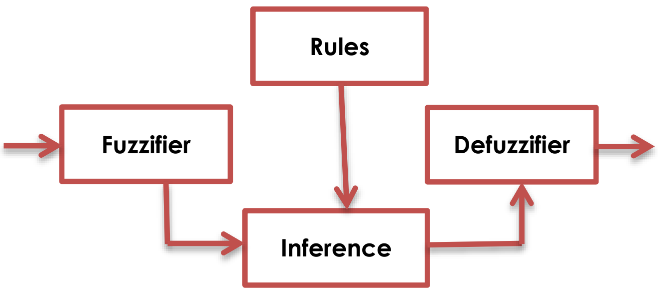


Figure 6 Structure of fuzzy control design [15]

In self-tuning PID controller, the fuzzy controller tunes the PID parameter (K_p, K_i, K_d) and prediction value of these parameters. The process of fuzzy PID shown in Figure 7:

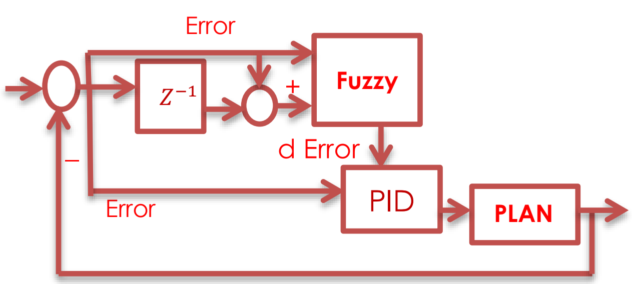


Figure 7 Structure of Fuzzy PID [15]

The Fuzzy Logic rules are the most important part of this controller [17,18]. The fuzzy PID control rules are shown in Table 4-6:

Table 4 Fuzzy rules for computing K_p

ve \ ce	NL	NS	ZE	PS	PL
NL	PVL	PVL	PVL	PVL	PVL
NS	PML	PML	PML	PML	PML
ZE	PVS	PVS	PS	PMS	PMS
PS	PML	PML	PML	PM	PM
PL	PVL	PVL	PVL	PVL	PVL

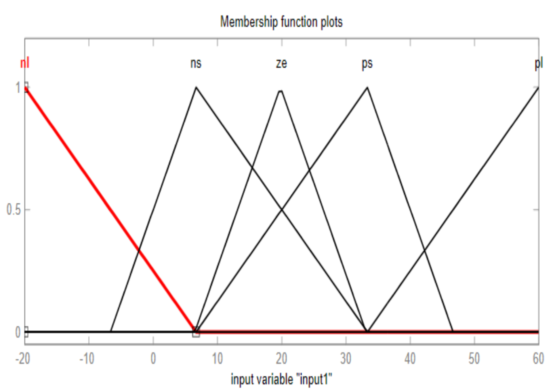
Table 5 Fuzzy rules for computing K_i

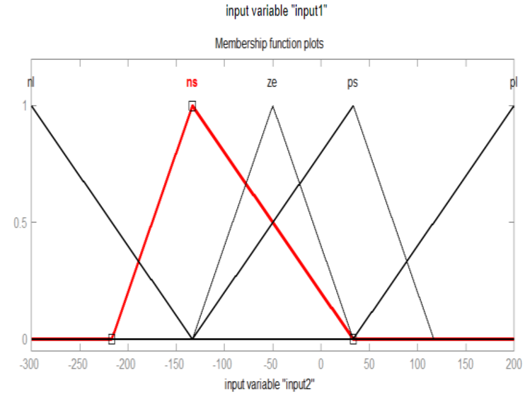
ve \ ce	NL	NS	ZE	PS	PL
NL	PM	PM	PM	PM	PM
NS	PMS	PMS	PMS	PMS	PMS
ZE	PS	PS	PVS	PS	PS
PS	PMS	PMS	PMS	PMS	PMS
PL	PM	PM	PM	PM	PM

Table 6 Fuzzy rules for computing K_d

ve \ ce	NL	NS	ZE	PS	PL
NL	PVL	PVL	PVL	PVL	PVL
NS	PMS	PMS	PMS	PMS	PMS
ZE	PS	PS	PVS	PS	PS
PS	PMS	PMS	PMS	PMS	PMS
PL	PM	PM	PM	PM	PM

According to the charts, NL (Negative Large), NS (Negative Small), ZE (Zero), PS (Positive Small), PL (Positive Large) are inputs and PVS (Positive very Small), PS (Positive Small), PML (Positive Medium Large), PM, PMS (Positive Medium Small), PL (Positive Large), PVL (Positive very Large) are as outputs. This Fuzzy control in Fuzzy Logic Self-Tuning PID (FLST-PID) has the task to tuning the K_p, K_i, K_d. This means that fuzzy has two inputs and three outputs. Five membership's triangular functions are shown below:





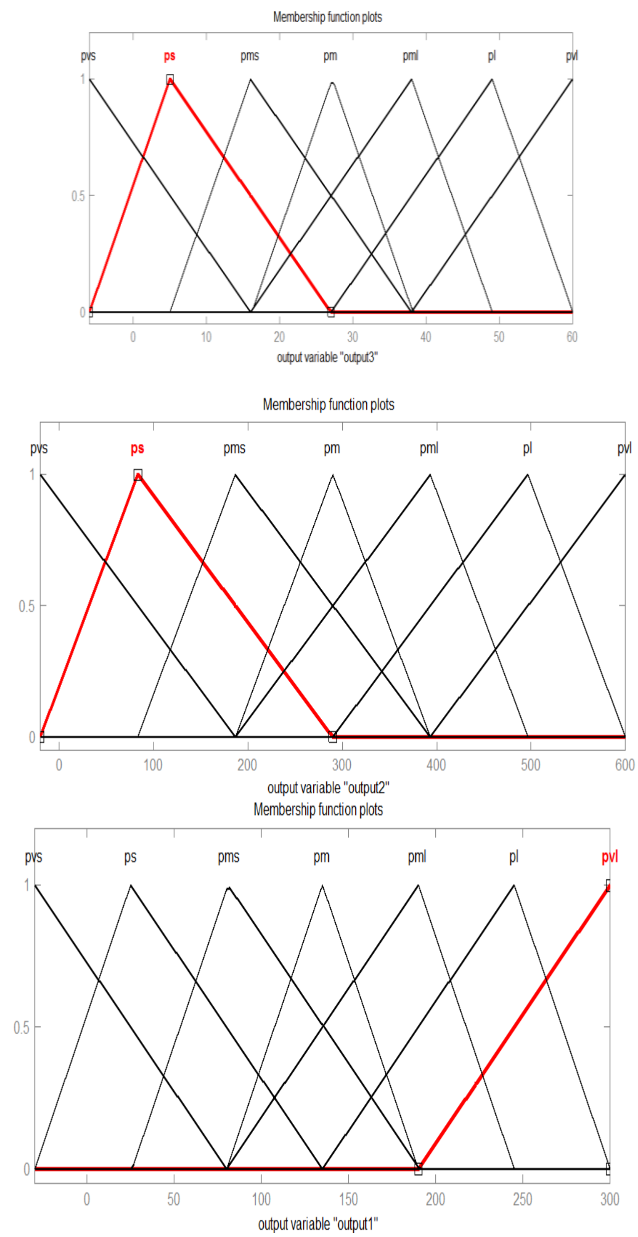


Figure 8 Inputs and outputs Function membership

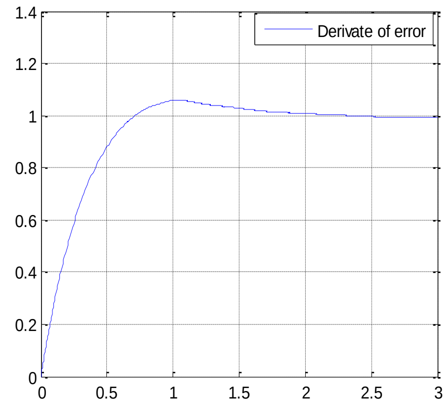


Figure 9 Step response of FL-ST PID by using Derivate of error

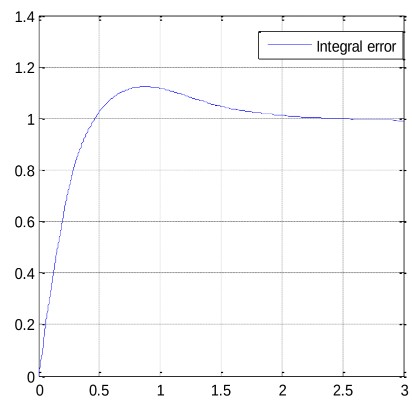


Figure 10 step response of FLST-PID by using Derivate of error

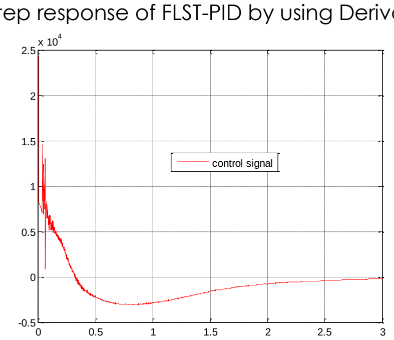


Figure 11 Control signal of FLST-PID

To have a good response, the controller should be able to stabilize the system. The present paper focuses on the structures of FLST-PID and evaluate two states; the derivate of error and the integral of error. The histogram in Figs 9 and 10 indicates this subject. The step response is plotted to determine the angle after finding the optimum PID values by using FLST controller. PID controller response simulated by MATLAB in bellow:

Based on the data in Table 2 and 7, the FLST-PID controller is better than PID because it has lower overshoot and lower settling time. Therefore, it has better speed response.

Table 7 Integral error and derivative error response

	Overshoot	settling time	Rise time
Integral error	10%	3	0.30
Derivative error	6%	3	0.50

5.0 CONCLUSION

In this paper, the PID controller and FLST-PID was designed for single link flexible joint robot**.** The foregoing discussion has attempted to suggest a self-tuning PID based on fuzzy logic to control the position in flexible joint robot. The purpose of doing control is the tracking of a desired trajectory with flexible link. The robot system is considered as a nonlinear model. After using PID controller we can see that it has a higher level of overshoot and long settling time. In the next part in order to achieve high speed and low overshoot value, FLST-PID is designed. It is used in two states, derivation and integral error. The result of the simulation shows that the proposed controller has high speed and lower overshoot value. Therefore, step response is better than PID controller.

References

[1] M. Benosman and G. Le Vey. 2004. Control Of Flexible Manipulators: A Survey. Robotica. 22(5): 533-545.

[2] S. Ozgoli and H. Taghirad.. 2006. A Survey On The Control Of Flexible Joint Robots. A*sian Journal of Control*. 8: 332-344.

[3] A. Fenili and J. M. Balthazar. 2011. The Rigid-Flexible Nonlinear Robotic Manipulator: Modeling And Control. *Communications in Nonlinear Science and Numerical Simulation*. 16: 2332-2341.

[4] Y. Li, S. Tong, and T. Li.. 2013. Adaptive Fuzzy Output Feedback Control For A Single-Link Flexible Robot Manipulator Driven DC Motor Via Backstepping. N*onlinear Analysis: Real World Applications.* 14: 483-494.

[5] S. Ulrich, J. Z. Sasiadek, and I. Barkana. 2012. Modeling And Direct Adaptive Control Of A Flexible-Joint Manipulator. *Journal of Guidance, Control, and Dynamics*. 35: 25-39.

[6] R. H. Cannon and E. Schmitz. 1984. Initial Experiments On The End-Point Control Of A Flexible One-Link Robot. *The International Journal of Robotics Research*. 3: 62-75.

[7] J.-H. Park and H. Asada.. 1992. Integrated structure/control design of a two-link nonrigid robot arm for high-speed positioning. Robotics and Automation Proceedings. 1992 IEEE International Conference on. 735-741.

[8] A. Green and J. Z. Sasiadek. 2004. Dynamics and trajectory tracking control of a two-link robot manipulator. Journal of Vibration and Control. 10: 1415-1440.

[9] X. Bo, K. Fujimoto, and Y. Hayakawa. 2005. An Energy-Based Nonlinear Control for a Two-Link Flexible Manipulator. Transactions-Japan Society For Aeronautical And Space Sciences. 48: 92-107.

[10] Y. Li, G. Liu, T. Hong, and K. Liu. 2005. Robust control of a two-link flexible manipulator with quasi-static deflection compensation using neural networks. *Journal of Intelligent and Robotic Systems*. 44: 263-276.

[11] A. Dehghani and H. Khodadadi.. 2015. Fuzzy Logic Self-Tuning PID Control for a Single-Link Flexible Joint Robot Manipulator in the presence of Uncertainty. 15th International Conference on Control, Automation and Systems (ICCAS) 2015 IEEE International Conference. 186-191.

[12] I. Akyuz, E. Yolacan, H. M. Ertunc, and Z. Bingul. 2011. PID And State Feedback Control Of A Single-Link Flexible Joint Robot Manipulator. 2011 IEEE International Conference on Mechatronics (ICM). 409-414.

[13] K. Groves and A. Serrani. 2004. Modeling and nonlinear control of a single-link flexible joint manipulator. APENDICES.

[14] M. F. Rahmat and M. M. Ghazaly. 2006. Performance Comparison Between PID and Fuzzy Logic Controller in Position Control System of DC Servomotor. Jurnal Teknologi. 45(D): 1-17.

[15] H. Wicaksono, Y. Gunawan, A. Yodinata, Leonardie. 2015. Performance Analysis Fuzzy-PID Versus Fuzzy for Quadcopter Altitude Lock System. *Jurnal Teknologi.* 77(22): 33-38.

[16] M. J. Nekooei, Jaswar and A. Priyanto. 2015. A Simple Fuzzy Logic Diagnosis System for Control of Internal Combustion Engines. *Jurnal Teknologi.* 74(5): 121-124.

[17] R. Adnan, M. Tajjudin, N. Ishak, H. Ismail, and M. H. F. Rahiman. 2011. Self-Tuning Fuzzy PID Controller For Electro-Hydraulic Cylinder. Signal Processing and its Applications (CSPA), IEEE 7th International Colloquium on. 395-398.

[18] R. Akbari-Hasanjani, S. Javadi, and R. Sabbaghi-Nadooshan. 2015. DC Motor Speed Control By Self-Tuning Fuzzy PID Algorithm. *Transactions of the Institute of Measurement and Control*. 37(2): 164-176.