ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 9, March 2015

A Controller Design for Servo Control System Using Different Techniques

Eyad Soliman, Magdy Saoudi, Hamid Metwally
eyad_soliman@yahoo.com
STCE (Science and Technology Center of Excellence),11785 Cairo, Egypt
Faculty of Engineering, Zagazig University, 44519 Zagazig, Egypt
Faculty of Engineering, Zagazig University, 44519 Zagazig, Egypt

Abstract— This paper presents and analyzes different methods of speed and position controller design for a servo mechanism system. First, the servo motors and rotating masses were modelled, and PID controller was developed to achieve the desired response. Then, under the absence of system model, both PID and Fuzzy controllers were designed to achieve the desired response. Ziegler-Nichols method was used to design the PID controller. A real-time computer was used to realize the controllers obtained from each approach using rapid prototyping technique. Finally the paper compares the results obtained from the different control techniques.

***Index Terms—* Automatic control, Automation, Servomechanism, Xpc Target Box, PID, Fuzzy, PMBLDC Motors, Matlab Simulink**

I. INTRODUCTION

A servomechanism, sometimes shortened to servo, is defined as feedback control system in which the controlled variable is physical position or motion[1]. Many servomechanisms are used to maintain an output position or motion in close correspondence with an input reference signal and hence they are follow up systems. Alternatively a servomechanism is an automatic device that uses error-sensing negative feedback to correct the performance of a mechanism and is defined by its function. It usually includes a built-in encoder. The term servomechanism is correctly applied only to systems where the feedback or error-correction signals control the mechanical position, speed or other parameters.

II. MECHANICAL SYSTEM DESCRIPTION

The mechanical system under study consists of two rotating masses and one conveyor belt linking these masses, Each rotating mass is drived by a permanent magnet brushless direct current (PMBLDC) servo motor which must follow the desired speed profile. Each servo motor is driven by an amplifier (driver) supplying the motor with pulse width modulated electric power. The conveyor belt gets its moving energy from both motors. The synchronization between the two rotating masses must be maintained to prevent the damage of the conveyor belt (intermediate conveyor) as shown in Fig. 1. Each servo motor requires a speed control loop to achieve the desired speed profile and a third position control loop is required to ensure the synchronization of the two motors and to prevent any damage to the conveyor belt.

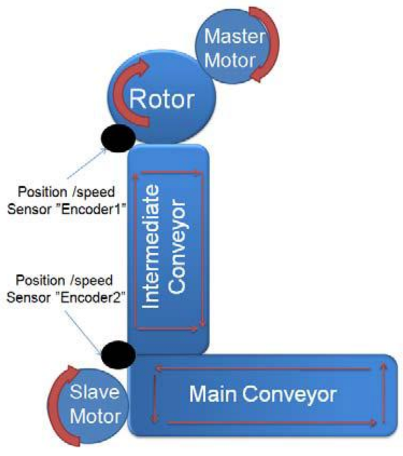
Master Motor
Rotor
Position /speed Sensor "Encoder1"
Intermediate Conveyor
Position /speed Sensor "Encoder2"
Slave Motor
Main Conveyor

Fig.1- real mechanical system.

III. EXPERIMENTAL SYSTEM

For the purpose of experimental work, an electromechanical system is built to act as the plant to be controlled as shown in Fig.2. This system consists of two PMBLDC motors, two motor drives, two incremental encoders, a real-time computer outfitted with input and output modules and a host computer with Matlab package installed on.

The controller model is developed on the Matlab Simulink tool and then compiled into C code and downloaded to the real time computer through an Ethernet cable. The real-time computer runs the controller program providing voltage control signal to the motor amplifier, which provides the PWM of the PMBLDC motors, each motor has an incremental encoder coupled to its shaft through a gear coupling. The input module in the real time computer receives incremental encoder output signal. The motors are outfitted with dummy inertial mechanical loads to test the various control techniques.

The MATLAB package is used as a readymade programming package that can be used to create controller algorithm on host PC that could be executed on a real-time

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 9, March 2015

computer and turn it into a rapid prototyping platform, (xPC target box) [2]. The MATLAB SIMULINK package contains tools to achieve and design both PID and fuzzy logic controllers[3]. The controller module designed on the SIMULINK environment can be compiled to C code and built on the real-time computer. The C code is transferred to the real-time computer through an Ethernet cable.

The real time computer receives the two encoder readings via it's incremental encoder reading DAQ card and produces two voltage control signals to compensate the motor produced torque through two analogue output channels of the I/O card. The voltage signals are supplied to the torque control input of the driver. The driver supplies PWM to the motors to produce proportional torque to the input torque control voltage. The two encoders coupled to the motors' shaft return feed back to the real-time computer's encoder reading channels. The position of the shaft is calculated from the reading of the incremental encoder counter and converted into radians which in turn are used to calculate the angular velocity of the motors to be compared with the command speed and produce error signals.

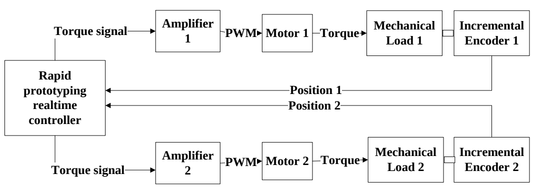

Fig.2 Block diagram of the experimental system

IV. Servo System Modeling and PID Controller Design

A-System Modeling

A.1.Amplifier Motor Modeling: In current (torque) operation mode, the amplifier output current I is directly proportional to the input voltage V, and the proportionality factor is K_a thus:

$$I = K_a V \qquad (1)$$

The developed torque

$$T_d = K_t I \qquad (2)$$

Where K_t is the torque constant. If J is the total moment of inertia of the load and motor, T_f is opposing friction, then:

$$J\alpha = T_d - T_f \qquad (3)$$

Where α is the angular acceleration

$$\alpha = \frac{d\omega}{dt}$$

where ω is the angular velocity. taking Laplace transform

$$\omega = \frac{1}{s}\alpha \qquad (4)$$

$\omega =$ where θ is the angular position of the motor also,

$$\theta = \frac{1}{s}\omega \qquad (5)$$

Combining equations 1 through 5 and neglecting friction yields:

$$\frac{V}{[illegible]} = \frac{K_t K_a}{[illegible]} \qquad (6)$$

A.2.-Position Sensor Modeling: The encoder uses two output channels (A and B) to sense position. Using a disk with two code tracks with sectors positioned 90 degrees out of phase. The encoder generates N pulses per revolution. Channels A & B produce 1024 pulses for each encoder rotation. As two signals are shifted by one quarter of a cycle, the controller can divide each encoder cycle into four quadrant counts resulting in an effective resolution of 4N counts per revolution or turn. Since each revolution is 2π radians, the resulting encoder gain is 1024 pulse per rev and the encoder has an equivalent gain of 636 counts/ radian.

A.3.-Observer Modeling: In control theory, a state observer (also referred to as an estimator) is a system that provides an estimation of the internal state of a given real system, from measurements of the input and output of the real system. It is typically computer-implemented, and provides the basis of many practical applications. In the system at hand the encoder measures the angular position of the motor shaft in order to measure the angular velocity a differentiator might be used where $\omega = d\theta$ but The use of differentiators in a control system leads to noise amplification, delay in the system, and introduction of a phase lag of 90°. It is well known that this phase lag reduces stability margins and could even destabilize the closed-loop system. Therefore, in general, the phase lag should be avoided as much as possible. In the comparative study of differentiators, made by Jing Liu[4] it was observed that the observer-based differentiators offer the best results plus it was the simplest in practical implementation. The block diagram of the state observer is shown in Fig.3.

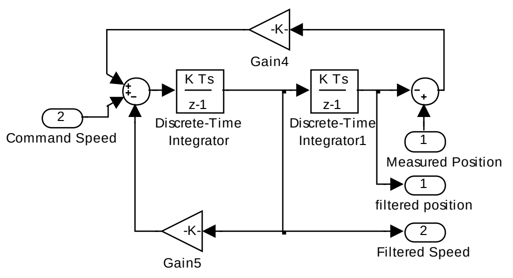

Fig.3- State observer (estimator) block diagram.

The state observer model can be deduced from Fig.3 as

$$\frac{[illegible]}{[illegible]} = \frac{[illegible]}{[illegible]^2 + 100[illegible] + 500} \qquad (7)$$

where [illegible] is the angular velocity and θ_m is the measured position.

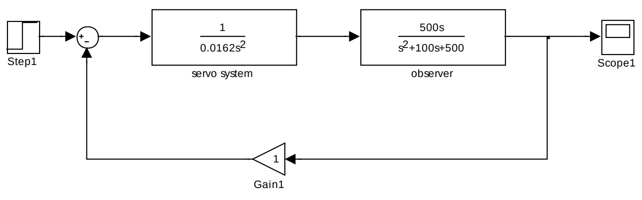

Fig.4- Servo system + observer model closed loop speed control

B- PID Speed Controller Design Using Matlab SISO Tool

PID controller can be designed using the MATLAB SISO Tool[5] based on the system transfer function obtained from Fig.4. PID controller consists of an integrator and a complex zero and can be represented as $k\frac{[illegible]}{[illegible]}$ and a step input . From the tool's GUI the CLOSED LOOP r to y option is chosen[5]. The analysis plot shows the transient and steady state response of the system. The desired requirements for steady state and transient responses are added and achieved by manually moving the roots or by using the automatic system optimization feature. The PID parameters are obtained from the SISO tool are listed in Table 1. The obtained PID parameters were used to control the speed of the two motors. Fig.5. shows the unity feedback time response of the two motors speed with no controller, Table 2 Shows the time response parameters of this case where T_r is the rise time, T_s the settling time and Os % is the percentage overshoot. Fig.6. shows the unity feedback time response of the two motors speed with the PID controller obtained from the SISO Tool. Table 3 shows the time response parameters with speed PID control. PID controller resulted in reduction in T_r, T_s and Os % for the first motor. For the second motor the steady state error was eliminated, even though T_r, T_s and Os % were slightly increased.

Table 1- Speed PID controller parameters obtained using SISO Tool.

Control type	P(kp)	I(Ki)	D(Kd)
Motor1 Speed PID	1.073	0.6438	0.11
Motor2 Speed PID	1	0.77	0.16

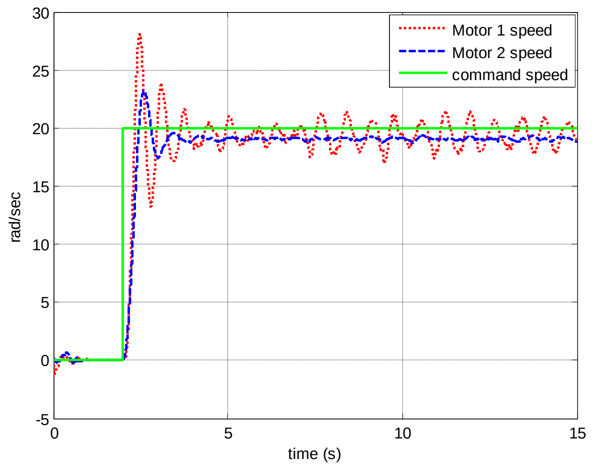

Fig.5. Unity feedback speed response of the system without controller

Table2- Shows the time response parameters of the unity feedback speed of the system without controller.

	Tr(sec)	Ts(sec)	Os %
Motor1	0.335	inf	40.5
Motor2	0.449	1.114	16.6

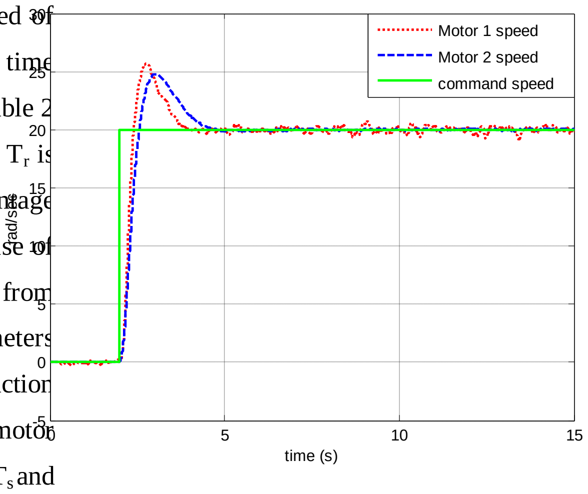

Fig. 6. Speed response of the real servo system with the PID

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 9, March 2015

speed controller obtained from the SISO tool.

Table3- Shows the time response parameters obtained from the system using the PID controller parameters obtained by the SISO.

	Tr(sec)	Ts(sec)	Os %
Motor1	0.398	1.617	28.6
Motor2	0.557	2.113	24

V. Practical Controller Design

For a system of unknown or a complicated mathematical model to be derived, real-time rapid prototyping hardware helps to simplify the process of controller design. If we consider the PID controller, there are many practical solutions to the problem of PID controller design for any unknown system. Ziegler-Nichols method can be used to design the PID controller for this type of systems[6]. But these methods would require a controller with the ability to change the PID parameters over a wide range to reach an optimum values for the PID controller. If we consider the traditional analogue controllers for this function, it would have serious limitations, real-time rapid prototyping hardware offer a unique solution for that problem as it offers the ability to change the controller variables easily and across a wide range of variation. Most practical solutions wouldn't offer the exact values of an optimum controller and the controller values would require further tuning to enhance its operation and to achieve any desired performance.

A- PID Controller Design Using Ziegler-Nichols Closed Loop Tuning Method

Ziegler and Nichols derived two empirical methods for obtaining PID controller parameters. These methods are closed-loop tuning method and open-loop tuning method.[7]

The closed-loop tuning method allows the use of the ultimate gain value K_u, and the ultimate period of oscillation P_u, to calculate the PID parameters as shown in Table 4. It is a simple method of PID tuning and can be refined to give better approximations of the controller. The closed-loop tuning method is limited to tuning processes that cannot run in an open-loop environment. This method is experimentally easy since only the proportionality factor K_p needs to be changed even though experimentation could be time consuming. It includes the dynamics of the whole process, which gives a more accurate picture of how the system is behaving, on the other hand it can venture into unstable regions while testing the K_p controller, which could cause the system to become out of control.

To find the parameters of the PID the following steps are performed:

1-The I-term and the D-term in the controller are turned off.

2-The P-term is turned to zero and then increased slowly until the output exhibits sustained oscillations.

3-At this "quasi steady-state" point we have the critical gain K_u and the oscillation time period P_u.

Table 4- Shows The parameters of the PID controllers obtained as suggested by Ziegler and Nichols.

Control Type	K_p	K_i	K_d
P	$0.50K_u$	-	-
PI	$0.45K_u$	$1.2K_p / P_u$	-
PID	$0.60K_u$	$2K_p / P_u$	$K_pP_u / 8$

The obtained parameters from the Ziegler-Nichols method may not achieve the optimum performance and further tuning of the system will be required to minimize the overshoot and reduce the rise and settling times. Fig. 7 shows the time response of the motor speed using the PID obtained from the Ziegler Nichols closed loop method listed in Table5 and open loop position control. Fig. 8 shows the position error between the two motors in this case.

Table 5- Speed PID controller parameters obtained using Ziegler-Nichols method.

Control type	P(kp)	I(Ki)	D(Kd)
Speed PID	0.22	0.12	0.015

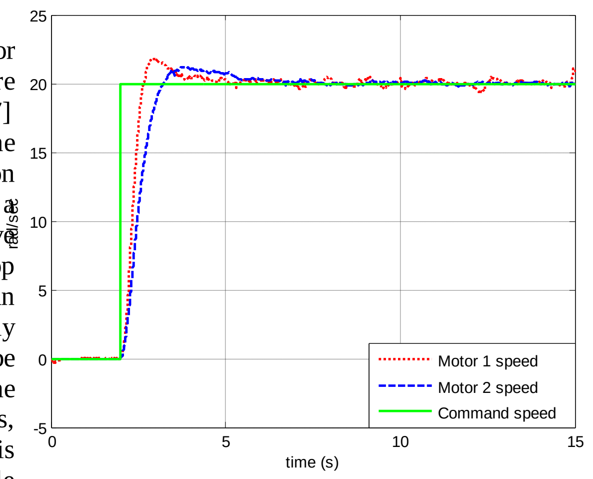

Fig. 7. Speed response with Ziegler Nichols PID speed controller and open loop position control.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 9, March 2015

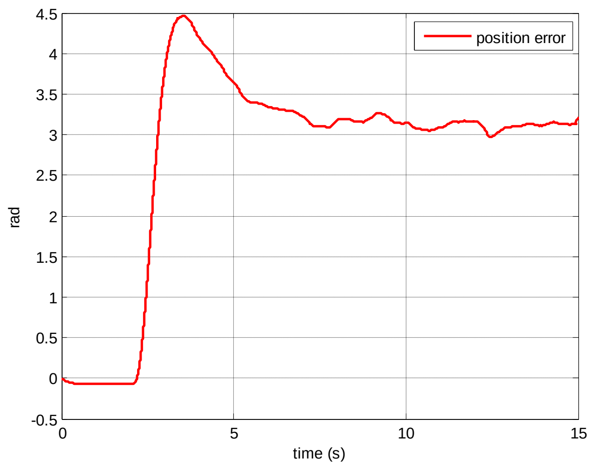

Fig. 8. Position error with Ziegler Nichols PID speed controllers and open loop position control

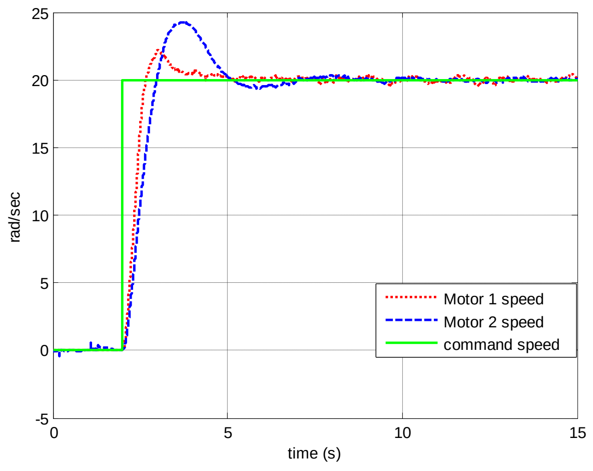

Fig. 10. Speed response with PID speed controllers and unity gain feedback position control loop.

A third position control loop was introduced to ensure the synchronization of the two motors and to prevent any damage to the conveyor belt as shown in Fig. 9. The time response of this case with unity feedback on the position loop is shown in Fig. 10 and Fig. 11.

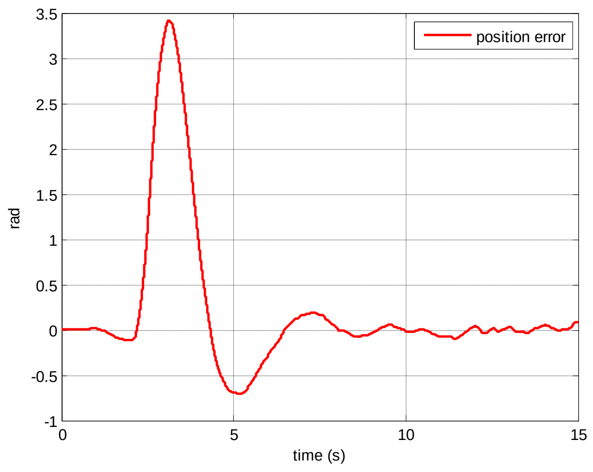

Fig. 11. Position error with PID speed controllers and unity gain feedback position control loop.

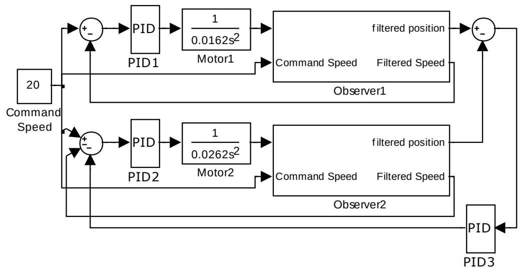

Fig. 9. Block diagram of the system showing the three control loops

The Ziegler-Nichols method was also used to design a controller for the 3rd control loop to minimize the position error between the two motors and prevent damage to the mechanical system. The position PID controller parameters obtained are shown in Table 6 and the time response is shown in Fig.12 and Fig.13. The time response of the PID controllers parameters obtained from Ziegler-Nichols method are listed in Table 6.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 9, March 2015

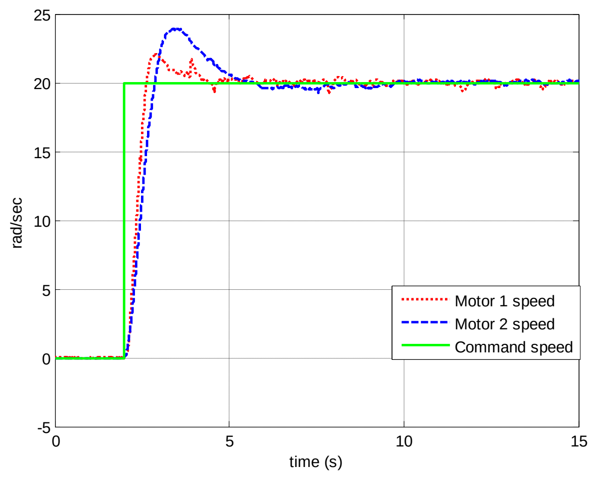

Fig. 12. Speed response with PID speed controllers and PID position controller.

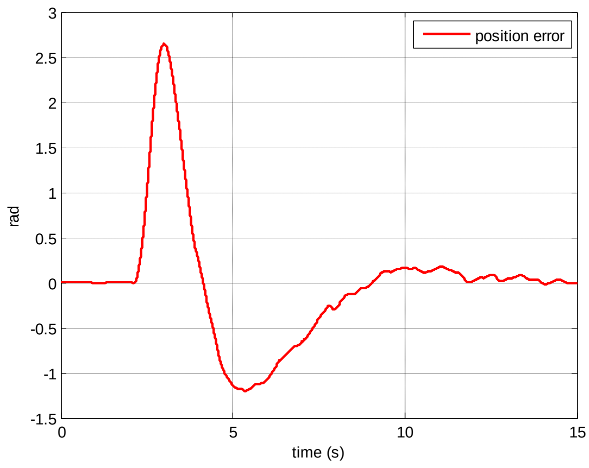

Fig. 13. Position error with PID speed controller and PID position controller

Table 6- position PID controller parameters obtained using Ziegler-Nichols method.

Control type	P(kp)	I(Ki)	D(Kd)
position PID	0.7	0.22	0.01

Table 7- Time response of PID controllers parameters obtained using Ziegler-Nichols method for motor 1 and motor 2.

	Control type	Tr(sec)	Ts(sec)	Os %
M1	PID speed control no position control	0.655	1.524	9.45
	PID speed and position control	0.666	1.321	9.25
M2	PID speed control no position control	1.216	2.136	6.3
	PID speed and position control	0.884	3.114	16

B- Fuzzy Logic Controller (FLC) design

Traditional control design methods use mathematical models of the system and its inputs to design controllers that analyze their effectiveness. Fuzzy control systems are rule-based systems in which a set of so-called fuzzy rules represent a control decision mechanism to adjust the effects of certain system stimulus[7]. The aim of fuzzy control systems is normally to replace a skilled human operator with a fuzzy rule-based system [8]. A fuzzy algorithm is an ordered sequence of instructions which may contain fuzzy assignment and conditional statements. The execution of such instructions is related to the compositional rules of inference [9].FLC uses fuzzy sets and fuzzy inference to derive control laws in which no precise model of the plants exists, and most of the priority information is available only in qualitative form. The basic idea of FLC is to make use of expert knowledge and experience to build a rule base with linguistic rules [10]. Proper control actions are then derived from the rule base. A fuzzy rule is a conditional statement, expressed in the form IF-THEN. The deduction of the rule is called the inference and requires the definition of a membership function characterizing this inference. This function allows the designer to determine the degree of truth of each position[10]. The FLC block in Matlab is shown in Fig.14 with the two inputs to the FLC block being the speed error and the change in speed error.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 9, March 2015

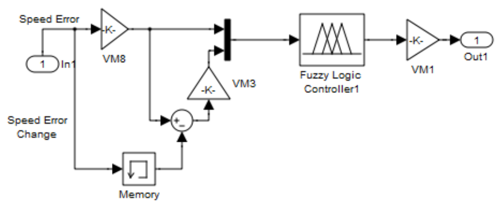

Fig. 14. Fuzzy controller block diagram in simulink.

The FLC technique was used to replace the PID speed controllers discussed earlier and the resulting time response is shown in Fig.15 and Fig.16. The FLC technique was also used to replace the PID controller in the position loop with the results shown in Fig.17 and Fig.18. Table 8 shows time response comparison between PID and FLC speed controllers for motor 1 and motor 2. Table 9 shows time response comparison between PID and FLC position controllers.

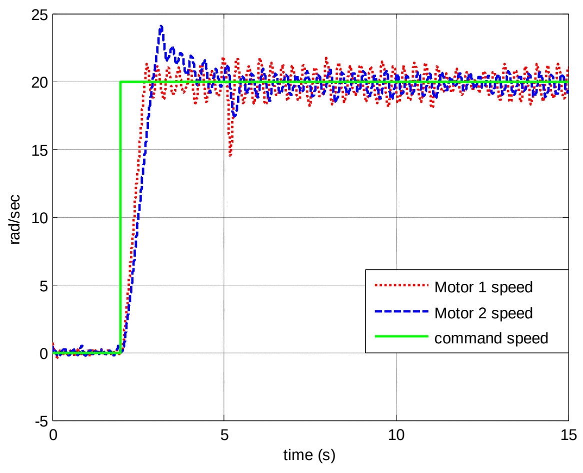

Fig. 15. Speed response with Fuzzy speed controller and PID position controller.

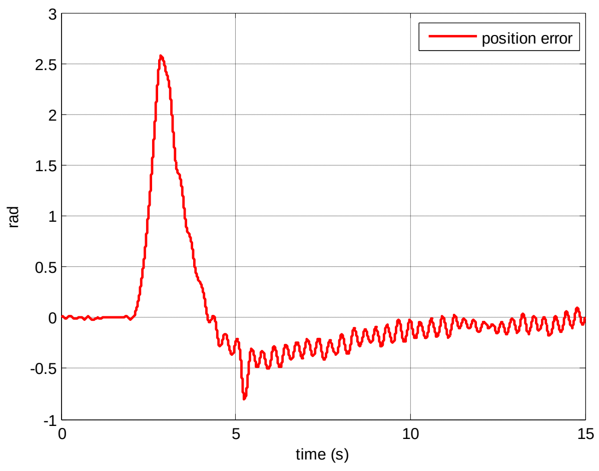

Fig. 16. Position error with Fuzzy speed controller and PID position controller.

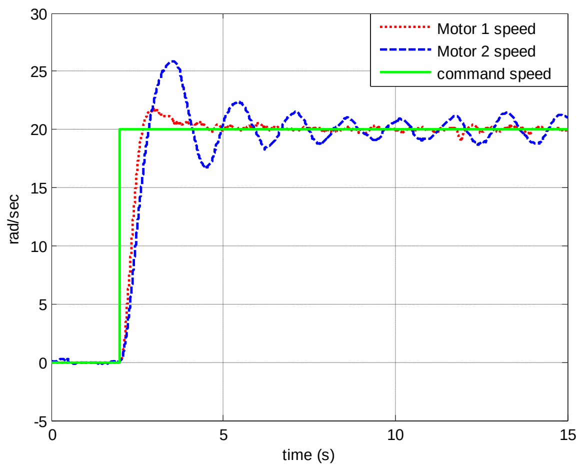

Fig. 17. Speed response with PID speed controller and Fuzzy position controller.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 9, March 2015

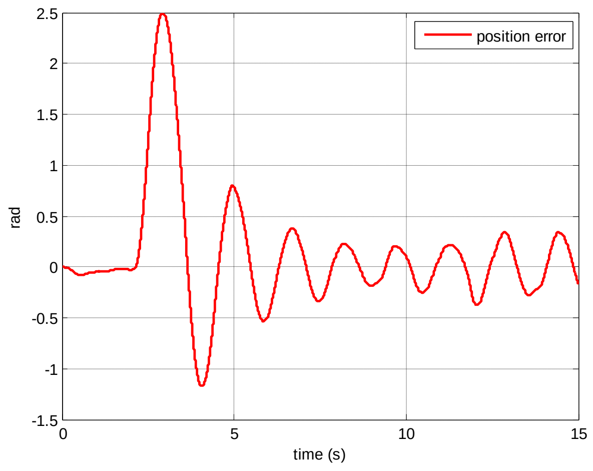

Fig. 18. Position error with PID speed controller and Fuzzy position controller.

Table 8- Performance comparison between PID and FLC speed control for motor 1 and motor 2.

	Control type	Tr(sec)	Ts(sec)	Os %
M1	PID	0.666	1.321	9.25
	FLC	0.69	inf	6.3
M2	PID	0.884	3.114	16
	FLC	0.925	3.429	20.3

Table 9- Performance comparison between PID and FLC position control .

	Tr(sec)	Ts(sec) at 0.5 rad	Os(rad)
PID	2.846	7.396	2.69
FLC	1.651	3.97	2.496

VI. Conclusions

In this work, three methods have been used to design a controller for a servo mechanism system. In the first method, the motors and rotating masses were modeled and PID controller was designed using MATLAB SISO Tool. The second method depends on Ziegler-Nichols closed-loop tuning method as an example of a heuristic method of tuning assuming an unknown model of the Electromechanical system. The third method depends on fuzzy logic control technique. The first method proved to be the most time consuming but resulting in a good time response characteristics. The second method is much less time consuming and gave results much similar to the first method. The third method was the least time consuming but the time response characteristics were not as good as the first .

In order to reduce the mechanical stress that is affecting the conveyor belt linking the two rotating masses, a position control loop was introduced to synchronize the position of the two motors, two methods were used to design the controller of that loop. The first method depends on Ziegler-Nichols closed-loop tuning. The second method depends on fuzzy logic control technique. The second method is less time consuming and slightly better time response characteristics with reduction in rise and settling time and slight reduction in percentage overshoot.

VII. Future Work

For future work, a high level controller will be designed to control the motion of the servo mechanism operating at various speeds and achieving different modes of operation, a human machine interface would be required to facilitate the control of the mechanism. The final control scheme will be programmed to embedded control circuit for mass production.

REFERENCES

[1] Robert N. Bateson, Introduction To Control System Technology 6th edition ,Prentice Hall, pp43-44.

[2] The MathWorks Inc., xPC Target For Use with Real-Time Workshop User's Guide(Version 3).

[3] The MathWorks Inc., Simulink Getting Started Guide.

[4] Jing Liu, "Comparative study of differentiation and integration techniques for feedback control systems" ,Cleveland State University, December, 2002

[5] The MathWorks Inc., Control System Toolbox User's Guide, Using the SISO Design Tool and the LTI Viewer.

[6] Ziegler, J.G and Nichols, N. B., Optimum settings for automatic controllers.

[7] Guesmi, K, Essounbouli, N., and Hamzaoui, A., “Systematic design approach of fuzzy PID stabilizer for DC–DC converters,” Elsevier, Energy Conversion and Management, No. 49, pp. 2880–2889, 2008.

[8] Taher, S. A., and Shemshadi, A., “Design of Robust Fuzzy Logic Power System Stabilizer,” World Academy of Science, Engineering and Technology, Vol. 27, pp. 8-14, 2007.

[9] Lin, B., “Power converter control based on neural and fuzzy methods,” Electric Power Systems Research, Vol. 35, pp. 193-206, 1995.

[10] Eker, I., and Torun, Y., “Fuzzy logic control to be conventional method,” Energy Conversion and Management, Vol. 47(4), pp.377-394, 2006.