PI-like Fuzzy Logic Position Controller Design for Electro-hydraulic Servo-actuator Based on Particle Swarm Optimization and Artificial Bee Colony Algorithms

Zeyad A. Karam
Department of Information Technology
Lebanese French University, Erbil, Kurdistan Region - Iraq
Email: zyad_karam88@yahoo.com

Abstract:

Electrohydraulic servo actuator mathematical model is one of the highly nonlinear hydraulic models. Electrohydraulic servo actuators used by aerospace, industrial and robotics applications for which accurate and fast performance are required in the presence of large loads or external disturbances. A servo actuator is made of a hydraulic cylinder or rotary actuator that is closely connected to an electrohydraulic servo actuator.

This work focuses on modeling and simulation of electrohydraulic servo actuator that used for position control of the flight surface in a military aircraft with presence of external forces. PI-like Fuzzy Logic position intelligent Controller (FLC) is designed and simulated to control the actuator desired position during a specified time with minimum steady state error, settling time and oscillations in position response. This controller is implemented by using MATLAB Simulink and it has a settling time of 0.168 (sec). By comparison with reference [3], which has settling time of 0.341 (sec), there is enhancement by using the proposed controller in settling time about 50.733%. In addition, there is a small fluctuation around the desired position because the controller of mentioned reference does not compensate the force effect and nonlinearities in the actuator model. Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) algorithms are used for tuning the gains of the PI-like Fuzzy Logic position Controller to satisfy the minimization in position error at the presence of the external force. The results show that the performance of ABC is more efficient than of PSO algorithm, because the trials by PSO have minimal fitness of 0.0008, but by ABC the minimal fitness achieved is 0.00072.

Key word: Electrohydraulic Servo Actuator, PI Controller, FLC, PSO and ABC Algorithms.

1. Introduction:

The closed-loop hydraulic actuator appeared in industry applications when high performance was required. But these hydraulic servo actuators are still heavily used in high-performance industrial applications such as the automation of the production line. It begins taking a wide usage in variety of industries. Examples for these applications are material handling, plants, robotics, mining, automotive testing and oil exploration [1].

The servo or a servomechanism is a control system that directs the output in quick and accurate way to follow the command signal by measuring its own output. The significance in the technology of servo driver is increasingly becoming the major use in the machine automation, where the operators are demanding greater performance, faster response and simpler adjustment in operation [1]. However, the position response to the electrohydraulic servo actuator is one of the most important characteristic for actuator researches. Several works have been done in this regard.

Nitesh. In 2013 [2] designed a nonlinear dynamic model of a two spool electro hydraulic servo-actuator. The mathematical model of the two-stage and two spools split type with load was developed. The model incorporates the interconnection between the torque motor stage, spool position dynamics, chamber pressure dynamics, output flow and load dynamics relationship. The main drawback in the work was in the position response, where these responses were tested under no disturbances or external forces that appeared in the industrial application with the Electro-hydraulic servo-actuator and with inaccurate performance of the position controller.

Haitao. in 2013 [3] described the triplex redundant hybrid actuation system with the operation principle of electro hydraulic actuator. The simulation model was presented with the triplex redundant. In addition, the physical principle of the actuator work was tested with different loads and the results were investigated with position responses. The main drawback in the work is the inaccurate responses for the desired position by the PID controller and its need for improvement using more efficient controller for minimizing the error in position response with faster performance.

Edvard. in 2011 [4] proposed a self-organizing and self-learning fuzzy algorithm for position control of hydraulic servo drive. The algorithm has the ability of self-organization and self-learning. It is able to make the adaptation to the parameter change and to deal with the nonlinear dynamics in drive motion. The main drawback in the work is that there is no algorithm proposed to provide the convergence of a learning mechanism and the position error. Nowadays there are many algorithms like Genetic, particle swarm optimization, Ant, Bee colony are used as optimization algorithms for improving the controller's performance by finding the optimized controller's parameters.

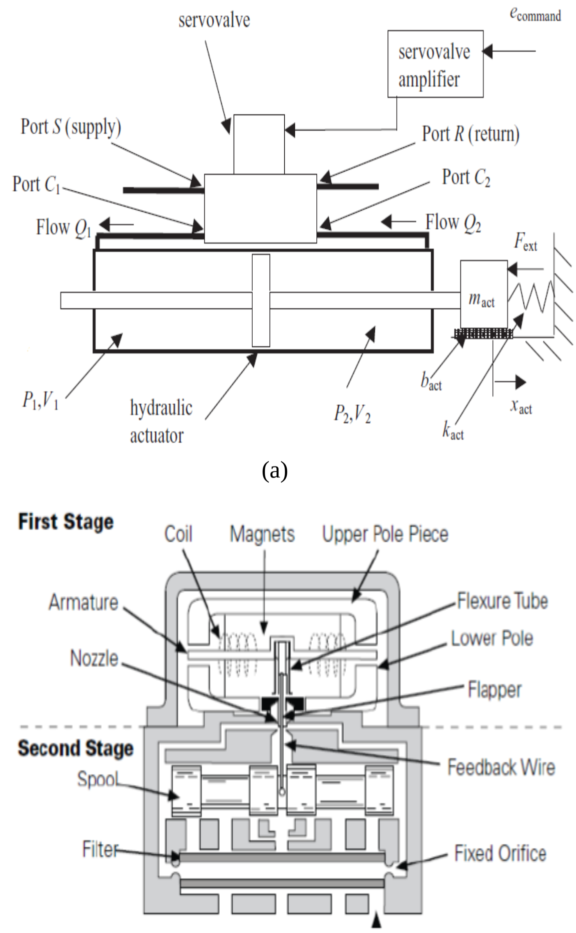
In this work, an intelligent controller for electrohydraulic servo-actuator is designed to precisely control the position of a flight control surface in a military aircraft. The nonlinear mathematical model for the actuator is modeled and simulated by Matlab. Also, the effect of external force on the actuator when touched to the load (air) is considered in the actuator model. The controller is PI-like Fuzzy Logic position Controller (FLC) used to deal with the model nonlinearities, external force and position error in actuator response.

A Particle Swarm Optimization Algorithm (PSO) is used to tune the PI-like Fuzzy logic position controller gains to get the desired actuator position with minimized steady state error, settling time and oscillation in the simulated model response. This position response was also improved using Artificial Bee Colony (ABC) for tuning the PI-like FLC gains. These algorithms were used to make the position controller get the desired position in improved way.

Not all the previous works considered control system that can deal with model nonlinearities and they not have an efficient way for adapting the controller to finding the minimum steady state error, settling time and oscillations in position response. These are the main points that should be solved in this work.

2. Mathematical Model of Electrohydraulic Servo-actuator:

The basic component in Electrohydraulic Servo-actuator is servo valve. Servo valve has special design for the flow control valve that have ability to direct the flow between zero and some desired value. The hydraulic valve can controlled the flow direction by their design through moving the internal component (valve spool), between two or three desired positions. Figure (1) shows the schematic for servo actuator. Figure (2) shows the frame of the valve spool and ports of a typical four-way hydraulic valve. The valve spool has free motion in a cylindrical core that has ports cut into it. As the spool moves to the right, it discloses the control ports (C_1 and C_2) to the supply pressure and return ports, respectively. The motion of the spool to the left makes the connections reversed and the flow of the fluid been from supply to C_2 and from C_1 to return. A servo valve has a complex design that makes the spool quickly and accurately positioned at any desired point between fully closed and fully open. The form of the model equations represents the mathematic model of the valve (velocity and position).



(a)

(b)

Figure1: (a) Electrohydraulic servo actuator scheme, (b) Design of the two-stage servo actuator.

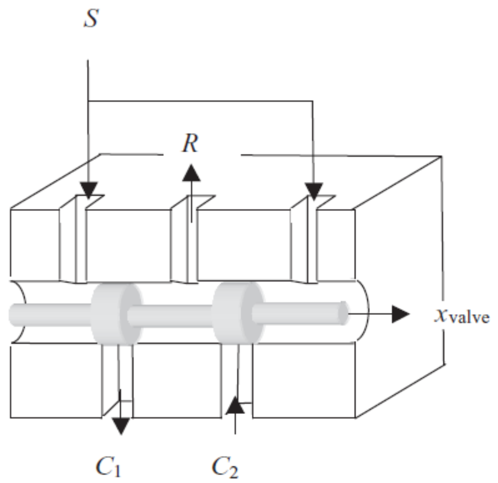


Figure2:.The valve spool and ports.

$$\dot{x}_{valve} = v_{valve} \quad .... (1)$$

$$\dot{v}_{valve} = a_1 v_{valve} + a_0 x_{valve} + b_0 e_{command} \quad ..... (2)$$

Where a_1, a_0 and b_0 are the model coefficients, x_{valve} by (m), v_{valve} by (m/s) and $e_{command}$ by (volt). Equations (1) and (2) make up the first two state equations for the servo-actuator model.

Figure (1.a) shows that in the mechanical parts of the system there are two energy-storing elements. One is the combined mass of the piston, piston rod and the m_{act}, and the other is the spring k_{act} . The state variables for these two elements are the position and the velocity of the actuator:

$$\dot{x}_{act} = v_{act} \quad \dots (3)$$

$$v_i(k+1) = \chi(v_i(k) + c_1 r_1 (x_p best_i(k) - x_i(k)) + c_2 r_2 (x_p best - x_i(k))) \quad \dots (4)$$

Where k_{act}, b_{act} and m_{act} represent the coefficients associated with the motion of the actuator and its load. x_{act} by (m), v_{act} by (m/s), F_{ext} by (N), P_1 and P_2 by (MPa). Any externally applied load force for the flight surface (considered as model input for this model) can represented by F_{ext}. In the hydraulic part, the force of motion for the actuator system comes from cylinder pressures that make the flow entering and leaving the actuator by ways of the valve. The others state variables representing the potential energy that stored in the two hydraulic capacitors are the entrapped volumes between the actuator piston and the valve that can be modeled as fluid capacitances with the pressures of $P1$ and $P2$:

$$\dot{P}_1 = \frac{1}{C_{f1}}(Q_1 - A_s v_{act}) \quad \dots (5)$$

$$\dot{P}_2 = \frac{1}{C_{f2}}(A_s v_{act} - Q_2) \quad \dots (6)$$

$$C_{f1} = \frac{V_1(x_{act})}{\beta} \quad \dots (7)$$

$$\boldsymbol{C_{f2}} = \frac{\boldsymbol{V_2(x_{act})}}{\boldsymbol{\beta}} \quad \dots (8)$$

Where β is fluid physical property that quantifies the relation between pressures change and volume change. V_1 and V_2 by (m^3) are the entrapped volumes of the chambers. C_{f1} and C_{f2} by (m^3/(N/m^2)) dependents on the position of the actuator piston. Q_1 (m^3/s) is the flow out of valve port C_1 and into the left-hand side of the cylinder, Q_2 (m^3/s) is the flow into the valve at port C_2 coming from the right-hand side of the cylinder. The flows through the valves (Q_1 and Q_2) are described as functions of the state variables. However, the spool valve for this actuator modeled as sharp-edged orifices with turbulent flow is proportional to the square root of the pressure differential through the valve. For the positive valve spool (to the right in Figure 2), then Q_1 can modeled as:

$$Q_1 = C_d w_{valve}\, x_{valve} \sqrt{\frac{2}{\rho}\,(P_s - P_1)} \quad \dots (9)$$

The last two considerations to complete the actuator model are: First, equation (9) is incorrect if the pressure cylinder P_1 is higher than the supply pressure P_s as a consequence the modification for the equations should happen to covering all pressure possibility, second, if there is negative valve spool position, that means the Q_1 and Q_2 represent the flow directions to and from different valve ports. The following conditional equations formulate these cases [5]:

If $x_{valve} > 0$ (Right)

$$Q_1 = C_d w_{valve} x_{valve}\, sgn(P_s - P_1) \sqrt{\frac{2}{\rho}|P_s - P_1|} \quad \dots (10)$$

$$Q_2 = C_d w_{valve}\, x_{valve} \sqrt{\frac{2}{\rho} P_2} \quad \dots (11)$$

If $x_{valve} < 0$ (Left)

$$Q_1 = C_d w_{valve}\, x_{valve} \sqrt{\frac{2}{\rho} P_1} \quad \dots (12)$$

$$Q_2 = C_d w_{valve} x_{valve}\, sgn(P_s - P_2) \sqrt{\frac{2}{\rho}|P_s - P_2|} \quad \dots (13)$$

3. Simulink Model of Electrohydraulic Servo-actuator :

In this work, the Electrohydraulic servo-actuator is used to control the position of the flight surface in a military aircraft. The effective mass of the control surface and the actuation mechanism is 40 (kg) (including the moving part of the actuator), and the estimated overall linear damping is 6000 (N.s/m). The mechanism effective stiffness is about 800 (N/m), [5].

The bore of the hydraulic cylinder is about 0.1 (m), and the rod diameter is 0.02 (m), and a total stroke of 0.7(m). The valve is a high performance servo valve rated at 5 (kg/min), has the ability to respond for sinusoidal command with limit of 100 (Hz) without significant degradation. The other parameters of the system used in the Simulink model are listed in Table (1). Figure (3) shows the Simulink model of Electrohydraulic Servo-actuator by Matlab.

Table 1: Electrohydraulic Servo-actuator Parameters.

Servo-valve and fluid parameters			Load/actuator parameter		
Parameter	Value	Units	Parameter	Value	Units
$\boldsymbol{w_{valve}}$	0.0025	M	m act	40	Kg
$\boldsymbol{b_0}$	90	m/V s^2	b_{act}	6000	N s/m
$\boldsymbol{a_0}$	360000	$1/s^2$	k_{act}	800	N/m
$\boldsymbol{a_1}$	1000	1/s	Stroke	0.7	m
$\boldsymbol{\beta_{fluid}}$	689	MPa	A_s	0.0075	m^2
$\boldsymbol{\rho_{fluid}}$	900	Kg/m^3	P_s	20.7	MPa
$\boldsymbol{C_d}$	0.61	Dimensionless			

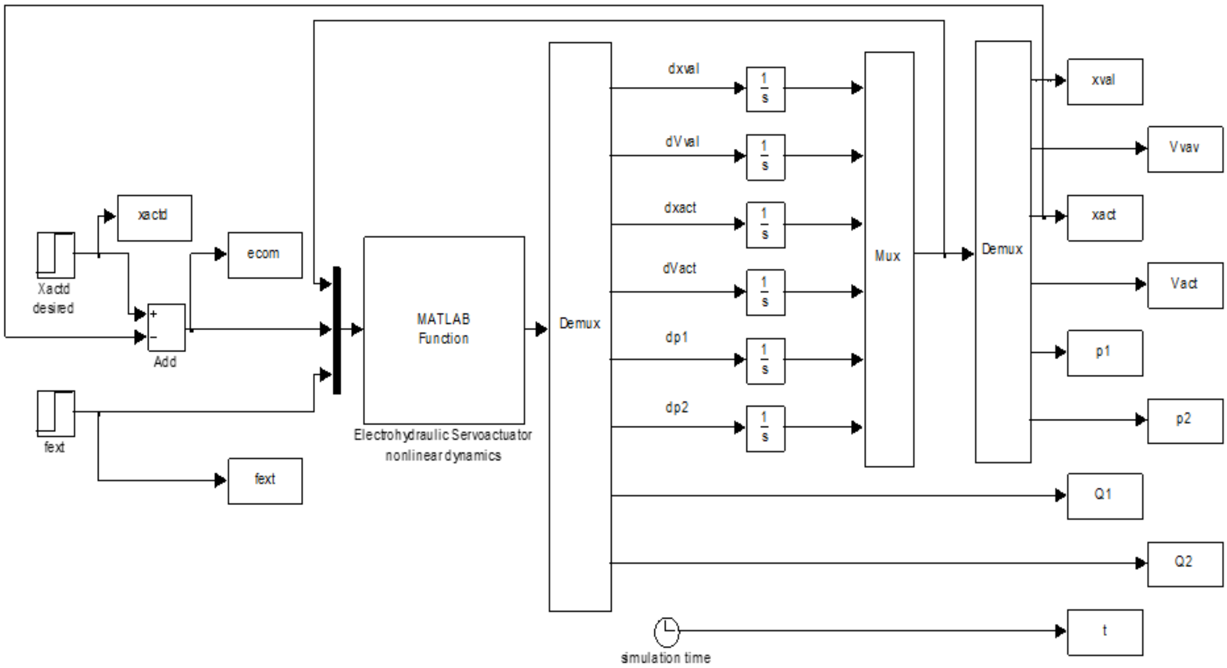


Figure (3).Electrohydraulic Servo-actuator Simulink model.

4. Design and Simulation of PI-like Fuzzy Logic Position Controlled Electrohydraulic Servo-actuator:

The actuator used for controlling the fight surface with desired direction by changing the displacement of the valve piston to a desired distance. In this work the desired distance is 0.01 (m) and applied as a step input to the system at presence of external forces of $F_{\text{ext}} = 100\sin(2\pi f)$, f=30 (Hz) as normal frequency. For a nonlinear system like electrohydraulic servo-actuator, the desired input can matched only through precise controller. Figure (4) shows the block diagram of the system incorporating system model with controller part.

The block diagram of figure 4 incorporates the PI-like FLC as a position controller for the desired input, before starting with intelligent controller the PID conventional controller with auto tune mechanism by Matlab Simulink is used for controlling the system and the best result of position response after many trials is shown in Figure(5).

Figure (5) shows the actual position response for the piston position but it has coupled with overshoot and steady state error of small percentage due to the high nonlinearity in the actuator model with the presence of external force. These problems in the actuator position response are undesirable in flight control system, so there is a need for another controller with high performance in position response. PI-like FLC is proposed for dealing with control problems by controlling the piston position of the actuator and built by Matlab Simulink. The PI-like FLC controller is used to deal efficiently with the nonlinearities in the actuator model and tracking the desired position with eliminated overshoot, minimized steady state error, minimized settling time and minimized oscillations. The controller equation is [6]:

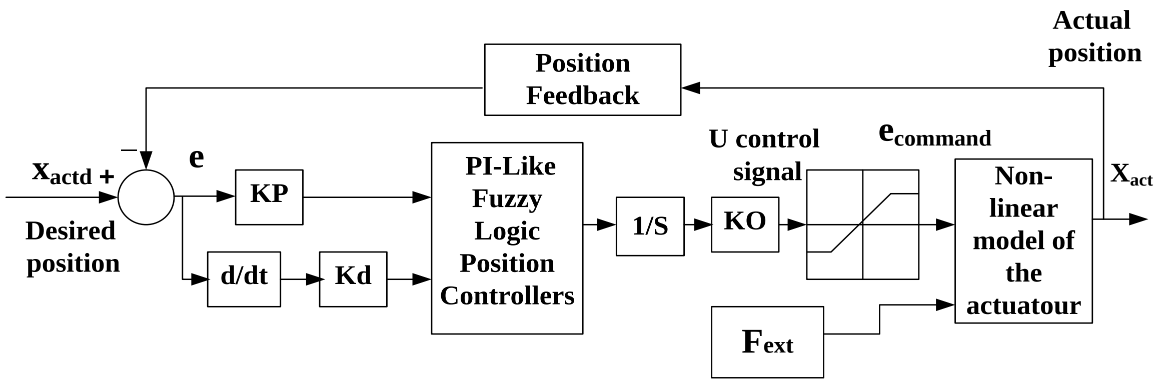


Figure 4: block diagram of controlled actuator by PI-FLC.

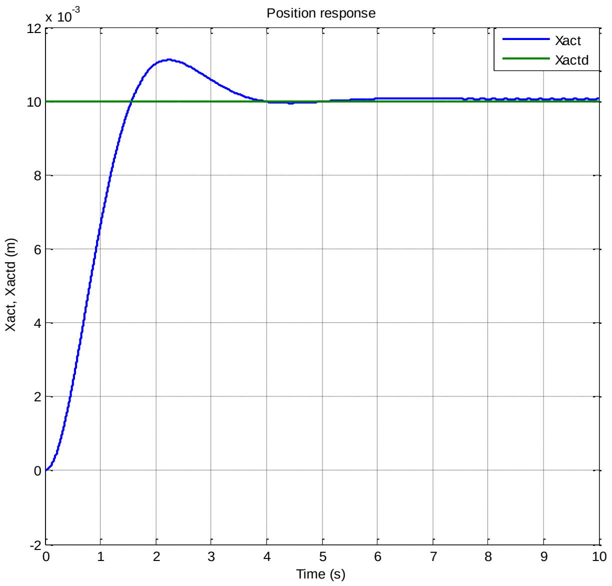


Figure5: Spool position response using conventional PID controller.

$$U(t) = K_p\, e(t) + K_d\, \dot{e}(t) \qquad \ldots.\ (14)$$

The inputs to the FLC are e (t) which is the position error signal and ė (t) which is the change of position error. The K_p and K_d are the error and change in error gains and the output from the controller integrated by integrator to get the PI control signal and K_O is the output gain used to adjust the outputs membership range. The FLC used in this work is Mamdani type. The inputs and output membership functions are seven triangular shaped. The defuzzification mechanism is selected to be Centre of gravity method [6]. Figure (6) shows the membership functions of the inputs and output. Table (2) lists the rules of the PI-like FLC Position controller that was chosen by trial and error.

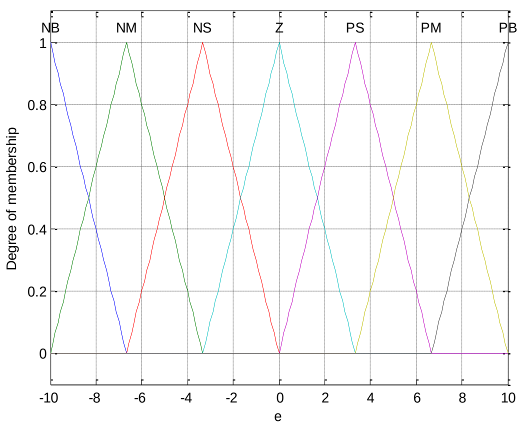


Figure 6: I/O Membership functions of FLC.
Table (2). FLC rules table.

ė / e	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	Z
NM	NB	NB	NB	NM	NS	Z	PS
NS	NB	NB	NM	NS	Z	PS	PM
Z	NB	NM	NS	Z	PS	PM	PB
PS	NM	NS	Z	PS	PM	PB	PB
PM	NS	Z	PS	PM	PB	PB	PB
PB	Z	PS	PM	PB	PB	PB	PB

The linguistic variables of FLC are NB (Negative Big), NM (Negative Medium), NS (Negative Small), Z (Zero), PS (Positive Small), PM (Positive Medium), and PB (Positive Big) with universe of discourse of [-10, 10], that represents the actual input model limits. The Simulink model of the PI-like FLC connected with Simulink actuator model is shown in Figure (7).

Many trials were done for tuning the PI-like FLC parameter gains to enhance the performance of the desired position response, the parameters gains at first found manually by trial and error were K_P =30, K_I =330 and K_O =135. Figure (8) shows this case of piston position response with PI-FLC of manual tuning. (In fuzzy system there is no PI controller, because of the fuzzy system built by depending on error and derivative of error, so the integrator was added to the output signal of PD controller to been PI controller, that means the gain K_P of PI controller is K_d in the PD controller but after integration becomes K_p, the gain K_I of PI controller is K_p at the PD controller but becomes K_I after adding the integration to the PD controller).

From figure 8, there is 0.000168 (m) as overshoot in position response coupled with small percentage of steady state error that still appeared in the majority of trial and error tuning results.

Nowadays many new optimization algorithms are used for finding the optimum controller gains by using a specified cost function relates to the system, which minimize the position error and overshoot or other features. In this work, the Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) algorithms are selected to tune the gains of the PI-like Fuzzy logic position controller to minimizing the error, minimizing the settling time and minimizing the oscillation in position response.

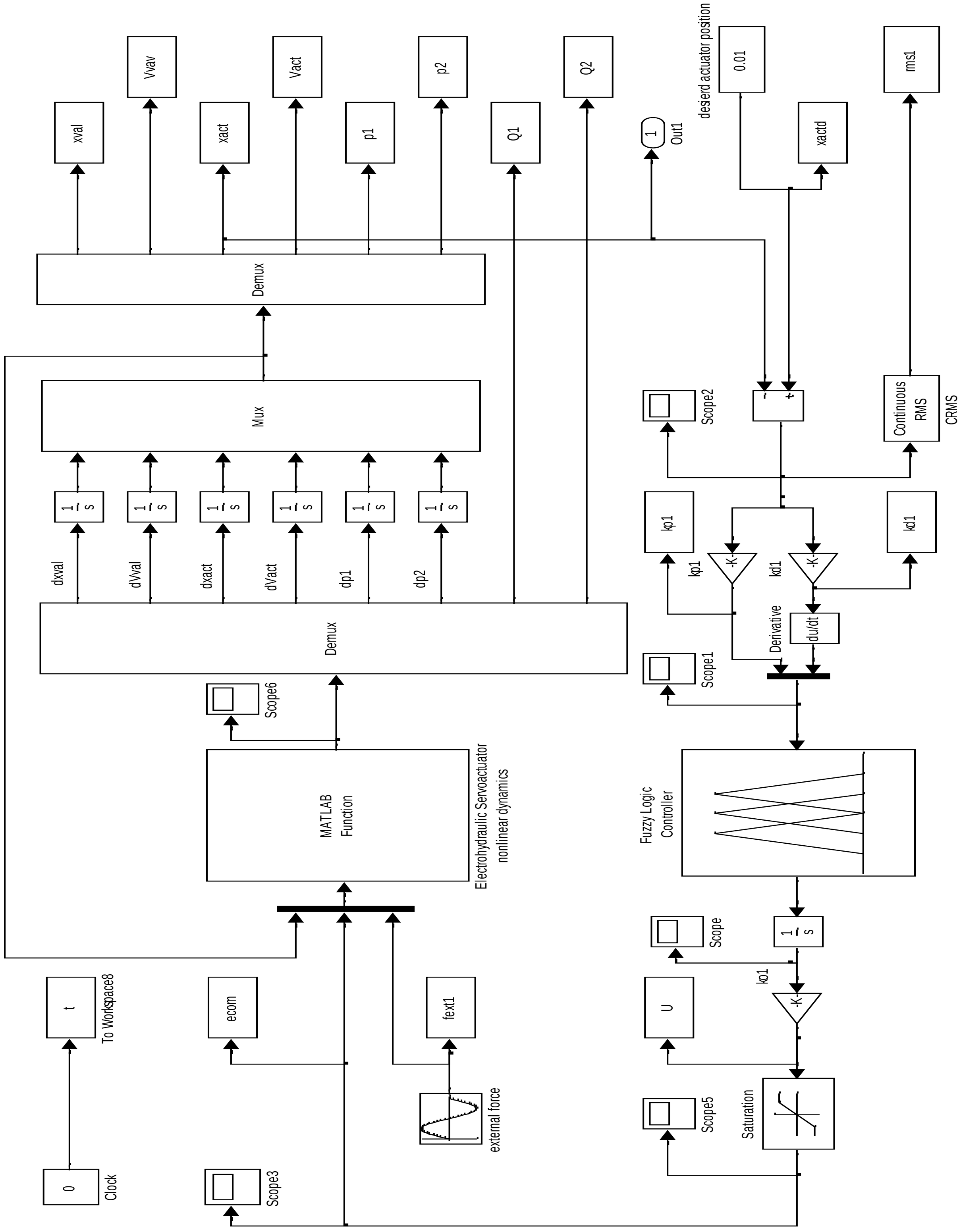


Figure 7: Simulink model of PI-like FLC with actuator model blocks.

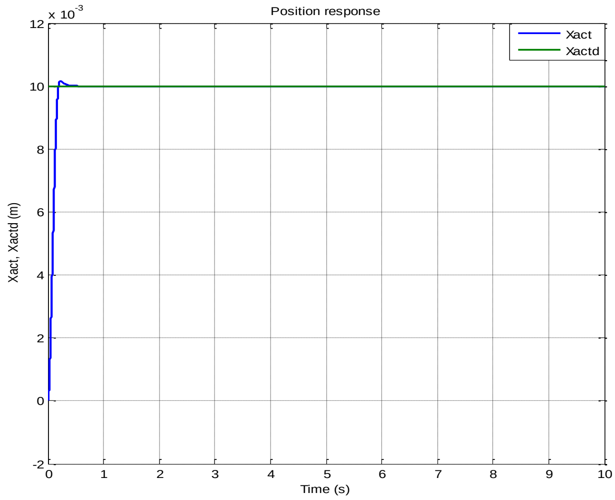


Figure 8: Spool position response using conventional PI-like FLC controller of manual tuning.

5. PSO as an off-Line Position Controller Gains Tuning Algorithm:

In Particle swarm optimization (PSO) algorithm, each population member is called a particle, and each particle flies around within multidimensional search space with specified velocity, which is updated constantly depending on the particle's own experience and the experience of the particle's neighbors or the experience of the whole group (swarm). It has already applied in many areas, such as function optimization, artificial neural network training, pattern classification and fuzzy system control [7]. The PSO is a population based stochastic optimization algorithm, firstly introduced by Kennedy and Eberhart in 1995 [8]. The particle swarm optimization (PSO) algorithm is performed as follows:

1) The particles here are the size of the swarm that makes the population size (*n*).

2) Random initialization for the particles start, then these particles move in the searching space for minimizing the goal objective function.

3) The parameters estimation (D dimension of the problem-controller gains= 3 in this work) done by objective function minimization.

4) The particle fitness is evaluated by depending on the objective function to make update for the x_p best of particle and the x_g best among the positions of all particles as goals in each step of evaluation.

5) The particle is directed to its x_p best and the x_g best among the particles. However, the particles amid to flies for the better searching areas over the searching space. The velocity of *i*-th particle v_i calculated by [9]:

$$v_i(k+1) = \chi(v_i(k) + c_1 r_1 (x_p best_i(k) - x_i(k)) + c_2 r_2 (x_p best - x_i(k))) \quad .... (15)$$

Where for the *i*-th particle in the *k*-th iteration, (c_1) and (c_2) are the acceleration coefficients named by the cognitive and social scaling parameters, (r_1) and (r_2) are two random numbers in the range of [0 1] and χ given by [9]:

$$\chi = \frac{2}{|4-\phi-\sqrt{\phi^2-4\phi}|} \quad .... (16)$$

Where ($\phi = c_1 + c_1$, $\phi > 4$) constriction coefficient used for controlling the convergence of the particle. That prevents explosion and directs particles to convergence. A new position of the *i-th* particle is evaluated by:

$$x_i(k+1) = x_i(k) + v_i(k+1) \quad .... (17)$$

The particle swarm is used as an optimization algorithm (off-line) to find the best gains of Fuzzy PI-like position controller for obtaining the accurate position responses by minimizing the error in the position, minimizing settling time and minimizing oscillation in the position response. This minimization gives closer matching between the desired and the actual position. The PSO runs off-line, thus computing time is not as important as in real time control. All elements of PSO are changed for many tests until the largest enhancement at minimum fitness function is reach of. Figure (9) shows the block diagram for PSO-based controller.

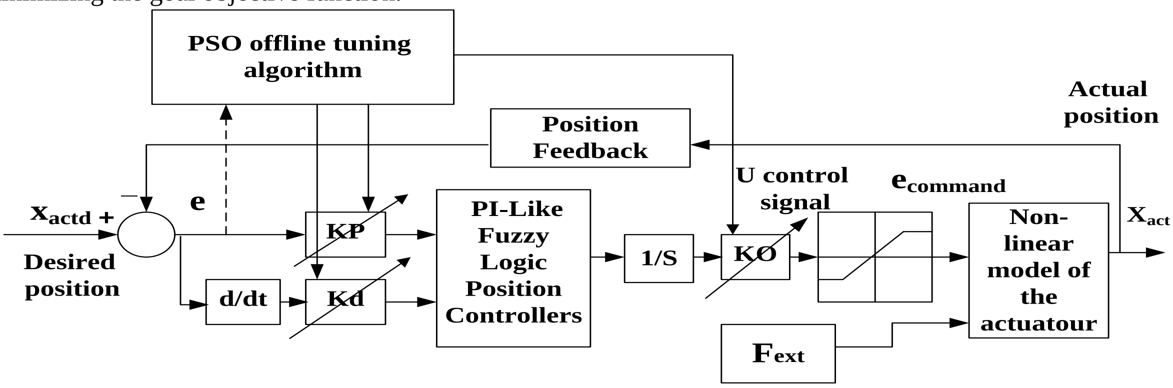


Figure 9: PSO-based PI-like FL Position Controller block diagram.

The fitness function that is used to evaluate the fitness of each particle is the RMSE and the overshoot in position responses is included in the fitness function. The RMSE included is:

$$RMSE = E(k) = \frac{1}{N}\sum_{i=1}^{N}\sqrt{e_1^2(i)} \qquad .... (18)$$

Where $e_1(i)$ is the position error of the i-th sample, N is the number of samples, and k is the iteration number.

6. ABC as an off-Line Position Controller Gains Tuning Algorithm:

Karaboga et al. in 2007 [10] proposed the Artifice bee colony. ABC algorithm is one of the optimization tools that depend on the intelligent behavior of the honeybee swarm.

At first, this ABC algorithm is proposed for unconstrained optimization problems that gives high performance on that kind of problems. For this work, the ABC algorithm is used for solving constrained optimization problem by tuning the parameters of PI-like Fuzzy Logic position controller to minimize the position error, minimize the settling time and minimize the oscillation [11]

The basic ABC algorithm consists of three types of bees: employed bees, onlooker and scout bees. The first half from the swarm of colony consist of employed bees and the second half includes the onlooker bees. In this algorithm, it is assumed that for each food source there is only one employed bee. That means, around the hive, the number of employed bees in the colony is equal to the number of food sources. Employed bees go to the location of food source and come back to the hive, and dance on this area to shearing information with onlooker bees. Finding of a new food source depends on the employed bee whose food source has been abandoned, so this bee become a scout one and starts searching for new food source. The scouts bee are randomly search the surrounding area for finding the new food source depending on an internal motivation of her memory or the memory of other bees or randomly. The onlooker bee waits in the hive and decide which food source to utilize depending on the information shared with the employed bees previously. The food source position that represents the possible solution for the optimization problem and the amount of nectar in the food source corresponds to the fitness of the related solution. The number of the solutions in population is equal to the number of employed bees or the onlooker bees [12].

The Artificial bee colony (ABC) algorithm performs as follows [13]:

1) Randomly initialize the food sources that are produced for all employed bees.
2) Repeat search.
3) Each one of employed bee goes to the one of initial food source in her memory and determines the neighbor source also, then evaluates its nectar amount (fitness) and dances in the hive area to shear nectar source information with onlooker bees.
4) Now, for the onlooker bees each one of them watches the dance of employed bees and after that chooses one of their sources depending on the dances (information shared about nectar) , and then goes to that source. After choosing a neighbor, food source around that one, she evaluates its nectar amount (fitness).
5) Consumed food sources (limited nectar) are determined and replaced with the new food sources discovered by scout's bee.
6) The best food source found so far is registered.

UNTIL (the required fitness satisfied).

The mathematical representation for this algorithm is as the follows [10]:

- Generate the initial population x_i (i= 1,2…. SN) is a D-dimensional vector. Where SN is the number of food sources (population size), and D is the number of the problem optimization parameters. D represents the PI-like FLC parameters to be optimized, where D= [K_P K_I K_O].
- After initialization in each iteration of all given cycles of solutions, every employed bee finds a neighborhood food source of its current food source and evaluates its nectar amount (fitness).
- The onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with the probability related to its nectar amount (fitness), the P_i that is associated with food source for sharing information calculated by the following equation:

$$P_i = \frac{a * fit_i}{(\sum_{n=1}^{SN} fit_n)+b} \qquad .... (19)$$

Where fit_i evaluated by its employed bee, which is proportional to the nectar amount of the food source in the position (i). (a and b) are the coefficients of probability function.

- For producing new elected food position by depending on the old one, the ABC uses the following equation:

$$v_{ij} = x_{ij} + \Phi_{ij}(x_{ij} - x_{kj}) \qquad .... (20)$$

Where $k \in$ {1, 2, . . , Number of food source} and $j \in$ {1, 2, . . ., D} are randomly chosen indexes.Φ_{ij} is a distributed random number between [−1, 1].

- The food source whose nectar is consumed by the bees is replaced with new food source discovered by the scouts. In the ABC algorithm, it is simulated by a function randomly creates new position and replacing it with the consumed one.
- Continue in cycling if the stopping criteria is not met (in this work is the fitness function used in PSO in the previous section). The block diagram of ABC algorithm work with PI-like FLC for Electrohydraulic Servo-actuator is like as the block used in figure (9) by using ABC instead of PSO.

7. PI-like FLC for Electrohydraulic Servo-actuator Position Based on PSO and ABC algorithms:

The fitness function selection for both PSO and ABC algorithms is:

*F=RMSE*0.95+ (overshoot*0.05)* …. (21)

For matching the desired position with presence of the external force, the gains of the controller are tuned using PSO and ABC algorithms. More than fifteen trials have been done to find the optimized controller parameters. Table (3) illustrates the best two trials parameters by PSO that gave best gains according to correction criteria (Percentage of enhancement between the fitness found by the PI-like FLC and the PSO-based controller) that satisfied the required fitness 0.0009 and 0.0008 respectively, and Table (4) illustrates the gains of tests obtained by PSO. Table (3) illustrates that the second trial gives the higher percentage of correction.

Table3: Experiments for the PI-like FLC PSO-based.

Trial	Fitness by intelligent controller (PI-like FLC)	Global Fitness using PSO	C1	C2	Number of Iterations	Number of particles	Percentage of enhancement
1	0.0012	0.0009	3	2.9	40	30	25%
2	0.0012	0.0008	3	2.7	40	35	33.333 %

Table 4: Gains obtained by PSO for the PI-like FLC.

Gains	KP	Ki	KO
First trial	11.6036	90.9983	308.1274
Second trial	17.1708	202.9259	316.9109

Figure (10) shows fitness response obtained using PSO-based controller for the second trial. For the same fitness function at equation (21) the ABC algorithm was also used for tuning the PI-like FLC for getting the desired position of minimized position error, minimized rise time and minimized oscillation under the same applied force. Table (5) illustrates the best two trials parameters by ABC that gave best gains according to correction criteria (Percentage of enhancement between the fitness found by the PI-like FLC and the ABC-based controller) that satisfied the required fitness minimum than 0.0008 and Table (6) illustrates the gains for each case obtained by ABC. Table (5) illustrates that the second trial gives the higher percentage of correction. Figure (11) shows fitness response obtained using ABC-based controller for the second trial.

Table 5: .Experiments for the PI-like FLC ABC-based.

Trial	Fitness by intelligent controller (PI-like FLC)	Global Fitness using ABC	a	b	Number of Iterations	Number of Bees	Percentage of enhancement
1	0.0012	0.00073003	0.7	0.1	40	40	39.164%
2	0.0012	0.00072880	0.9	0.1	30	60	39.266 %

Table 6: Gains obtained by ABC for the PI-like FLC.

Gains	KP	Ki	KO
First trial	35.6228	400.0000	400.0000
Second trial	35.2147	400.0000	393.2774

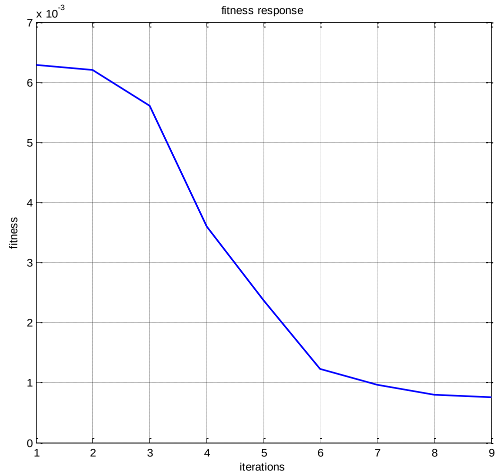


Figure 10: .Fitness response of best trial for PSO-based controller.

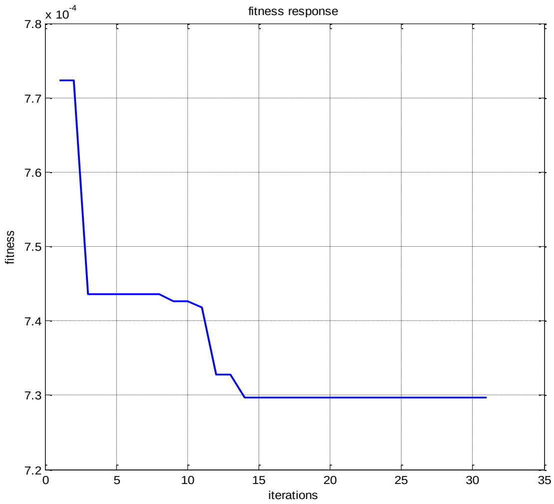


Figure 11: .Fitness response of the best trial for ABC-based controller.

From the trials of PSO and ABC it can be noted that the minimal fitness achieved by PSO is 0.0008 (global minimum by PSO), but by ABC the fitness achieved is 0.00072, this proves that the ABC algorithm has the ability of getting out of a local minimum in the search space and finding the global minimum. Figure (12) shows the position response by conventional and intelligent controllers and Figure (13) shows the control signal for each controller.

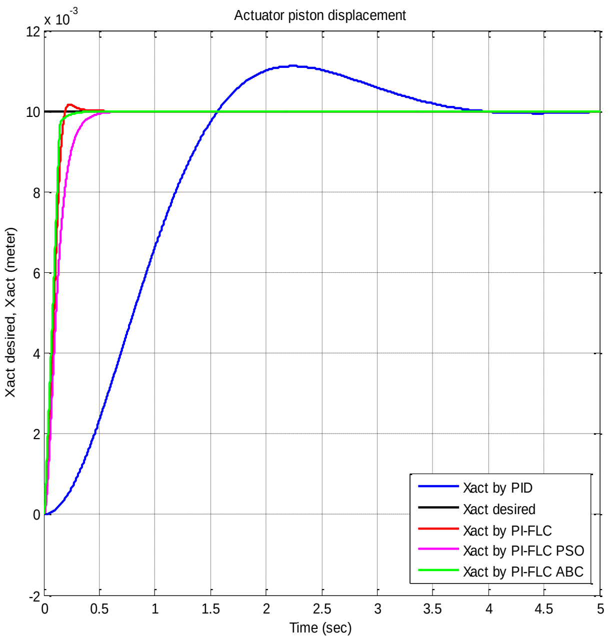


Figure 12: Electrohydraulic servo-actuator position response by conventional and intelligent controllers.

Figure (12) shows the performance of position response by PI-like FLC based on PSO and ABC. These responses have the minimized error with no oscillation and minimized settling time by PSO of (0.307 sec) and by ABC the settling time of (0.141 sec). These responses by the controller's gains were found in the second trial of tables 4 for PSO and 6 for ABC algorithm.

Figure (14) shows the flow rate from the actuator ports by these two best position responses. Figure (15) shows the actuator piston velocity with the spool velocity of these two best position responses.

Figure (14) shows the optimal flow rate by controllers based on PSO and ABC.

Figure (15) shows the velocity of the actuator piston with valve spool that accompaniment for the position response by controllers based on PSO and ABC optimization algorithms.

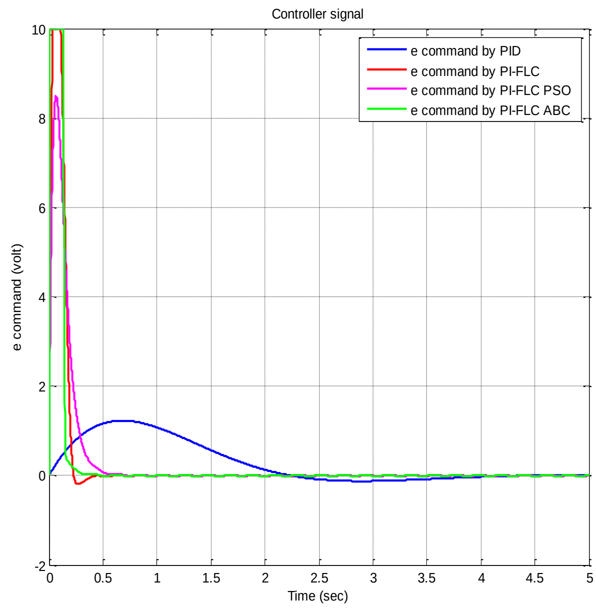


Figure 13: . Control signal of conventional and intelligent controllers to matching the desired position.

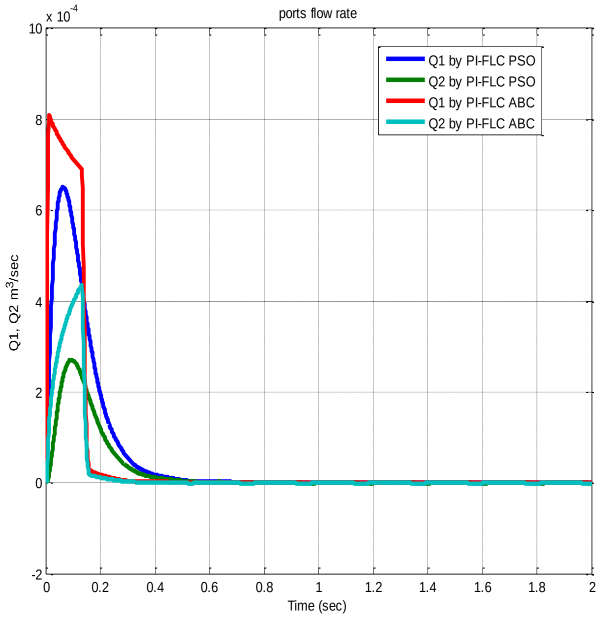


Figure 14: Flow rate from actuator ports by PI-FLC based on PSO and ABC.

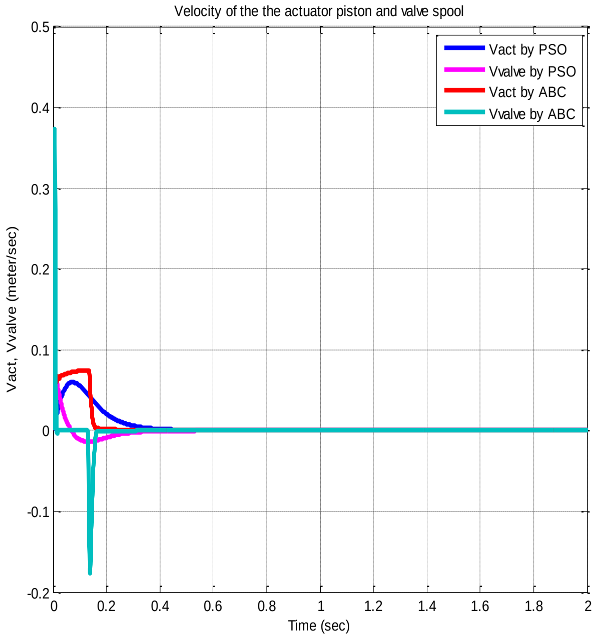


Figure 15: Actuator piston velocity with valve spool velocity by PI-FLC based on PSO and ABC.

8. Conclusions:

It can be concluded from this work that: the PI-like FLC was designed to control the position of simulated Electrohydraulic Servo-actuator model. This controller design to deal with actuator nonlinear model and to control the position response for the actuator with presence of external force effect in high performance. This performance can achieved by minimize the position steady state error, oscillation, settling time in the controlled system position response. The proposed PI-like FLC has settling time of 0.168 (sec). Comparing this result with the result of controller used in reference [3], which has settling time of 0.341 (sec). The enhancement between this work and the work of reference [3] in settling time is 50.733%. Also there is a small fluctuation around the desired position because of the controller of reference [3] does not compensate the effect of external force and nonlinearities in the actuator model.

PSO and ABC algorithm are used to tune the gains of PI-like FLC for minimizing the position steady state error, the settling time and the oscillation. This using result in enhancement in responses as shown in table (3) with second PSO trial the enhancement about 33.333% and in table (5) with second ABC trial about 39.266% compared with the PI-like FLC fitness of this work.

Finally, using the intelligent PI-like FLC for nonlinear Electrohydraulic Servo-actuator model gave high position performance based on optimization algorithms for finding the global minimum solution that subjected to the specified fitness function.

Nomenclature

A_s = Spool area
ABC=Artificial Bee Colony
b_{act}= Damping coefficient
C_1= Valve port of chamber 1
C_2= Valve port of chamber 2
C_d = Orifice coefficient
C_{f1}= Fluid capacitance of chamber 1
C_{f2} =Fluid capacitance chamber 2
$e_{command}$ = Command voltage – the control signal
fit_i = fitness value of the solution (i)
F_{ext} = Externally applied force
FLC= Fuzzy Logic Controller
k_{act} = Stiffness coefficient
m_{act} = Mass coefficient
n = Swarm size
P_1 =Chamber 1 pressure
P_2 =Chamber 2 pressure
P_s = Supply pressure
PI= Proportional Integral
PID= Proportional Integral Derivative
PSO= Particle Swarm Optimization
P_i= Probability value
Q_1 = Flow out of valve port C_1
Q_2 = Flow out of valve port C_2
RMSE=Root Mean Squared Error
v_i= Velocity of the particle
v_{act} = Actuator velocity
v_{valve}= Valve spool velocity
V_1= Volumes of chamber 1
$V2$= Volumes of chamber 2
w_{valve} = Valve port width
x_{valve}= Valve spool position
x_{act} = Actuator position
x_{actd} = Actuator desired position
x_i = Current particle position
x_p best =previous best position
x_g best =global best position
β =Bulk modulus
ρ = Fluid mass density
χ= Constriction coefficient

9. References :

[1] Hassan. A. A. and Jassim. M. T.,"Design and Analysis of Electro-Hydraulic Servo System for Speed Control of Hydraulic Motor,"Journal of Engineering, Baghdad, Iraq, Vol. 19, No. 5, pp. 562-573, May, 2013.

[2] Mondal. N. and Datta. B. N.,"A STUDY ON ELECTRO HYDRAULIC SERVOVALVE CONTROLLED BY A TWO SPOOL VALVE,"International Journal of Emerging Technology and Advanced Engineering, Howrah, India, Vol. 3, Special Issue 3: ICERTSD 2013, pp. 479-484, Feb, 2013.

[3] Haitao. Q., Yan. L. and Yongling F., "Modeling and Simulation of Dissimilar Triplex Redundant Hybrid Actuation System,"Journal of

Applied Sciences, Beijing, China, pp.1564-1569, 2013.

[4] Detiček. E. and Župerl. U.," An Intelligent Electro-Hydraulic Servo Drive Positioning, "Journal of Mechanical Engineering, Slovenia, USA, pp. 394-404, March, 2011.

[5] Bohdan. T. K., John F. G. and Lowen S., *Dynamic Modeling and Control of Engineering Systems*, Cambridge University Press, 3rd Edition, 2007.

[6] Reznik. L., *Fuzzy Controllers*, First Edition Newness, *Melbourne* Australia, 1997.

[7] Yang. X., Yuan. J., Yuan. J. and Mao. H., "A Modified Particle Swarm Optimizer with Dynamic Adaptation," Elsevier –Applied Mathematics and Computation, Vol. 189, Issue 2, pp. 1205–1213, 15 June, 2007.

[8] Kennedy. J. and Eberhart. R., "Particle Swarm Optimization," IEEE Transactions on Evolutionary Computation, Washington, USA, pp. 1942-1948, 1995.

[9] Majid. M. Z., Mohammad. M. F. and Mahdi. A. S., "Type-2 Fuzzy Control for a Flexible-joint Robot Using Voltage Control Strategy," Springer - international Journal of Automation and Computing, Vol. 10, Issue 3, pp. 242-255, June, 2013.

[10] Karaboga. D. and Basturk. B., "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, "Journal of Global Optimization , Vol. 39, Issue 3, pp. 459-471, Nov, 2007.

[11] Akhilesh. K. M., Anukaran K., Navin. K. S. and Vivek. K. M.,"Speed Control of DC Motor Using Artificial Bee Colony Optimization Technique," Universal Journal of Electrical and Electronic Engineering, pp.68-75, Dec, 2013.

[12] Kesarkar. A. Anil and Selvaganesan. N.,"Tuning of optimal fractional -order PID controller using an artificial bee colony algorithm,"Systems Science & Control Engineering Journal, India, Vol. 3, pp.99–105, Dec, 2014.

[13] Ebrahim. E. A.,"Artificial Bee Colony-Based Design of Optimal On-Line Self-tuning PID Controller Fed AC Drives," International Journal of Engineering Research, Cairo, Egypt, Vol. No.3, Issue No.12, pp. 807-811, Dec, 2014.

تصميم مسيطر موقع ضبابي منطقي من نوع PI للصمام الكهربائي الهيدروليكي باستخدام خوارزمية سرب الجسيمات وخوارزمية مستعمرات النحل الاصطناعي

زياد عبدالواحد كرم
قسم تكنولوجيا المعلومات
الجامعة اللبنانية الفرنسية –العراق-اربيل

الخلاصة:

نموذج الصمام الكهربائي الهيدروليكي المؤازر هو نموذج رياضي غير خطي للغاية ويستخدم في منظومات الطيران والفضاء وفي التطبيقات الصناعية والانظمة الروبوتية التي تحتاج سرعة ودقة في الاداء اثناء وجود حمل عالي واضطرابات خارجية. الصمام المؤازر يتكون من اسطوانة هيدروليكية او متحكم دوار يثبت للصمام الكهربائي الهيدروليكي.

هذا العمل يشتمل على نمذجة ومحاكاة الصمام الكهربائي الهيدروليكي المؤازر للسيطرة على مستوى الطيران في طيارة عسكرية عن طريق السيطرة على موقع حركة جناح الطائرة بواسطة الصمام وبوجود قوة الدفع الخارجية. تم استخدام مسيطر ذكي ضبابي منطقي من نوع PI صمم لتتبع الموقع المطلوب التحرك له من قبل الصمام خلال زمن محدد واقل خطأ بالموقع واقل زمن استقرار واقل ذبذبة ممكنة. تم بناء المسيطر باستخدام محاكاة برنامج الماتلاب. وكانت نتيجة زمن استقرار الاستجابة بهذا المسيطر هي 0.168 ثانية مقارنة مع المسيطر المستخدم في المصدر رقم [3] بزمن استقرار 0.341 ثانية، حيث وجدت نسبة تحسين بزمن الاستقرار في المسيطر المقترح وهي 50.733%. أيضا توجد نسبة من التذبذب في المسيطر المستخدم في المصدر المذكور بسبب عدم استخدام مسيطر قادر على التعامل مع الأنظمة الغير خطية. تم استخدام خوارزمية سرب الجسيمات وخوارزمية مجمعات النحل الاصطناعية لإيجاد أفضل المعايير لمسيطر الموقع الضبابي المنطقي نوع PI للحصول على اقل نسبة خطأ مع الموقع المطلوب بوجود تأثير القوة الخارجية على جناح الطائرة. النتائج بينت ان اداء خوارزمية مجمعات النحل الاصطناعي أفضل من اداء خوارزمية سرب الجسيمات في تعيير المسيطر للحصول على اقل نسبة خطأ في الموقع واقل زمن استقرار واقل تذبذب حركي. حيث كانت نسبة صلاحية التعيير مع خوارزمية سرب الجسيمات هي 0.0008 ومع خوارزمية مجمعات النحل الاصطناعية هي 0.00072.