DOI: https://doi.org/10.29121/ijetmr.v5.i9.2018.289

IJETMR

International Journal of Engineering Technologies and Management Research

A Knowledge Repository

OPTIMIZATION OF PID PARAMETERS BASED ON PARTICLE SWARM OPTIMIZATION FOR BALL AND BEAM SYSTEM

Haytem Ali [1], Abdulgani Albagul [*2], Alhade Algitta [1]

[1] Department of Control Engineering, Collage of Electronic Technology, Baniwalid, Libya

[*2] Engineering and Information Technology Research Center, Baniwalid, Libya

Abstract:

This paper introduces the application of an optimization technique, known as Particle Swarm Optimization (PSO) algorithm to the problem of tuning the Proportional-Integral-Derivative (PID) controller for a linearized ball and beam control system. After describing the basic principles of the Particle Swarm Optimization, the proposed method concentrates on finding the optimal solution of PID controller in the cascade control loop of the Ball and Beam Control System. Ball and Beam control system tends to balance a ball on a particular position on the beam as defined by the user. The efficiency of Particle Swarm Optimization algorithm for tuning the controller will be compared with a classical method, Trial and Error method. The comparison is based on the time response performance. The two tuning methods have been developed by simulation study using Matlab\ m-file software. The evaluations show that Evolutionary method Particle Swarm Optimization (PSO) algorithm gives a much better response than trial and error method.

Keywords: Particle Swarm Optimization; Ball and Beam System; PID Controller; PID Tuning Method.

Cite This Article: Haytem Ali, Abdulgani Albagul, and Alhade Algitta. (2018). "OPTIMIZATION OF PID PARAMETERS BASED ON PARTICLE SWARM OPTIMIZATION FOR BALL AND BEAM SYSTEM." *International Journal of Engineering Technologies and Management Research,* 5(9), 59-69.

1. Introduction

The ball and beam system is a simple mechanical system which usually difficult to control [1]. It is one of the most popular and widely used laboratory models for control systems and has always remained one of the favorite introductory control system problems for control engineers [2]. The ball and beam system is connected to real control problem such as horizontally stabilizing an airplane during landing and in turbulent airflow [3]. The ball and beam system is very easy to understand, and many control techniques can be studied on this system to cover many classical control design methods. It consists of rigid beam, which is free to rotate in the vertical plane at the pivot, with a solid ball rolling along the beam [4]. As the name implies, in this system, a ball is placed on a beam and the control system should tend to balance the ball at particular position on the beam as prescribed by the user. The ball and beam system can be categorized into two configurations. The first configuration is shown in figure 1, which illustrates that the beam is supported in the middle, and rotates against its central axis. Most ball and beam systems use this

ISSN: 2454-1907
DOI: 10.5281/zenodo.1443549

configuration. This type of configuration is normally called as ‘Ball and Beam Balancer’. This type has the advantage of easiness of building and the simplicity of the mathematical model.

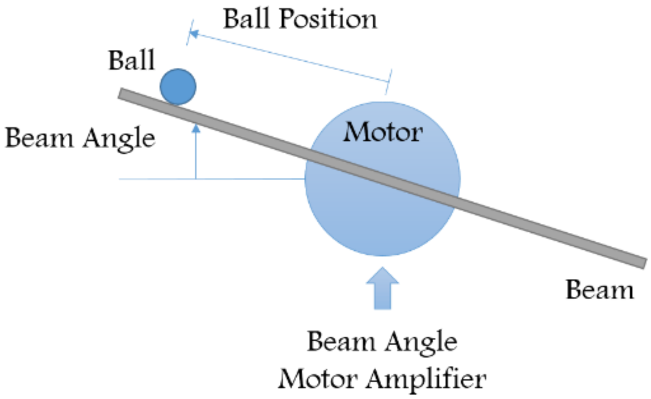

Figure 1: Beam supported at the centre

The second configuration is shown in figure 2. In this type, the beam is supported on both ends by two level arms. One of the level arms acts as the pivot, and the other is coupled to motor output gear. The disadvantage is that more consideration of the mechanical parts, which meant adding some difficulties in deriving a mathematical model. This type of configuration is called ‘Ball and Beam Module’. The ‘Quanser’ ball and beam system uses this configuration for its commercial product. The advantage of this system is that relatively small motor can be used due to the existing of gearbox. This type of configuration will be used in this paper.

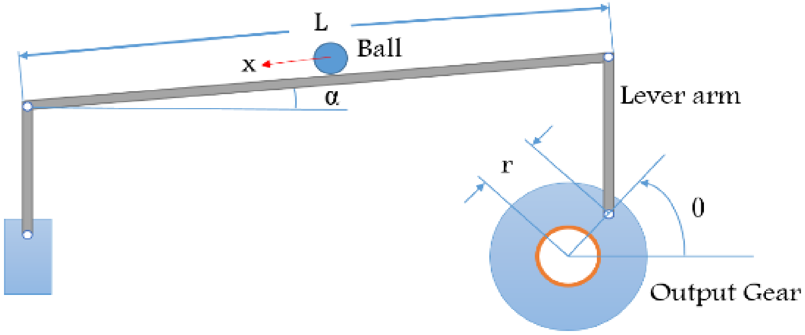

Figure 2: Beam supported at both sides

The ball and beam system is an open loop and important to point out that it is unstable and nonlinear since the ball position changes with acceleration without limit for a fixed beam angle [1]. The control problem can be approximated by linearized the model, hence the linear feedback control such as PID control can be applied and the stability analysis can be determined based on linear state-space model or transfer function. In this paper, the PID controller will be used to stabilize a ball and beam system at its equilibrium position. However, tuning a PID controller to acquire the most optimum results is a tricky task and various methods have been proposed to search the parameters of PID controllers in the past to do so. The classical PID tuning methods include: Ziegler Nichol’s method [5], ITAE equations method [6], Gain and phase margin tuning method [7], SISO tool and so on [8]. Furthermore, many unconventional techniques have been developed over the past few decades, which have proven to be very fruitful in control systems.

The unconventional techniques have been given much attention by many researchers because of their ability to find global optimal solutions. Evolutionary algorithm and Fuzzy logic [9] are examples of such unconventional methods. Particle Swarm Optimization (PSO) algorithm is one of such evolutionary algorithm. The PSO is recently proposed heuristic search method for optimization of continuous nonlinear functions, which can be used in tuning PID controllers [2], inspired by the swarm methodology. The method was derived through simulation of simplified social models such as bird flocking, fish schooling and swarming theory in particular. Kennedy and Eberhart invented particle swarm optimization in 1990's while trying to simulate the choreographed, graceful motions of swarms of birds. Particle swarm optimization has two roots. One of them is to tie to artificial life. It is also related to evolutionary computation such as genetic algorithms and evolutionary programming. The ability of flocks of birds, schools of fish and herds of animals to adept to their environment, to avoid predators and to fined rich sources of foods by implementing an information sharing approach intrigued the inventors of the methodology [10]. There has been much attention in terms of implementation of PSO in control theory. Particle swarm optimization has successfully been applied to a wide variety of problems such as neural networks [11], structural optimization [12], share topology optimization [13] and fuzzy systems [14].

2. Mathematical Modeling Materials and Methods

The mathematical description of this system consists of two separated systems, the first one is the DC servomotor, which is an electromechanical system that receives electrical signal from controller and produces an output as a rotational displacement (angle). The second one is the ball and beam model, which is a mechanical system that receives rotational displacement (angle) from the motor and converts it into a linear displacement.

2.1. Motor Model

The position of a DC motor is observed to have a transfer function of:

$$G_m(s) = \frac{\theta(s)}{V_a(s)} = \frac{K}{s(\tau_1 s+1)} \quad (1)$$

Based on the motor parameters in [15], motor model can be written as follow:

$$G_m(s) = \frac{\theta(s)}{V_a(s)} = \frac{0.7}{s(0.014s+1)} \quad (2)$$

2.2. Ball and Beam Model

As illustrated in figure 2, a ball is placed on a beam that free to roll along the length of the beam at horizontal plane. A lever arm is attached to the beam at one end and a servo gear at the other. The servo gear turns by an angle (θ), and the lever changes the angle of the beam by (α). The force that accelerates the ball as it rolls on the beam comes from the component of gravity that acts parallel to the beam. The ball actually accelerates along the beam by rolling, but we can simplify the derivation by assuming that the ball is sliding without friction along the beam. The mathematical modeling of ball and beam system consists of DC servomotor dynamic, alpha theta

ISSN: 2454-1907
DOI: 10.5281/zenodo.1443549

relation, and ball on the beam dynamic. The dynamic equation of the ball on the beam has been described by [4] using Newton second law as given in equation (3)

$$F_{tx} + F_{rx} = mg\sin(\alpha)$$

$$m\ddot{x} + \frac{2}{5}m\ddot{x} = mg\sin(\alpha) \tag{3}$$

Where:
F_{tx}: Force due to transnational motion
F_{rx}: Force due to rotational motion
m: Mass of ball
g: Gravitational acceleration
α: Beam angle coordinate
$\ddot{x}$: Acceleration of the ball position

The derivation of equation (3) is based on diagram depicted in figure 3. Now, equation (3) can be linearized to obtain a transfer function of the ball and beam system, for small angle, $\sin(\alpha) = \alpha$. Therefore, equation (3) becomes:

$$\ddot{x} = \frac{5}{7}g\alpha \tag{4}$$

By taking the Laplace transform of equation (4), we find

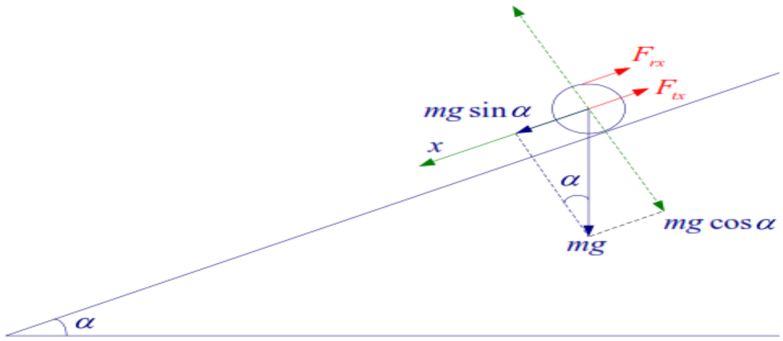

Figure 3: Force acting on the ball and beam system

$$\frac{x(s)}{\alpha(s)} = \frac{5}{7}\frac{g}{s^2} = \frac{7}{s^2} \tag{5}$$

The beam angle (α) can be related to motor gear angle (θ) by approximate linear equation $\alpha L = \theta r$ where d=lever arm offset and L=beam length. Substitute L = 16.75cm and r = 2.54cm will give another transfer function as follows,

$$\frac{\alpha(s)}{\theta(s)} \cong \frac{r}{L} = \frac{2.54}{16.75} \tag{6}$$

2.3. The Complete System Model

Combining the results from the previous subsections, the open-loop transfer function of the ball and beam model with the motor becomes,

DOI: 10.5281/zenodo.1443549

$$G(s) = \frac{x(s)}{V_a(s)} = \frac{0.7}{s(0.014s+1)} \frac{2.54}{16.75} \frac{7}{s^2} = \frac{0.742}{s^3(0.014s+1)} \tag{7}$$

3. Particle Swarm Optimization (Pso) Algorithm

Particle Swarm Optimization algorithm, as depicted in the flowchart in figure 4, is one of such evolutionary algorithm, which can be, used in tuning the PID controllers. The algorithm simulates the movement of birds in flocks. The algorithm works on the scenario of birds randomly searching for food. It begins with a swarm of birds/particles being initialized at random positions in the problem space, each particle in the swarm represents a solution to the problem, each of the particles is treated as a point in N-dimensional space which adjusts its flying according to its own flying experience as well as the flying experience of other particles. Each bird keeps track of its coordinates in the problem space. A record is also kept of the least distance (fitness value) from the food achieved by each individual bird so far, called pbest (personal best). The minimum value of pbest in the swarm is called gbest (global best), that is, the minimum distance achieved by any of the birds in the swarm so far [2].

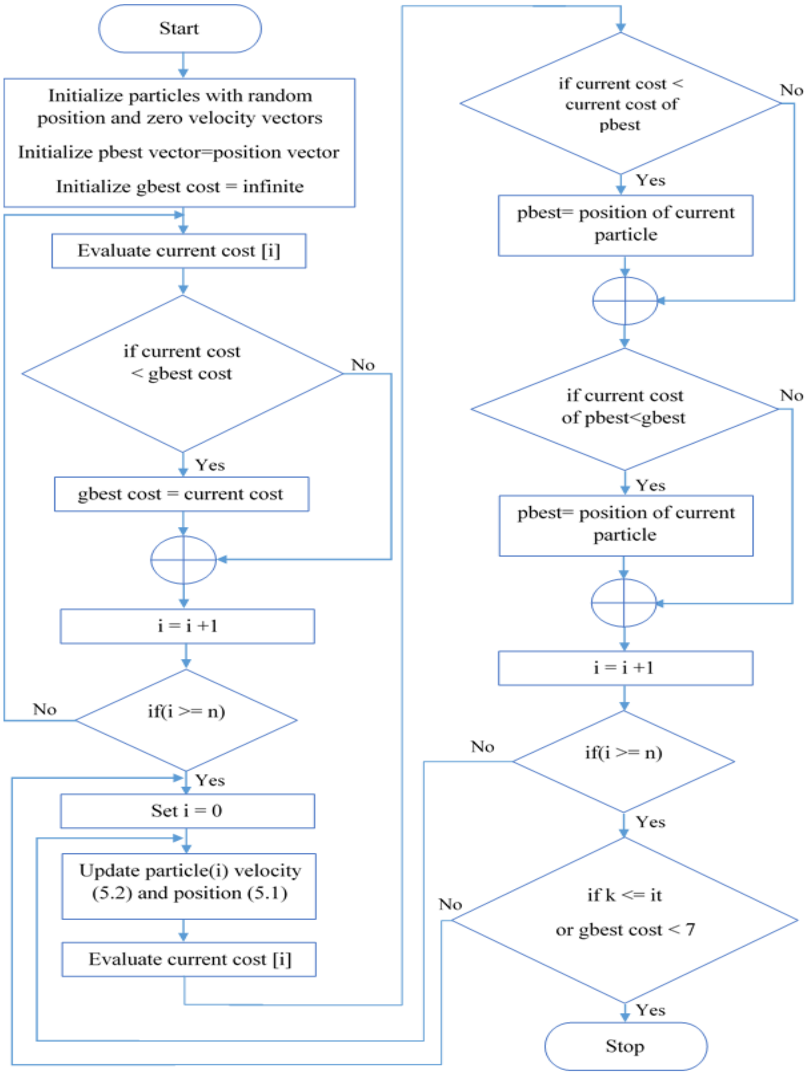
Start
Initialize particles with random position and zero velocity vectors
Initialize pbest vector=position vector
Initialize gbest cost = infinite
Evaluate current cost [i]
if current cost < gbest cost
No
Yes
gbest cost = current cost
i = i +1
No
if(i >= n)
Yes
Set i = 0
Update particle(i) velocity (5.2) and position (5.1)
Evaluate current cost [i]
if current cost < current cost of pbest
No
Yes
pbest= position of current particle
if current cost of pbest<gbest
No
Yes
pbest= position of current particle
i = i +1
No
if(i >= n)
Yes
No
if k <= it
or gbest cost < 7
Yes
Stop

Figure 4: The flow chart of general POS algorithm

ISSN: 2454-1907
DOI: 10.5281/zenodo.1443549

In a physical n-dimensional search space, the velocity and the position of particle (i) are represented as the vectors $X_i = (x_{i1}, \dots, x_{in})$ and $V_i = (v_{i1}, \dots, v_{in})$ in the PSO algorithm. Therefore, the best position of particles and their neighbours best position are $pbest_i = \left(x_{i1}^{pbest}, \dots, x_{in}^{pbest}\right)$ and $gbest_i = \left(x_{i1}^{gbest}, \dots, x_{in}^{gbest}\right)$.

In each iteration, the velocity of each bird is updated and added to the current coordinates of the respective birds.

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{6}$$

The velocity is governed by three factors which are inertia (V_i^k), cognitive influence (pbest) and social influence (gbest). The directions of cognitive and social influences depend on the direction vector of the bird's position from pbest and gbest respectively. Both these influences are updated each iteration and added to inertia, to get the new velocity. Furthermore, randomness is maintained in the system by weighing the influences with random numbers between (0) and (1), called weight factors ($C_1 and\ C_2$) and the inertia with another random number in the same range, called inertia factor (w) [2].

$$V_i^{k+1} = w \times V_i^k + C_1 \times rand1 \times \left(pbest_i^k - X_i^k\right) + C_2 \times rand2 \times \left(gbest_i^k - X_i^k\right) \tag{7}$$

Where; (V_i^k) is the velocity of particle (i) at iteration (k), (X_i^k) is the position of particle (i) at iteration (k), (w) is the inertia weight, (C_1, C_2) is the weight factors, ($rand1, rand2$) denotes random numbers between 0 and 1, ($pbest_i^k$) is the best position of particle (i) until iteration (k), and best position of the group until iteration (k) is denoted as ($gbest_i^k$). Figure 5 shows the trajectory of the particle in the swarm.

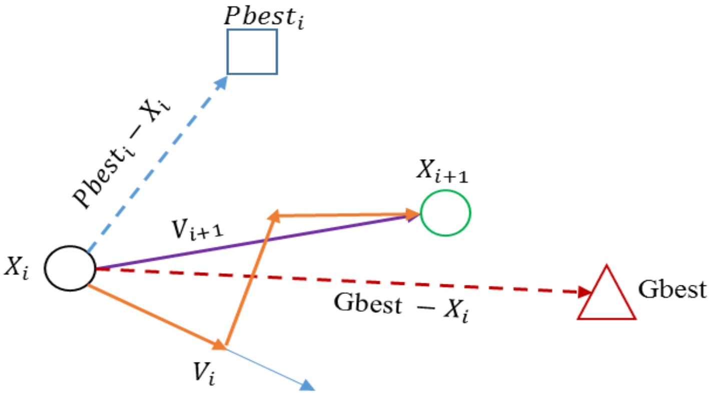

Figure 5: The trajectory of the particle after velocity updating

4. PID Controller Design

It was found that the overall open loop transfer function of the system is a fourth order system. However, it is difficult to design a controller to control a third order and higher order system. Therefore, to make the controller design become easier and realizable, the whole system is divided into two feedback loops as shown in figure 6.

ISSN: 2454-1907
DOI: 10.5281/zenodo.1443549

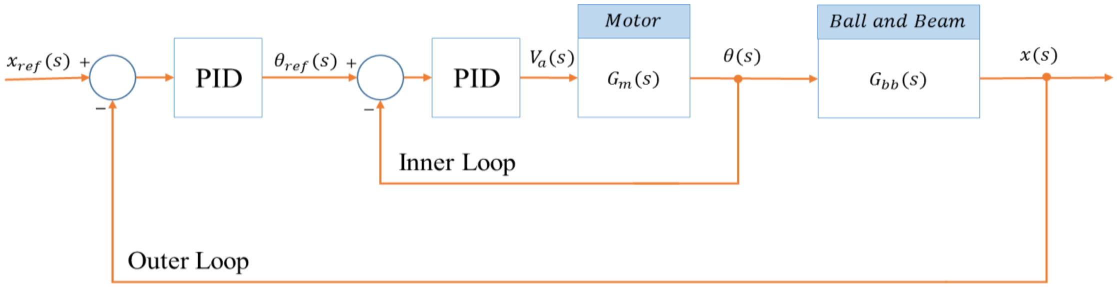

Figure 6: Block diagram of the ball and beam PID control system

The design strategy is to first stabilize the inner loop followed by the outer loop. The purpose of the inner loop is to control the motor angle position. Inner controller should be designed so that the motor angle tracks the reference signal. The outer loop uses the inner feedback loop to control the ball position. Therefore, the inner loop definitely must be designed before the outer loop.

4.1. PID Tuning with Particle Swarm Optimization Algorithm

Before implementing the Particle Swarm Optimization technique on a problem, one has to consider first the number of variables to be tuned in order to optimize the solution to the problem. The number of dimensions of the matrices in the algorithm must equal the number of variables in the system. In addition, certain parameters need to be defined. Selection of these parameters decides largely the ability of global minimization. The maximum velocity affects the ability of escaping from local optimization and refining global optimization. The size of swarm balances the requirement of global optimization and computational cost [2]. Initializing the values of the parameters is as per table 1. Table 2 shows the upper and lower limits set for the PID controller gains. The particles are initialized at random positions within these limits and are not allowed to trespass during the algorithm.

Table 1: PSO selection parameters

Population size	50
Number of iteration	20
Number of variables	3
Maximum velocity	0.9
Minimum velocity	-1
Inertia weight	1
Personal learning coefficient	2
Global learning coefficient	2

Table 2: The bounds imposed on PID parameters

PID Gains	$\mathbf{K_p}$	$\mathbf{K_i}$	$\mathbf{K_d}$
Lower bound	0	0	0
Upper bound	10	10	10

The cost function represents the function that measures the performance of the system. In our case, the cost function for the PSO algorithm is defined as a function of closed loop system performance. The following cost function is proposed and applied.

ISSN: 2454-1907
DOI: 10.5281/zenodo.1443549

$$cost\ function = \ Mp + ts \tag{7}$$

Where (Mp) and (ts) are peak-overshoot and settling time respectively. Alternatively, any performance integral can also be used as cost function e.g. IAE (Integral Absolute Error), ISE (Integral Squared Error), MSE (Mean Square Error) ITAE (Integral Time Absolute Error) [2]. Table 3 shows the optimal PID parameters in the inner and the outer loops obtained using PSO algorithm. Figure 7 shows the step response of the system tuned by PSO algorithm.

Table 3: The PID parameters of the system tuned by PSO algorithm

Inner Loop			**Outer Loop**		
K_p	K_i	K_d	K_p	K_i	K_d
10	1	9.83	2.62	1	10

4.2. PID Tuning with Trial and Error Method

Trial and error tuning method is used to determine the parameters of a PID controller by inspection the dynamic behaviour of the controlled process output. It is very important to understand the effects of the tuning parameters on the behaviour of the process output for successful trial and error tuning [16]. By using the effect of increasing the PID parameters on a closed loop system in the table 1, we change the PID parameters until we find the good response.

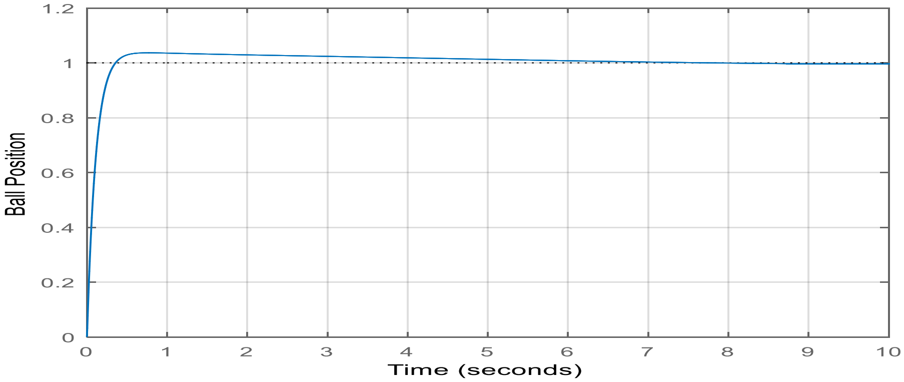

Figure 7: The step response of the system tuned by PSO algorithm

Table 4 shows some of trials for finding the parameters of the PID controller. Figure 8 shows the step response of the system tuned by trial and error. It is clear that the best PID parameters are obtained by "Trial 5" as shown in figure 8.

Table 4: The PID parameters tuned by Trial and Error method

	Inner loop			Outer loop		
	K_p	K_i	K_d	K_p	K_i	K_d
Trail 1	1	0	0	1	0	0
Trial 2	2	1	0	1	1	1
Trial 3	4	1	1	2	1	2
Trial 4	6	2	1	2	1	4
Trial 5	8	3	1	1	1	9

ISSN: 2454-1907
DOI: 10.5281/zenodo.1443549

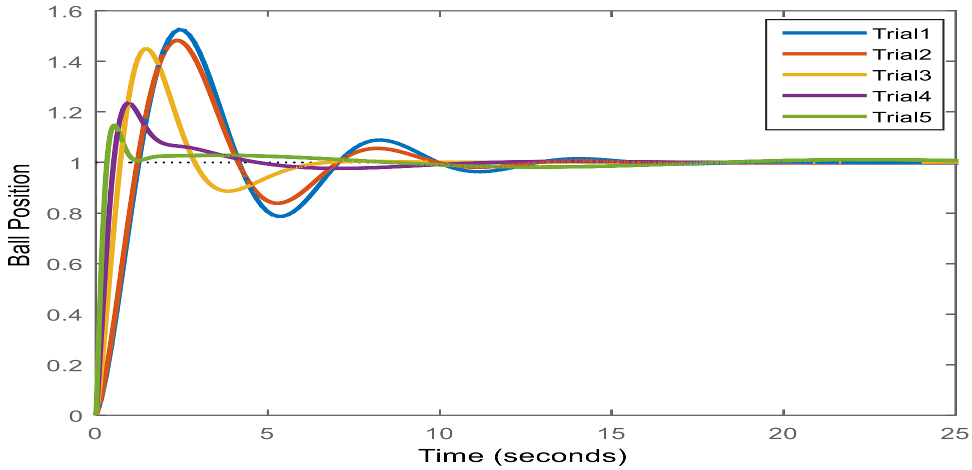

Figure 8: The step response of the system with various trials

5. Results and Discussion

Table 5 and figure 9 summarize the performances of the system obtained with PID controller under two different tuning techniques. It can be seen that the PSO-tuned method had better performance for ball and beam system compared to the conventional tuning method, which was trial and error method. It is clear that the PSO method gives promising results better than the Trial and Error method; the PSO gave the settling time of 3.7 seconds and rising time of 0.209 seconds compared to Trial and Error, which gave, 5.88 seconds and 0.235 seconds respectively. The transient response has 3.67 % overshoot for PSO method and 14.6 % overshoot for Trial and Error method. For steady state error (ess) Trial and Error and PSO, methods are zero.

Table 5: The performance of the system based on classic method and PSO algorithm

	Trial and Error method	PSO algorithm
Ts (s)	5.88	3.7
OS (%)	14.6	3.67
Tr (s)	0.235	0.209
Tp (s)	0.534	0.77
SSE	0	0

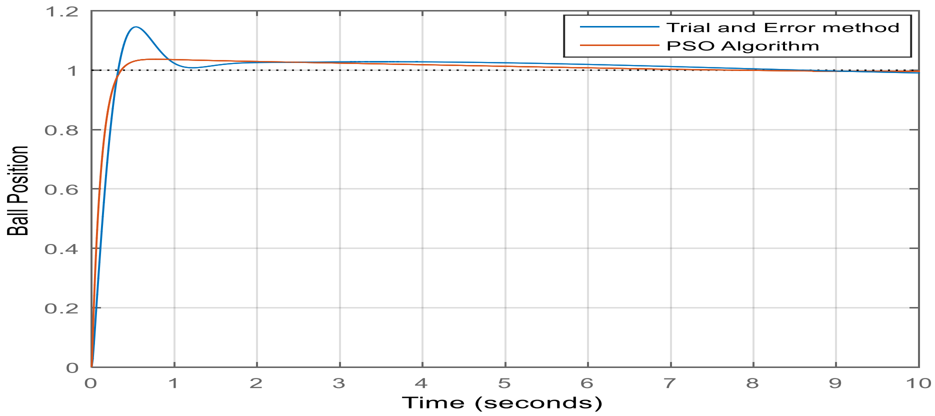

Figure 9: The step response of the system based on classic method and PSO algorithm

6. Conclusion

In this paper, the mathematical model for a ball and beam system has been derived successfully. The system consists of three main components which are servomotor model, angle conversion gain, and ball on the beam dynamic equation. Both servomotor and ball beam dynamics have a second order transfer function. It is quite tedious to design the fourth order system, thus for conventional method, two controllers have been implemented to control those second order components. The Particle Swarm Optimization (PSO) algorithm has been implemented successfully as a useful tool for optimizing the parameters of the PID controller. The efficiency of Particle Swarm Optimization algorithm for tuning the parameters of the PID controller has been compared with a classic method, Trial and Error method. From the analysis, PID-tuned by PSO shown a better performance and successfully reduce the values of Ts, Tr, OS and SSE than conventional method. The dynamic performance of the system controlled by Trial and Error method is relatively low and time consuming in comparison to the Particle Swarm Optimization method. The PSO method gives best performance and smooth behavior.

References

[1] Peter E. Wellstead, “Introduction to Physical System Modelling”, Academic Press Limited, pp. 221-227, 2000.
[2] M. A. Rana, Z. Usman, Z. Shareef, “Automatic Control of Ball and Beam System Using Particle Swarm Optimization,” CINTI 2011 • 12th IEEE International Symposium on Computational Intelligence and Informatics, pp. 529-534, 2011.
[3] Wei Wang, “Control of a Ball and Beam System,” M. Sc. Thesis, University of Adelaide, AUSTRALIA, June 2007.
[4] M. F. Rahmat, H. Wahid, and N. A. Wahab, “Application of intelligent controller in a ball and beam control system,” International journal on smart sensing and intelligent systems, vol. 3, pp. 45-60, 2010.
[5] Ziegler J. G. et al, “Optimum settings for automatic controllers”, Transactions of ACME, 1942, vol.64, pp. 759-768.
[6] Astrom K. J. and Hagglund T., “Automatic tuning of simple regulators with specifications on phase and amplitude margins”, Automatica, 1984, vol. 20, pp. 645-651.
[7] D’Azzo J. J. and Houpis C.H., “Linear control system analysis and design: conventional and modern”, McGraw-Hill Series in Electrical and Computer Engineering, New York 1995, 4th edition.
[8] Astrom K. J. and Hagglund T., “PID Controllers: Theory, Design, and Tuning”, Instrument Society of America, 1995, 2nd ed., pp. 134-229.
[9] Zang H., Zhang S. and Hapeshi K., “A review of nature-inspired algorithms”, Journal of Bionic Engineering, September 2010, vol. 7, Supplement 1, pp. S232-S237.
[10] Q. Bai, “Analysis of Particle Swarm Optimization Algorithm”, Computer and Information Science, vol 3 (1), February 2010, pp. 180-184.
[11] M. Rahmani, “Particle swarm optimization of artificial neural networks for autonomous robots,” M. Sc. Thesis, Chalmers University of Technology, SWEDEN, 2008.
[12] J. Im, J. Park, “Stochastic structural optimization using particle swarm optimization, surrogate models and Bayesian statistics,” Chinese Journal of Aeronautics, pp. 112-121, 2013.
[13] W. H. Lim and N. A. Mat, “Particle swarm optimization with increasing topology connectivity,” Engineering Applications of Artificial Intelligence, pp. 80–102, 2014.
[14] D. Tian and N. Li, “Fuzzy Particle Swarm Optimization Algorithm,” International Joint Conference on Artificial Intelligence, pp. 263-267, 2009.

ISSN: 2454-1907
DOI: 10.5281/zenodo.1443549

[15] M. Amjad, Kashif M.I., S. S Abdullah and Z.Shareef, “A Simplified Intelligent Controller for Ball and Beam System,” International Conference on Education Technology and Computer (ICETC), vol. 3, pp. 494-498, 2010.

*Corresponding author.
E-mail address: albagoul@ yahoo.com