*i*Journals: International Journal of Software & Hardware Research in Engineering (IJSHRE)
ISSN-2347-4890
Volume 10 Issue 1 January 2022

Optimized PID Controller based on Artificial Bee Colony Algorithm for Robot Arm

Author: Salih M. Attya
College of Electronics Engineering / Ninevah University /Mosul, Iraq
salih.attya@uoninevah.edu.iq

DOI: 10.26821/IJSHRE.10.1.2022.100102

Abstract:
In this paper, an efficient method based on Artificial Bee Colony (ABC) metheuristic is implemented for tuning PID controllers In this paper, an efficient method based on Artificial Bee Colony (ABC) metaheuristic is implemented for tuning PID controllers In this paper, an efficient method based on Artificial Bee Colony (ABC) metaheuristic is implemented for tuning PID controllers. In this paper, an efficient method based on Artificial Bee Colony (ABC) metaheuristic is implemented for tuning PID controllers.

This paper deals with design and optimization of a proportional, integral and derivative (PID) controller tuned by Ziegler-Nichols method (ZN) by using artificial bee colony (ABC) optimization algorithm. This algorithm was used to find the optimal parameters of the controller to have optimal performance of a robot arm. A comparative study has been made to highlight the advantage of using ABC-based controller over a ZN-based controller. The validity of the controller tuning algorithms was tested in process control where the optimal tuning procedure of PID controller has also been performed by ABC with different colony sizes. From the perspectives of time-domain performance criteria, such as rise time, overshoot, settling time and steady-state error, the controller tuned by ABC gives better dynamic performances than controllers tuned by the ZN.

Keywords: Robot arm, Artificial Bee Colony (ABC), PID controller, Ziegler-Nichols (ZN) Method.

1-Introduction:

The arm of a robot has a number of joints where each joint gives the robot arm at least one degree of freedom. The modeling approach that has been used for the robot joints treats each joint of the robot as a simple joint servomechanism while ignoring the effect of other joints movements. As an actuator to drive the robot arm in industrial robots, hydraulic or pneumatic actuators could be used but electrical motors are more common due to their excellent speed and position control characteristics. For this reason, the actuator that has been used is assumed to be DC servo motor where it has been used widely in industry. Another assumption is that there is a gear box has been used since it is essential to increase the torque and reduce the rotational speed [1]. Proportional-Integral-Derivative (PID) controllers have been widely used for speed and position control of robot arms.

This paper endeavors to design a PID controller using ZN then using another method to design and optimize PID controller using Artificial Bee Colony Optimization (ABC) method to control the position of robot arm joints. Design methods aim to find the optimal gains of PID controller for the given system (plant). It is important to analyze the contribution of each part of the PID controller in order to understand the importance of each of them individually where the proportional gain makes the controller respond to the instant error while the integral and derivative gains takes in account the pervious errors and the rate of change of the error to eliminate steady state error and prevent overshoot respectively [2].
Using ABC algorithm to perform the tuning of the controller will result in the optimum controller being evaluated for the system every time [3].

2- Robot Control System Model:

The block diagram of the single-joint robot arm that it is connected to the base by motor and gears is shown in figure (1) [4]. By neglecting the gravity force (small value) and for rigid robot arm, the dynamic behavior of the robot arm control system is given by equation (5).

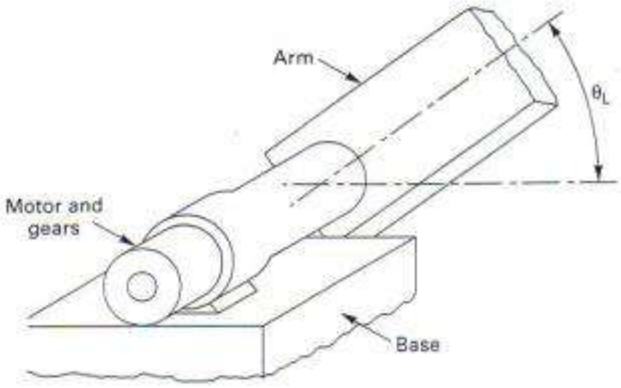


Fig. (1) Single-Joint Robot Arm

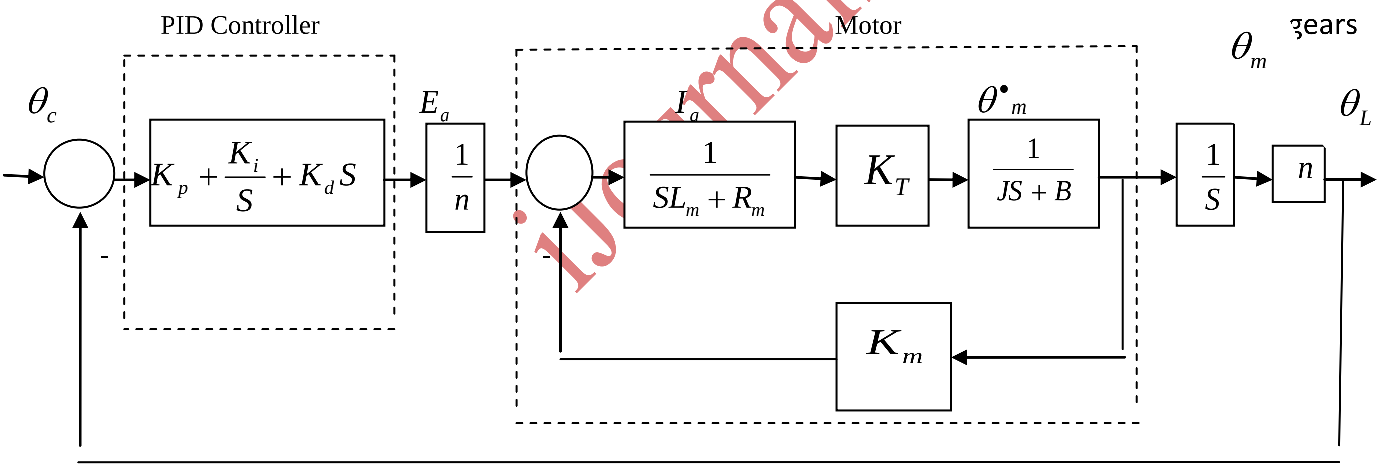


Fig. (2). Robot Joint Control System

$$e_a(t)=R_m i_a(t)+L_m\frac{di_a(t)}{dt}+e_m(t)\text{-----------------}1$$

$$e_m(t)=K_m\frac{d\theta_m(t)}{dt}\text{-------------------------}2$$

$$T_m=K_T i_a(t)\text{-----------------------------}3$$

$$T_m = J\frac{d^2\theta_m(t)}{dt^2} + B\frac{d\theta_m(t)}{dt} ---------------------4$$

$$J = J_m + n^2 Jl ----------------------------5$$

$$B = B_m + n^2 Bl ----------------------------6$$

$$\theta_L = n\theta_m -------------------------------7$$

After simplification and taking the ratio of $\theta_L(s) / E_a(s)$ the open loop transfer function is represented in equation (8)

$$\frac{\theta_L(s)}{E_a(s)} = \frac{K_T n}{JL_m S^3 + (R_m J + BL_m)S^2 + (K_T K_m + R_m B)S} --------8$$

Where.

R_m = armature- winding resistance in ohm.

L_m = armature - winding inductance in Henry.

i_a = armature - winding current in ampere.

e_a = armature voltage in volt.

e_m = back emf voltage in volt.

K_m = back emf constant in volt / (rad/sec).

T_m = torque developed by the motor in N.m

K_T = motor torque constant in N.m/A

J = moment of inertia of motor and robot arm in kg. m^2/rad.

B = viscous - friction coefficient of motor and robot arm in N.m/rad /sec.

θ_m = angular displacement of the motor shaft in rad.

θ_L = angular displacement of the robot arm in rad.

θ_c = angular displacement of the reference input in rad.

n = gear ratio N_1 / N_2

The robot arm control system under study has the following parameters.

$R_m = 21\ \Omega$, $L_m = 2\ \text{H}$, $K_T = 38\ \text{N.m/A}$, $J = 2\ \text{kg.m}^2/\text{rad}$, $B = 1\ \text{N.m/rad/sec}$, $K_m = 0.5\ \text{V/(rad/sec)}$ and $n = \frac{1}{20}$.

The block diagram of the servo control system for one of the joint of a robot is shown in Figure (3)[5].

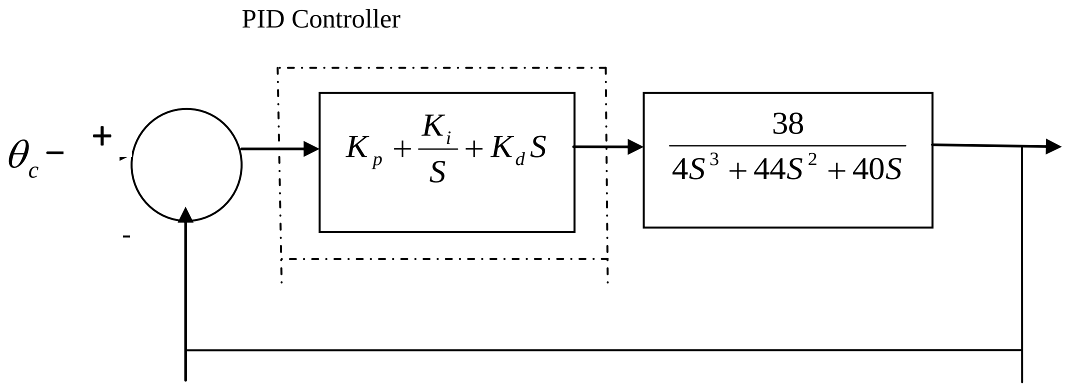


Fig. (3). Joint Control System for A Robot Arm.

3- Ziegler -Nichols Rule for Tuning PID Controller:

Choosing controller gains randomly causes unexpected system response, it can even drive the system to the un-stability region. For this reason, parameters tuning of PID controller is essential in order to have accepted system performance. Controller tuning involves the selection of values of k_p, K_i and K_d. The ZN method that has been used is close loop resonant method, in this method, a proportional gain is added in series with the system and it will be increased until the close loop response of the system became constantly oscillating. At this point, the value of k_u (Resonant gain) will be calculated after applying Routh-Hurwitz method then the value of the resonant peak T_u will be measured form the value of w (rad/s) after substitute s= jw then make $T_u = 2\pi/w$. After apply the steps above, it has been found that the value of Ku = 11.57 and T_u= 2 sec. Referring to table 1, values of K_p, K_i and K_d will be calculated [1].

Table 1: Ziegler-Nichols Tuning Rule

Type of controller	K_p	K_i	K_d
P	$0.5K_u$	0	0
PI	o.45K_u	$0.54\ K_u/\ T_u$	0
PID	$0.6K_u$	$1.2\ K_u/\ T_u$	$0.075\ K_u{*}\ T_u$

So the gains values will be shown in table (2)

Table 2: PID Controller Gain Values from ZN

Gain Coefficient	K_p	K_i	K_d
Values	6.94	6.94	1.73

The values shown in table (2) is considered as the start point and it required another fine tuning manually, this tuning could be done easily using MATLAB. After finishing the tuning, the new values are approximately twice the value suggested by the ZN tuning rule as shown in table (3) [1].

Table 3: PID Controller Gain Values from ZN after tuning

Gain Coefficient	K_p	K_i	K_d
Values	15.26	6.94	8.39

The standard transfer function of the parallel PID controller is shown in equation (9)

$$G_c(s) = \frac{K_d S^2 + K_p S + K_i}{S} ----------(9)$$

After applying values from table (3), controller transfer function G_c (s)

$$G_c(s) = \frac{8.39\,S^2 + 15.26\,S + 6.94}{S} -----(10)$$

The step response of the close loop system with conventionally PID controller is shown in Figure (4).

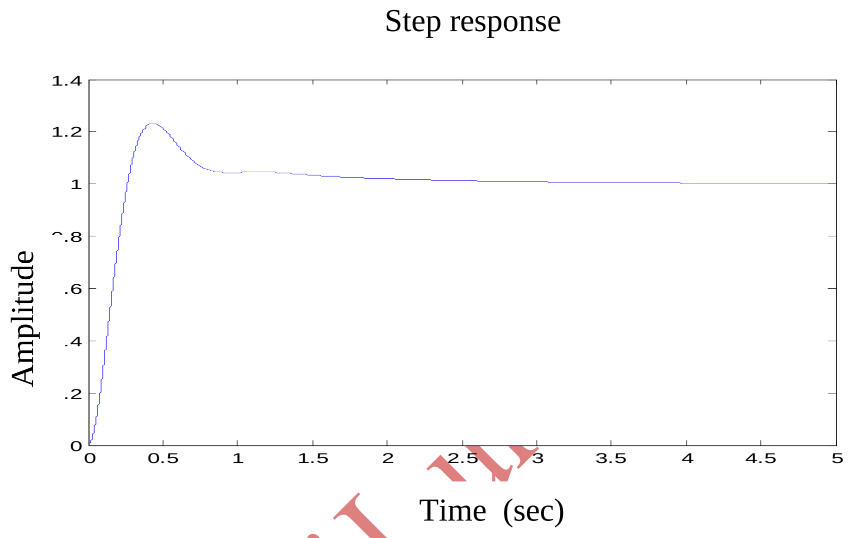


Fig (4). System step

response with Conventionally Tuned PID Controller

From figure (4), response parameters, Rise time, Maximum overshoot and settling time, can be analyzed as shown in table (4)

Table 4: Parameters for ZN-PID

Controllers Parameters	**ZN-PID**
Tr (sec)	0.2
Ts (sec)	2
Mp %	23.9
Kp	15.26
Ki	6.94
Kd	8.39

For the robot, the required system specifications, are shown in table (5). These specifications should be met, so, Artificial Bee Colony optimization will be used to further improvement for the system specifications [3].

Table 5: System Requirements

System Specification	Max Over Shoot	Rise Time (Sec)	Settling Time (Sec)
	< 10 %	< 0.4	< 0.9

4. Artificial Bee Colony Algorithm (ABC)

In real life, each member in bee colony has his own specific task, all of individuals are trying to increase the nectar amount to the maximum. This is performed by self-organizing and sufficient effort division. There are three main groups of artificial bees:

1- Employed bees
2- Onlookers
3- Scouts

Colony can be also divided into two halves where the first half is the employed artificial bees while the second half are the onlookers. Each employed bee looking for different resource of food, so the number of food sources will equal to the number of employed bees. The employed bee who's the food source has been abandoned by the bees becomes a scout. In term of optimization, food source position considered as possible solution while the nectar amount of the food source represents the fitness of the solution so the number of the employed bees or the onlooker bees is equal to the number of solutions in the population. [6-7]

A: ABC Algorithm

Steps (pseudo-coding) to initialize the artificial BA:

1. Initialize the population of solutions:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad ---- \quad (11)$$

Where,

x_{ij}, i = 1. . .SN,(SN is the number of food source) j = 1. ….D.(is the dimension of problem for optimization of PID [namely K_d ,K_i and K_p] ,there is D=3.) , ϕ is also a random number in the domain[-1,1] and k = 1. . . SN .

2. Evaluate the population (colony size).
3. Cycle=1
4. Repeat
5. Produce new solutions x_{ij} for the employed bees by using (4) and evaluate them.
6. Apply the greedy selection process.
7. Calculate the probability values Pi,j for the solutions xi,j by (4 & 3).

$$pi = \frac{fit_i}{\sum_{n=1}^{sn} fit_n} \quad ---- \quad (12)$$

8. Produce the new solutions xi,j for the on looking from the solutions xi,j selected depending on Pi,j and evaluate them.
9. Apply the greedy selection process.
10. Determine the abandoned solution for the scout, if exists, and replace it with a new randomly Produced solution xi,j by (4& 5).

$$x_i^j = x_{\min}^j + \phi(x_{\max}^j - x_{\min}^j) \quad --- \quad (13)$$

Where , ϕ is also a random number in the domain[0,1].

11. Memorize the best solution achieved so far.
12. Cycle = Cycle+1.

13. Until Cycle = MCN (Maximum Cycle Number).

B: Objective Function

The proposed Fitness function for the optimization of parameters of PID controller is defined as:

$$F= W_{max} . (1-\exp(-0.5)) . (Mp+Ess)+ W_{min} . \exp(-0.5) .(ts-tr)$$

where

Mp: Maximum Overshoot
Ess :steady state error
tr : Rise Time
ts : Settling time
Wmax: Maximum Inertia Weight
Wmin: Minimum Inertia Weight

The Parameters of ABC algorithms for optimizing the PID controller is shown in table 6.

Table 6: Parameters of ABC algorithms

Parameter	Values
Colony size	40
No. of Iterations	100
Maximum Speed	10
Maximum Inertia Weight(Wmax)	0.9
Minimum Inertia Weight(Wmin)	0.4

5.ABC-PID Controller

The ABC algorithm was mainly utilized to determine three optimal controller parameters *Kp*, *Ki*, and *Kd*, such that the controlled system could obtain a good output response. In this paper, the ABC algorithm is applied for searching the PID controller parameters. A set of good control parameters K_p, Ki *and* K_d can achieve a good output response for the system and result in minimization of performance criteria in the time domain including the settling time (T_s), rise time (T_r), maximum overshoot (%OS) and steady state error (e_{ss}). The ABC-PID controller for robot arm is shown in figure 5. [8]

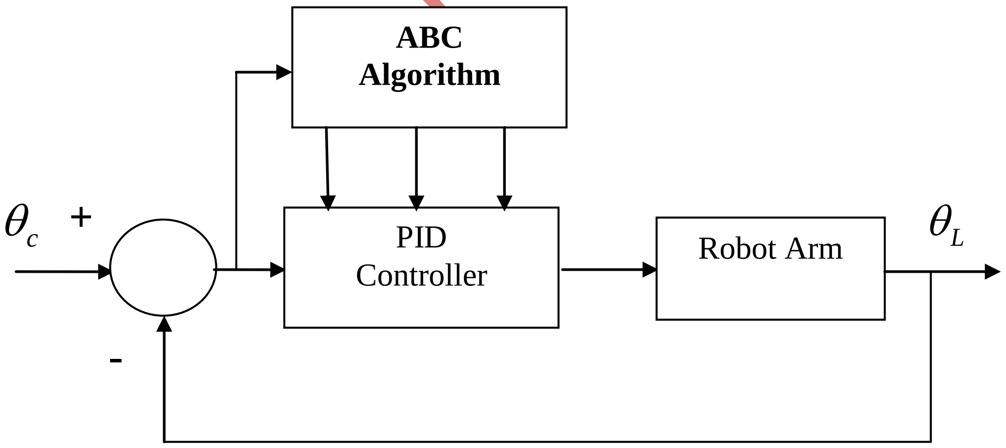


Figure 5. ABC-PID Controller Design for Robot Arm.

6.Simulation and Results

The unit step response of robot using ABC-PID controller are shown in figure 6 and 7 with colony size 20 and 40 respectively.

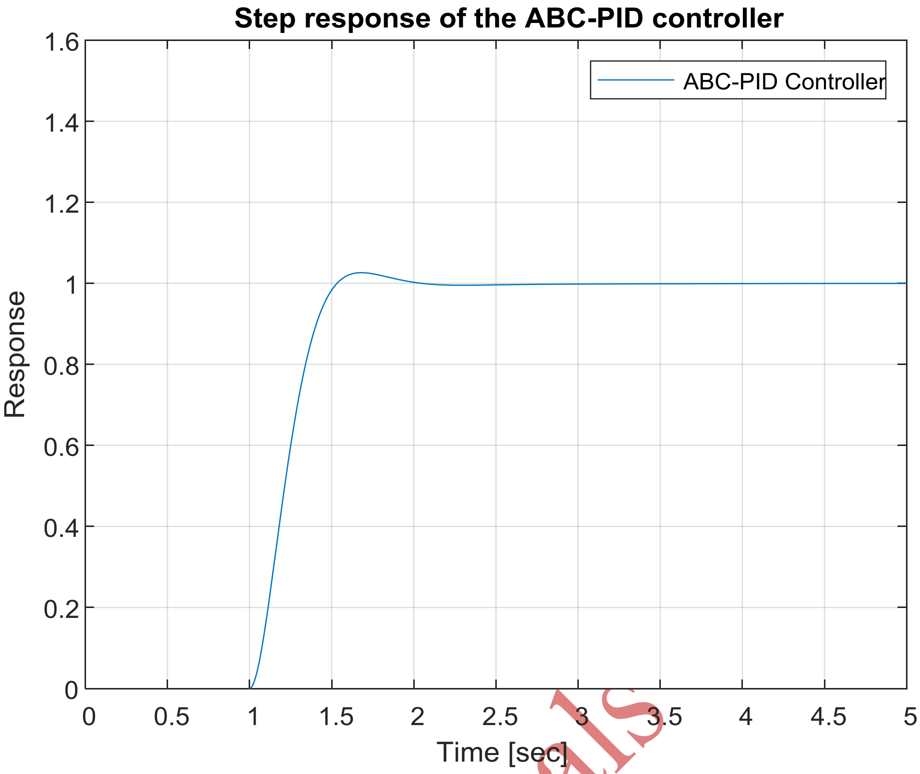


Fig.6: Robot arm step response of ABC_PID Controller with colony size 20.

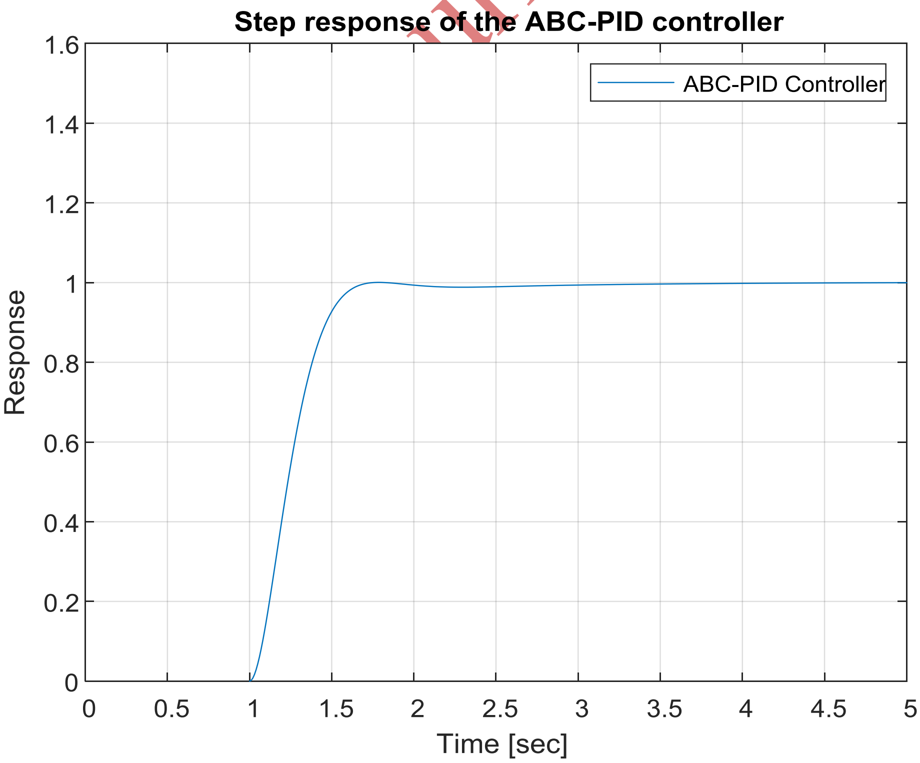
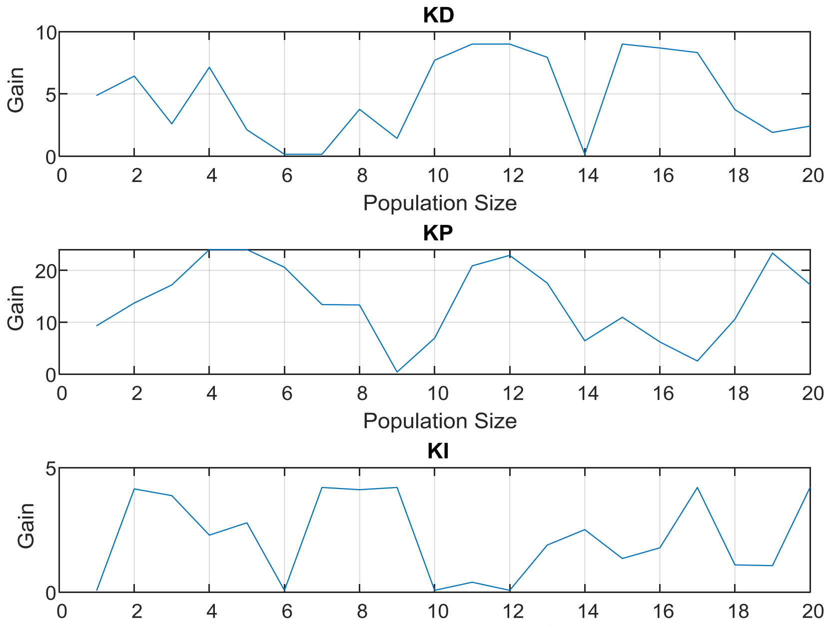


Fig.7: Robot arm step response of ABC_PID Controller with colony size 40.

The convergence curve for each gain is called as particle for Kp, Ki and Kd are plotted to give an idea how the ABC Algorithm converged to its final value has been illustrated in Figures 8, and 9 with colony size 20 and 40 respectively.



.

Fig.8: ABC_PID Controller parameter with colony size 20.

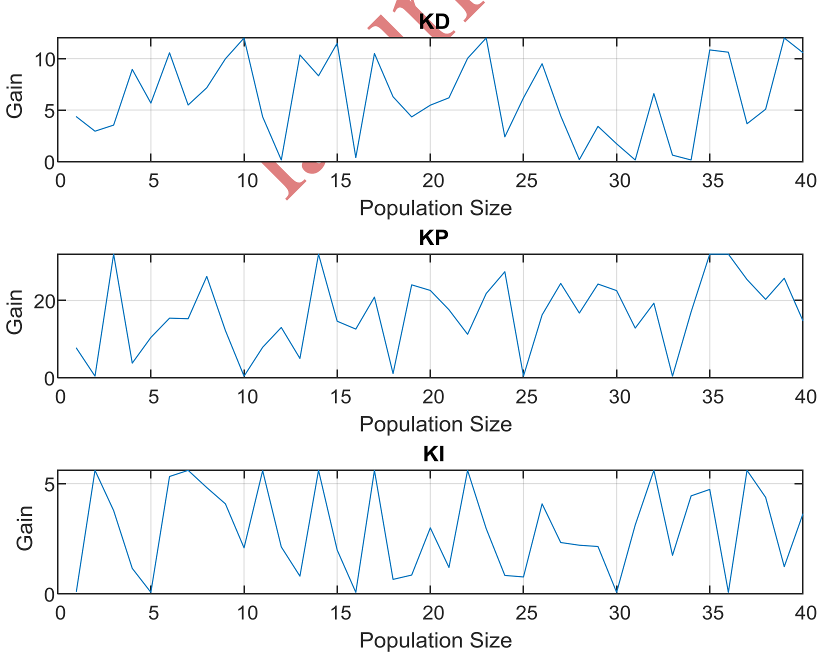


Fig.9: ABC_PID Controller parameter with colony size 40.

The parameters of the two controllers ZN_PID and ABC_PID are shown in table 7.

Table 7: Parameters for ZN-PID and ABC-PID controllers

Controllers Parameters	**ZN-PID**	**ABC-PID with colony size 40**	**ABC-PID with colony size 20**
Tr (sec)	0.2	0.387	0.332
Ts (sec)	2	0.607	0.793
Mp %	23.9	0.0563	2.63
Kp	15.26	3.8375	4.6682
Ki	6.94	0.0160	0.001
Kd	8.39	4.3535	4.8809

8-Conclusion:

The step responses under PID controller tuned by ABC for different colony size 40 and 20 are better than the response under PID controller tuned by Zigler–Nichols (ZN). However, the ZN method is good for giving the designer the initial guess for the PID tuning. The step response under PID controller tuned by ABC is better in terms of minimizing the max overshoot and the settling time. From the perspectives of time-domain performance criteria, such as settling time, rise time, overshoot, and steady-state error, the controller tuned by ABC gives better dynamic performances than controllers tuned by the ZN.

References:

[1] K Ogata, "Modern Control Systems", University of Minnesota, Prentice Hall, 2003

[2] M. S. Alam, M. M. Islam, "Artificial Bee Colony algorithm with Self-Adaptive Mutation: A novel approach for numeric optimization", IEEE Region 10 Conference In TENCON, pp. 49-53,(2011).

[3] S. Pareek, M. Kishnani, R. Gupta, "Application of Artificial Bee Colony Optimization For Optimal PID Tuning", IEEE International Conference on Advances in Engineering & Technology Research, (2014).

[4] L.phillips and D.Harbar, "Feedback Controller System ",3rd edition , Prentice Hall, 1996

[5] Chun Htoo Aung, Khin Thandar Lwin, and Yin Mon Myint," Modeling Motion Control System for Motorized Robot Arm using MATLAB" World Academy of Science, Engineering and Technology 42, 2008

[6] A. A. Kesarkar, N. Selvaganesan, "Tuning of optimal fractional-order pid controller using an artificial bee colony algorithm", Journal of Systems Science & Control Engineering, Vol. 3, No. 1, pp.99-105,(2015).

[7] W. Liao, Y. Hu, H. Wang, "Optimization of PID control for DC motor based on artificial bee colony algorithm", IEEE International Conference on Advanced Mechatronic Systems, pp. 23-27,(2014).

[8] S. Pareek, M. Kishnani, R. Gupta, "Application of Artificial Bee Colony Optimization For Optimal PID Tuning", IEEE International Conference on Advances in Engineering & Technology Research, (2014).