SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I3P113

Volume 11 Issue 3, 121-129, March 2024

Original Article

Comparative Analysis of Controller Strategies for Steerable Robotic Bar System

Kiwon Yeom[1]

[1]*Department of Human Intelligence and Robot Engineering, Sangmyung University, Cheonan-si, South Korea.*

[1]*Corresponding Author : pragman@naver.com*

Received: 10 January 2024 Revised: 16 February 2024 Accepted: 14 March 2024 Published: 31 March 2024

Abstract - *This paper presents a comparative analysis of several controllers which are frequently used in robotic systems. This research especially focuses on controlling the ball and beam system of which the characteristics are non-linear and unstable. This paper first derives the dynamics model to control the ball on the beam for balancing. Secondly, it introduces automatic tuning methods for the PID controller based on Particle Swarm Optimization (PSO) and Sliding Mode Control (SMC). This paper introduces a cascade PID control architecture, which is designed with inner and outer feedback loops for balancing the ball on the beam. Finally, the insight of the simulation results with the conclusion is described through in terms of the quantitative comparison with the traditional controllers such as Integral of Time multiply Absolute Error (ITAE) and a Fuzzy-Logic Controller (FLC).*

Keywords - *Bio-inspired control, Particle Swarm Optimization, Ball and beam control, Balancing control, Position control.*

1. Introduction

Most autonomous systems, such as medical robot systems, might have considerable nonlinearity and uncertainty. Designing a controller in most nonlinear systems is a very tedious and time-consuming task since it is difficult to predict the behavior of the system due to the non-linear characteristic [1]. For example, in medical robots, it is very important to inject the needle into the accurate location of the target vein. Although the human arm is locked, however, the vein is possibly and continually moves when the needle is inserted into the skin [2].

In this aspect, locating (or balancing) the robotic bar with the needle on the specific position under unstable situations is very important to achieve accurate venipuncture. Balancing a ball on a beam (or ball and beam), which is well known and popular control system, is simple but complex since it includes highly non-linear features.

As shown in Figure 1, the purpose of the system is to move the ball and stand it in a specific position by swinging the beam, which is connected to a driving motor. This ball and beam system provides a challenge to design the controller which satisfies the nonlinear behavior characteristics. In this system, the ball can move back and forth on the beam by the rotation of the lever arm, which is connected to the output gear from the DC motor. The velocity (or acceleration) of the ball depends on only the torque of the DC motor which affects the velocity of the lever arm movement. In other words, the motor torque that rotates the beam is the only possible control parameter, and the appropriate control value, such as torque, acceleration, and velocity to generate the desired beam angle can be derived from the signal of the DC motor.

As the ball moves on the beam back and forth by rotations of the lever, the displacement between the ball and the target position is gradually decreased and resultants the ball being stabilized in the given location (means that the ball is placed in the specific position without oscillation).

Many researches have suggested various optimization algorithms to solve and result in stabilizing the ball on the beam [3]. However, most suggested methods ignored the nonlinearity of the ball and beam system and used state-space models or transfer functions. Of course, such applications made the stabilization of the ball on the beam under a small angle rotation of the lever arm, and the initial state of the beam should be horizontal (which means the angle of the beam is almost 0°).

On the contrary, most control algorithms failed to balance the ball on the beam as the rotation angle of the beam is larger than a certain degree, or the initial angle of the beam is larger than a certain degree. Therefore, the nonlinearity feature becomes significant, and the system cannot reach the static balance [4]. In this case, it is possible to use non-linear control techniques for better performance. For the last decade, various researchers introduced new approaches to stabilize the rolling

ball on the beam using a Neural Network Controller (NNC), Slide Mode Control (SMC), non-linear controllers using a Fuzzy Logic Controller (FLC), Genetic Controller (GC), or some hybrid controllers such as FLC-GC, FLC-NNC, etc [5]. Nevertheless, since the methods are very complicated and case-specific, it is still limited and difficult to apply for the real application.

In general, Proportional-Integral-Derivatives controller, called PID controller, has provided a good performance for non-linear systems [6]. However, it is a very tedious and time-consuming task to tune the controller gains to achieve an appropriate or optimal response from the system when an input value is given [7].

Ziegler-Nichol's PID gain tuning method is the most famous method [8]. Furthermore, the Integral of Time multiply Absolute Error (ITAE) criterion is frequently used for PID tuning. Nowadays, various unconventional approaches for automatic gain tuning which reported reasonable performance have been introduced. Especially an evolutionary approach using a genetic algorithm or genetic programming is one of the unconventional automatic tuning algorithms [9].

In addition, a popular biological optimization algorithm is Particle Swarm Optimization (PSO) which is frequently applied to solve the optimization problem in nonlinear systems. The PSO is inspired by the social behavior of birds or fish schools. The swarm behavior can be observed when the individuals comprise a flock that adheres to the nearest - neighbor, keeping a certain distance matching velocity and acceleration [10]. By virtue of the characteristics of movement with velocity and acceleration to make a swarm, the PSO can be considered as a feasible optimization method to control the rolling ball on the beam which depends on the velocity of the ball and acceleration of the bar motion.

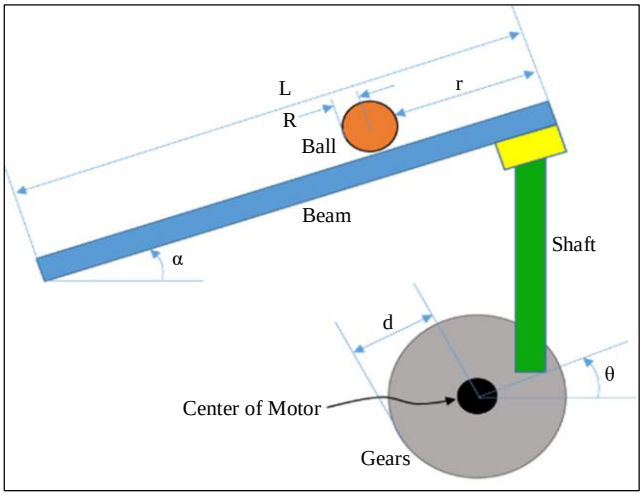


Fig. 1 System architecture of the ball and beam system

This paper investigates a PSO-based automatic PID gain tuning approach for balancing the ball on the beam based on a PID controller with a cascade feedback architecture. In addition, this paper introduces the mathematical ball and beam system model and an analytic design of the controller to balance the ball on the beam. Furthermore, this research shows the simulation results of the efficiency of the PSO tuning algorithm by quantitatively comparison to SMC and FLC-based automatic tuning methods.

2. Brief Literature Review of Particle Swarm Optimization

The iterative optimization process based on particle swarms called PSO uses the behavioral characteristics of birds, bees, or fish schools. Particle swarms were first devised as a tool for function optimization in [11]. The algorithm starts with searching food by a swarm of birds which are randomly scattered in the problem space. During the exploration to find food in the given space, each individual bird keeps the coordinate information. On each iteration, the minimum distance of an individual bird, called personal fitness value or personal best value, is recorded as pbest. The least value of pbests from the swarm is nominated as global best (gbest).

The optimization process updates the current velocity of each individual and adds the velocity information to the current position vector of individual birds, as described in Equation 1.

$$position_{new}[] = position_{cur}[] + velocity_{cur}[] \quad (1)$$

The velocity is correlated to the speed of the optimization or convergence which means that the swarm of the birds finally achieved the goal. In PSO, the direction and speed of convergence are related to the directional distance between pbest and gbest. That is these two factors influence the movement or behavior of the swarm. Therefore, information of pbest and gbest must be updated on each iteration. In addition, PSO uses weight values, called inertia and learning coefficients, for fast convergence in the velocity vector and the random movement with random numbers between 0 and 1. The velocity vector of the PSO can be described as follows.

$$\begin{aligned} velocity_{new}[] = {} & \varpi \times velocity_{old}[] \\ & + c_1 \times rand \times (pbest[] - positon[]) \\ & + c_2 \times rand \times (pbest[gbest] - position[]) \end{aligned} \quad (2)$$

Where c_1 and c_2 represents the learning factors and ϖ is the inertia factor.

3. Ball and Beam System

As shown in Figure 1, the hardware constituent involves a hollow ball, a beam (u-grooved bar), and a lever arm with a shaft connected to a geared motor.

The ball can freely roll on the beam along the entire u-grooved bar. The gear shaft of the DC motor is connected to the lever arm, which is linked to the end of the beam. The other end of the beam is coupled with the pivot joint so that the beam can be rotated.

Rotation of the motor makes an up-and-down motion of the lever arm. Since the one end of the lever arm is conjoined to the one end of the beam, the beam can be tilted whereby the ball can roll along the u-grooved bar. Therefore, the acceleration (or velocity) of the ball is proportional to the tilt angle only, which is the inclination between the horizontal plane and the beam.

The ball position on the beam can be measured using various sensor systems. For brevity in this paper, an ultrasonic sensor is attached to the end of the beam, which is coupled with the pivot joint, and it makes it possible for the sensor can continuously track the ball and acquire the distance from the sensor.

The main goal of the ball and beam system is to control the beam angle which is connected to the lever arm, so that the ball can be exactly placed on a given position of the beam. The materials and methods section should contain sufficient detail so that all procedures can be repeated. It may be divided into headed subsections if several methods are described.

4. Dynamics Model of the Rolling Ball

In this system, a ball like a thin spherical shell (i.e., Ping-Pong ball) is placed on the u-grooved bar of the beam and freely rolls back and forth. As mentioned before, the rotation of the motor by an angle makes the lever arm move up and down and accordingly tilts the beam by α as shown in Figure 2. The torque, which is the force that comes from the moment of the motor, changes the ball speed by acceleration which is related to the tilted angle of the beam. In other words, the bigger the tilt the beam makes, the faster the movement of the ball is.

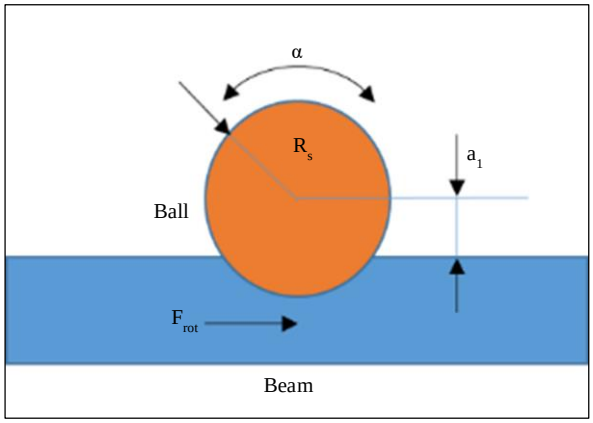


Fig. 2 Detail view of the inside of the ball and beam with u-grooved var

When the ball rolls on the bar, the force influencing the ball's motion comprises two different forces. Firstly, the ball is affected by the translational force, which is related to gravity as the beam is tilted. The other force is the rotational force, which comes from the rotation of the ball itself. Therefore, the net force should be balanced as follows.

$$\begin{aligned} F = m_{ball}a = m\ddot{x} &= \sum F_{ball} \\ &= F_{tran} - F_{rot} \\ &= m_{ball}g\sin\theta - F_{rot} \end{aligned} \quad (3)$$

Where m_{ball} is the mass of the ball, g is the gravity, θ is the angle of the beam from the horizontal position (i.e., the tilted angle of the beam), x is the moving distance between the current position and the previous position of the ball as the blever arm actuates the beam.

Equation 3 can be rewritten using directional derivatives as follows:

$$\sum F_{ball} = m_{ball}\ddot{x} + \eta\dot{x} \quad (4)$$

Where η is the coefficient of friction between the ball and the u-grooved bar.

In this research, although the ball on the u-grooved bar is slightly placed inside of the groove (see Figure 2), this paper assumes that when the angle of the lever arm is changed α, the moving distance of the ball is equal to the value that the angle of the lever arm and the rotational radius of the ball, d_{ctc} are multiplied as follows.

$$x = \alpha \cdot d_{ctc} \quad (5)$$

Where d_{ctc} is the distance between the center of the ball and the point of contract of the ball to the u-grooved bar.

As the ball moves on the bar, the torque generated by the rolling ball can be written as follows.

$$\sum \tau_{ball} = F_{rot} \cdot d_{ctc} = J_{ball}\ddot{\alpha} \quad (6)$$

Where $\ddot{\alpha}$ is the angular acceleration of the ball, J_b is the moment of inertia of the ball as follows.

$$J_{ball} = \frac{2}{5} m_{ball} R_{ball}^2 \quad (7)$$

Where R_{ball} is the radius of the ball.

The torque of the beam when the motor rotates can be defined as follows.

$$T_{bar} = T_{mot} + T_{ball} \quad (8)$$

Where T_{mot} and T_{ball} is the torque of the motor and the ball, respectively.

In general, the motor torque generated by the DC motor is known as follows.

$$T_{mot} = k_{mot} I_{mot} - J_{mot}\ddot{\theta} - d_{mot}\dot{\theta} \quad (9)$$

Where k_{mot} is the motor constant, I_{mot} is the current of the DC motor, J_{mot} is the moment of inertia generated by the rotor of the DC motor, d_{mot} is the damping constant of the DC motor, $\ddot{\theta}$ and $\dot{\theta}$ are the angular acceleration and the angular velocity of the beam, respectively.

T_{ball} can be described as follows,

$$T_{ball} = x m_{ball} g \cos\theta \quad (10)$$

Where *x* is the displacement of the ball from the reference (target) point or the center of the beam with a +/- sign with respect to the direction of the movement.

When the motor rotates, similar to Equation 7, two-moment factors occur in the ball and beam system. Since the rotation of the motor makes the rotation of the beam, which is connected to the pivot joint, the moment of inertia for the ball and beam system can be described as the sum of these two elements as follows.

$$J_{sum} = J_{mot} + J_{bar} \quad (11)$$

Where J_{mot} is the moment of inertia of the DC motor, J_{bar} is the moment of inertia of the bar (i.e., beam).

Equation 11 J_{mot} is generally known and J_{bar} can be modeled by introducing a simple assumption. This paper regards that the bar is a solid and rectangle form, J_{bar} is described thereunder as follows.

$$J_{bar} = \frac{1}{12} m_{bar} L_{bar}^2 \quad (12)$$

Where m_{bar} and L_{bar} are the mass of the bar and the length of the bar, respectively.

From Equations 9 to 12, the acceleration for deriving torque of the DC motor can be written as described in Equation 13.

$$\ddot{\theta} = \frac{k I_{mot} - x m_{ball} g \cos\theta - d_{mot}\dot{\theta}}{J_{sum}} \quad (13)$$

The Equation 13 expresses that the ball and beam system can be controlled by the angular acceleration and velocity which is transferred from the DC motor. Since the velocity of the DC motor is generally known at the manufacturing stage [12], the form can be described in terms of current, which is used to control the motor velocity as follows.

$$\dot{I}_{mot} = \frac{V_{mot} - R_{mot} I_{mot} - K_{bemf}\dot{\theta}}{L_{mot}} \quad (14)$$

Where L_{mot}, I_m, V_m, K_{bemf}, R_{mot} and are the induction, the electrical current, the input voltage, the back electromotive force by the rotation of the armature, and the reluctance, respectively.

In general, the induction of the motor is very small, it can be ignored and eliminated. For simplification, the induction can be ignored, and Equation 13 is described in Equation 14.

$$\ddot{\theta} = \frac{\frac{k}{R_{mot}} V_{mot} - x m_{ball} g \cos\theta - \left(\frac{k_{mot} K_{bemf}}{R_{mot}} - d_{mot}\right)\dot{\theta}}{J_{sum}} \quad (15)$$

Finally, the state space model for controlling the ball and beam system can be described in Equation 16.

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{g}{1+\frac{2}{5}(\frac{R_{ball}}{d_{ctc}})^2} & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{m_{ball} g}{J_{sum}} & 0 & 0 & -\frac{\frac{k_{mot} K_{bemf}}{R_{ball}} + d_{mot}}{J_{sum}} \end{bmatrix} \cdot \begin{bmatrix} x\cos\theta \\ \dot{x} \\ \sin\theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K_{bemf}}{R_{ball} J_{sum}} \end{bmatrix} V_{mot} \quad (16)$$

The left term of Equation 16 can be rewritten in terms of derivatives as described in Equation 17.

$$\dot{X} = [\dot{x}\ \ddot{x}\ \dot{\theta}\ \ddot{\theta}] \quad (17)$$

Where *X* is the state space vector.

The sensor data, which tracks the position of the ball, is used as a control signal and the controller regulates the voltage (or electrical current) which is supplied to the motor. The electric current generates the rotational torque of the DC motor, and the torque control is greatly related to the system performance.

In other words, the generated torque influences the ball to be placed on a position of the beam, and the displacement from the given target point can be gradually minimized by controlling the torque since the tilt angle of the beam only depends on the torque. This means the system output should be the displacement of the ball from the target point and the rotational angle of the beam or lever arm. Thus, the output of the system can be described in Equation 18.

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} X \quad (18)$$

The specifications of the ball and beam system are described in Table 1. Some parameters were determined from the data specification of the manufacturer (e.g., the DC motor). Otherwise, the values are directly observed and measured through pilot tests or warm-up trials of the system before the experiments. For example, the target position (x) to be located on the beam was established for the ball to converge in a few seconds due to the limit of the bar length. In addition, the control voltage (Vm) was not allowed to exceed 24V to protect the DC motor from an overvoltage or overcurrent release.

Table 1. System parameters

Parameter	Symbol	Unit	Value
Length of the Beam	L_{bm}	m	0.48
Mass of the Beam	M_{bm}	kg	0.02
Tilt Angle of the Beam	θ	rad	(-π/3, π/3)
Mass of the Ball	M_b	kg	0.025
Radius of the Ball	R_b	m	0.1
Control Voltage	V_m	Volts	(-10,10)
Ball Position	x	m	(-30,30)
Damping Ratio	d	Nm/(rad/s)	0.80
Friction Constant	η	Ns/m	0.001
Acceleration of Gravity	g	m/s^2	9.8
DC Motor Resistance	R_m	Ω	3.2
Constant of Electromotive Force	k	Nm/A	3.5
Electrical Inductance	L_m	H	0.006
Moment of Inertia	J_{sum}	Kgm^2	0.09
Moment of Inertia of the Beam	J_{bm}	Kgm^2	0.02
Moment of Inertia of the Motor	J_m	Kgm^2	0.025
Rotational Radius of the Ball	a_1	m	0.005
Motor Constant	K_{bemf}	Volts/(rad/s)	3.5

This paper also applies the small value simplification of some parameters which have a small amount of change, such as friction and electrical induction, to reduce the computational footage and increase the performance.

5. Cascade Controller Model with PSO

This research designs a cascade-based PID controller in which two PID control blocks are consecutively connected to control the ball and beam system. The one is used to generate an optimal electrical current (or voltage) to rotate the DC motor with respect to the input value (i.e., the current distance of the ball from the target position, which is calculated using the distance data from the sensor and a given target position). The other one is used to control the ball position by the movement of the beam and the current state information of the ball, which is transferred from inner control (i.e., motor control) results.

By actuating the DC motor back and forth according to the distance information, the ball can be gradually moved to the desired position of the beam. Therefore, the controller has to generate a suitable rotational angle of the beam DC motor by appropriate electric current or voltage. Since the rotational angle (α) is correlated to the angle (θ) of the shaft, which rotates according to the output of the DC motor, the plant can be considered as a system with a consecutive PID controllers. For this reason, this paper devises two separate feedback control loops, as shown in Figure 3.

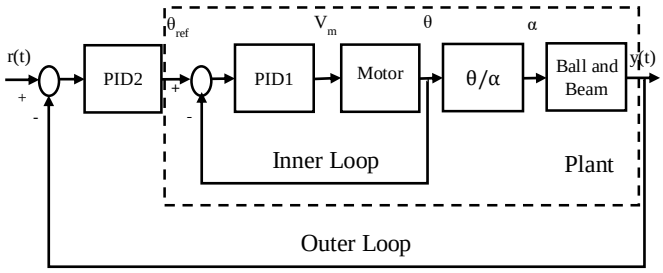


Fig. 3 The controller model which has two separated feedback loops for cascade connection of consecutive PID control

The PID1 controller of the inner loop is used to track the angle (α) by the reference signal, which is generated by the sensor. The PID2 controller of the outer loop is used to update the position information according to the output value (θ) from the inner loop and provides the control result to the inner control loop.

As a general performance evaluation guide of the controller, this paper sets four performance measurements for the proposed cascade controller such as rise time (T_r<1sec), settling time (T_s<2sec), percentage of overshoot ($R_{\%OS}$<0.5%), and steady-state error (S_e,1.5%). In addition, the values are chosen in terms of the fast response, convergence and effectiveness of the proposed controller, avoiding the ball veering off the beam at the end point.

The reference position in which the ball has to be steadily placed on the beam is analogous to the food location in the particle swarm. As known in [13], the small value of inertia factor (ω) can be greatly increased as the convergence time of the system might be longer and it can resultantly lead to the system failure. Due to this reason, this paper sets ω to 0.4 so that the system is able to reach the optimal solution within a relatively small number of iterations. In addition, the number of birds called the number of agents (i.e., individual agents) in PSO is established with 200 individuals through several warming-up trials. Lastly, to produce rapid convergence, this paper applies c1 and c2 are set to 1.2 and 1.4, respectively. More details of the parameter setting of PSO could be referred to in [13]. Therefore, the Equation 2 can be rewritten in Equation 19.

$$velocity_{new}[] = 0.4 \times velocity_{old}[] + 1.2 \times rand \times (pbest[] - positon[]) + 1.4 \times rand \times (pbest[gbest] - position[]) \quad (19)$$

Table 1 shows the regulation value of PID control parameters. In any event, the system is regulated not to exceed these values. The PSO algorithm starts with a random scattering of the particles within these limits.

Furthermore, the number of particles is managed not to trespass the limits during the whole optimization process. In this paper, the performance of the proposed controller is evaluated by the cost function (i.e., objective function) of the ball and beam system according to a matrix as follows.

$$w_1 \times R_{\%os} + w_2 \times T_r + w_3 \times T_s + w_4 \times S_e \quad (20)$$

Where $R_{\%os}$ is overshoot, T_r is rise-time, T_s is settling-time, and S_e is a steady-state error. w_1, w_2, w_3, and w_4 are corresponding weight values for each term, respectively.

Figure 4 shows the proposed PSO-based automatic PID tuning algorithm. This paper sets all weight values as one. However, these weight values can be changed according to the requirements of the control system or objectiveness aspect. For example, to minimize the rise time, w_2 should be one, and other values should be set to zero; w_2 could be a larger value than others, etc.

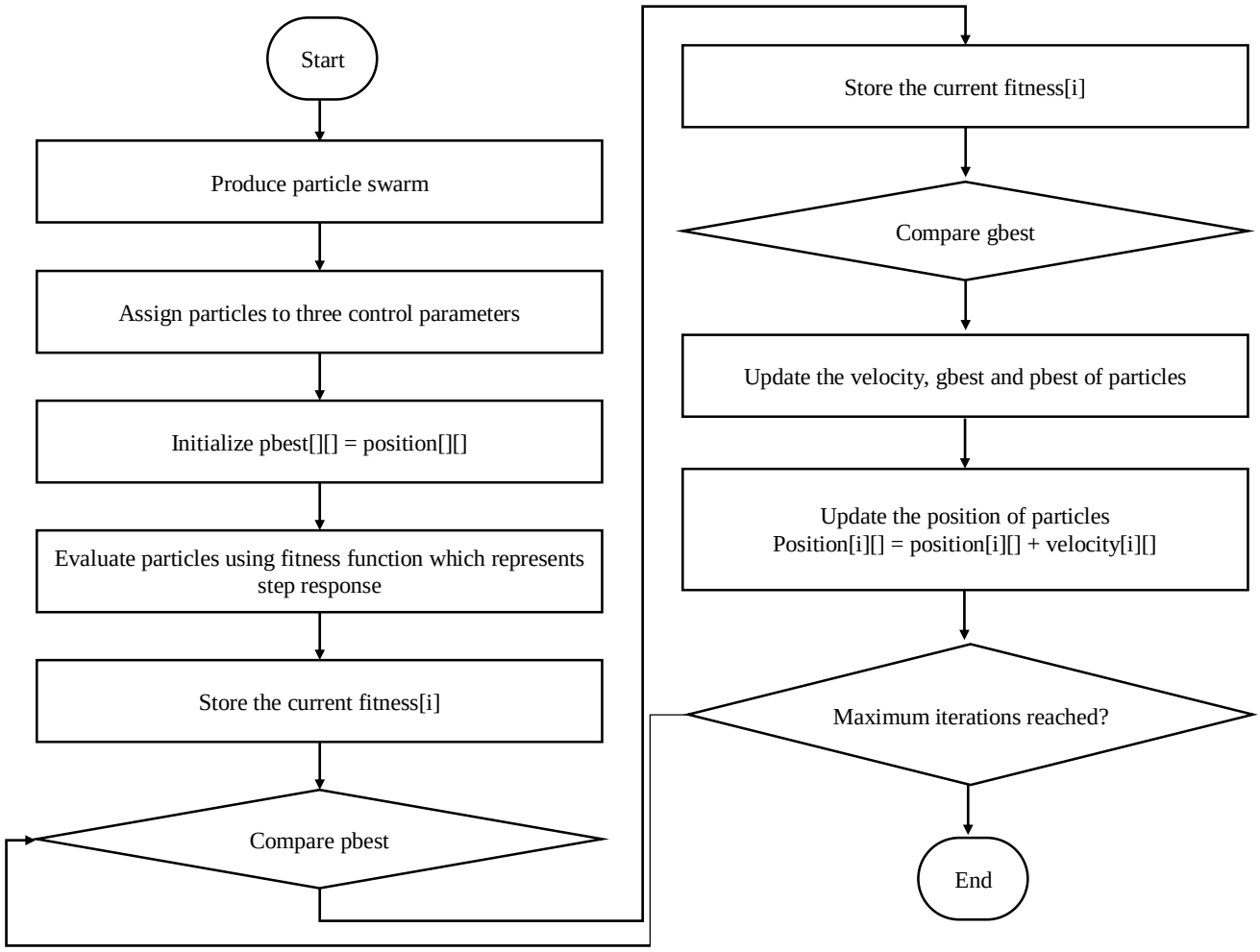


Fig. 4 PSO-PID control algorithm to automatically generate the PID parameters and control for the ball and beam system

Table 2. PID parameters were obtained for the 4 best PSO trials

	Outer-PID			Inner-PID		
	Kp	**Kd**	**Ki**	**Kp**	**Kd**	**Ki**
Trail 1	13.890	6.945	0.318	14.981	3.217	0.544
Trail 2	13.998	6.703	0.392	14.960	3.729	0.489
Trail 3	13.861	6.842	0.299	14.959	3.213	0.236
Trail 4	14.125	7.012	0.643	14.992	3.938	0.271

6. Experimental Results and Discussion

During the experiments, it was observed that the proposed algorithm can achieve global optimum within the given number of iterations. Figure 5 represents the performance comparison in terms of the ball position response, the beam angle response and the required voltage response, respectively. As shown in Figures 5(a) and 5(b), the results describe that the controllers can make the system stable within a certain time range, and PSO-PID controllers have lower settling time and overshoot than other controllers.

Although Figure 5(a) shows several cases in which the system would arrive at the local optimum, the phenomena were not observed over 35 iterations. Thereby, the proposed algorithm can converge and attain the global optimum after 40 iterations. From Figure 5(b) to 5(d), it is able to explain that the proposed algorithm can detect the optimal value for the PID controller with respect to the reference signal. In addition, the experimental results show that each PID parameter can be determined within around 30 iterations. As described before, each parameter value tends to converge after 30 iterations. In order to evaluate the effect of PSO, this paper performed the required voltages for three controllers and the results are shown in Figure 5(c). It tells the required voltage to actuate the DC motor and control the ball and beam system is fairly within the defined voltage range in the PSO method.

However, the required voltage for PSO-PID is higher than other optimization controllers. It is due to the initial swarm of particles for optimization at the very beginning stage. Nevertheless, the PSO-PID controller has the lowest steady-state error and rapid convergence, though the oscillation is detected at the very beginning stage to find the balance point. In other words, the proposed control strategy is more efficient than other controllers.

As shown in Figure 6, there is a minor difference between PSO-PID and SMC controllers, but the rising time can be attempted to trade off the steady-state error. The rising time of the SMC controller is a little bit better than PSO-PID, but the required time to convergence (steady-state) is longer than PSO-PID and fluctuates for a substantial time.

Since this paper considers the performance specifications as rising time, %overshoot, settling time and steady-state error, the proper controller can be chosen by the objectiveness. Therefore, these two algorithms might be options for tuning the PID parameters of the ball and beam system.

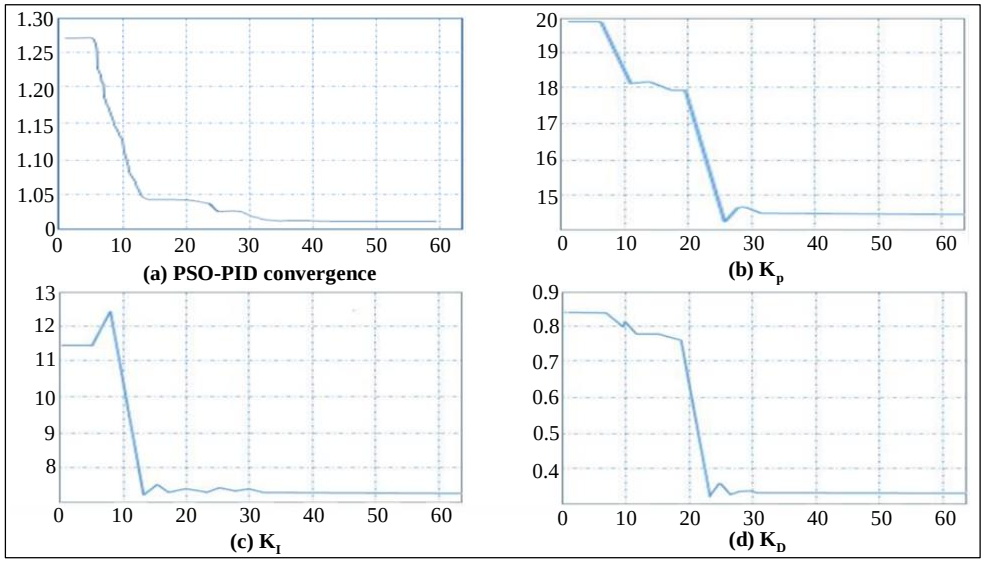
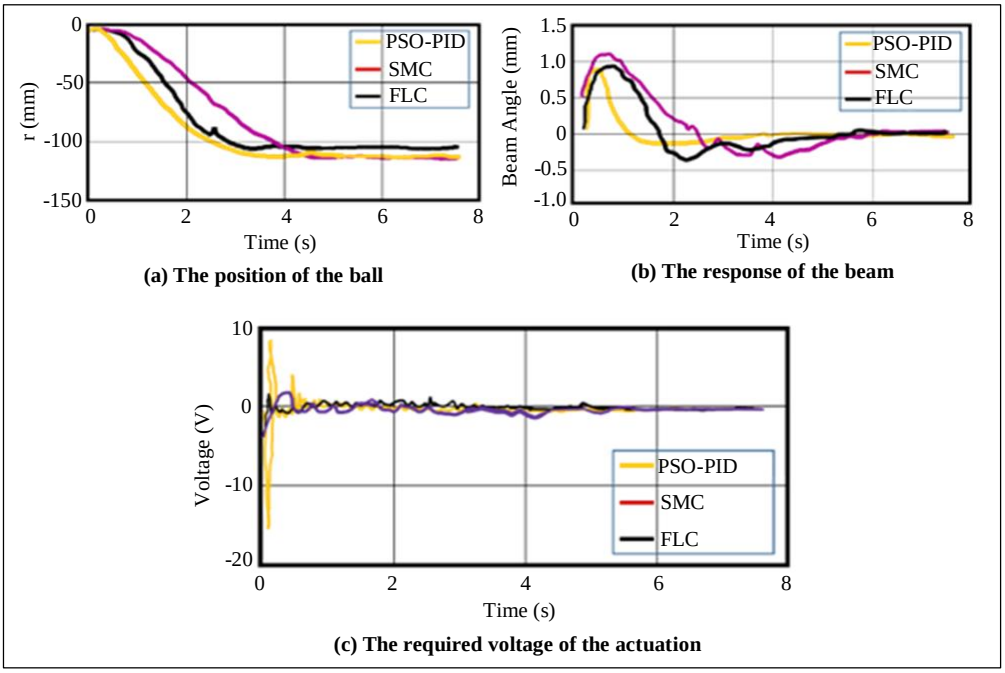


Fig. 5 Convergence of the proposed PSO-PID controller



(a) The position of the ball

(b) The response of the beam

(c) The required voltage of the actuation

Fig. 6 Performance comparison for PSO-PID with SMC and FLC

7. Conclusion and Future Works

This paper proposed the automatic PID tuning approach using Particle Swarm Optimization (PSO). For this, an analytical analysis to model the dynamic features of the ball and beam system was developed, and the performance of the proposed control was evaluated. Although the PSO-PID control algorithm showed that the controller could fall into a local minimum when the iterations were not enough, the proposed algorithm proved that the plant could successfully reach the global optimum and be stable quickly.

In addition, the comparison analysis with well-known control methods was performed in terms of rising time, percent overshoot, settling time, and steady-state error. Through the experiments and comparison, the PSO-PID showed better effectiveness than other controllers. This is due to the fact that the proposed algorithm can efficiently search the optimum value of the PID parameters of the ball and beam system using particle swarms. Furthermore, the proposed PSO-PID algorithm provided an easy-to-use automatic parameter tuning method to achieve non-trivial PID gain values without additional complex analysis of the target system or a bunch of trial and error.

As mentioned in the previous section, the difference between the SMC and PSO-PID controllers is apparent in terms of rising time and steady-state error. Therefore, it should be investigated which controller can be much more appropriate in specific conditions. In addition, it might be possible to investigate whether two controllers can be integrated and overcome the weakness with respect to rising time and steady-state error.

Funding Statement

This research was funded by a 2022 research Grant from Sangmyung University (2022-A000-0301).

References

[1] Mohamed Boukattaya, Neila Mezghani, and Tarak Damak, "Adaptive Nonsingular Fast Terminal Sliding-Mode Control for the Tracking Problem of Uncertain Dynamical Systems," *ISA Transaction*, vol. 77, pp. 1-19, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[2] Hamed Rahimi Nohooji, Ian Howard, and Cui Lei, "Neural Network Adaptive Control Design for Robot Manipulators under Velocity Constraints," *Journal of the Franklin Institute*, vol. 355, no. 2, pp. 693-713, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[3] Sung-Kwun Oh, Han-Jong Jang, and Witold Pedrycz, "The Design of a Fuzzy Cascade Controller for Ball and Beam System: A Study in Optimization with the Use of Parallel Genetic Algorithms," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 2, pp. 261-271, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[4] Vineet Kumar, and K.P.S. Rana, "Nonlinear Adaptive Fractional Order Fuzzy PID Control of a 2-Link Planar Rigid Manipulator with Payload," *Journal of the Franklin Institute*, vol. 354, no. 2, pp. 993-1022, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[5] Ye Cao, and Yong-Duan Song, "Adaptive PID-Like Fault-Tolerant Control for Robot Manipulators with Given Performance Specifications," *International Journal of Control*, vol. 93. no. 3, pp. 377-386, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Kangdi Lu et al., "Design of PID Controller Based on a Self-Adaptive State-Space Predictive Functional Control Using Extremal Optimization Method," *Journal of the Franklin Institute*, vol. 355, no. 5, pp. 2197-2220, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[7] P.T. Chan, W.F. Xie, and A.B. Rad, "Tuning of Fuzzy Controller for an Open-Loop Unstable System: A Genetic Approach," *Fuzzy Sets and Systems*, vol. 111, no. 2, pp. 137-152, 2000. [CrossRef] [Google Scholar] [Publisher Link]

[8] Nguyen Minh Tri, Dang Xuan Ba, and Kyoung Kwan Ahn, "A Gain-Adaptive Intelligent Nonlinear Control for an Electrohydraulic Rotary Actuator," *International Journal of Precision Engineering and Manufacturing*, vol. 19, no. 5, pp. 665-673. 2018. [CrossRef] [Google Scholar] [Publisher Link]

[9] Bo Zhu, Qingrui Zhang, and Hugh H.T. Liu, "Design and Experimental Evaluation of Robust Motion Synchronization Control for Multivehicle System without Velocity Measurements," *International Journal of Robust Nonlinear Control*, vol. 28, no. 17, pp. 5437-5463, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[10] Xinyu Zhang et al., "Improved UDE and LSO for a Class of Uncertain Second-Order Nonlinear Systems without Velocity Measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 7, pp. 4076-4092, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[11] Syed Ali Ajwad et al., "Sliding Mode Control of Rigid-Link Anthropomorphic Robotic Arm," *2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI)*, Rawalpindi, Pakistan, pp. 75-80, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[12] Shrey Kasera, Amit Kumar, and Lal Bahadur Prasad, "Trajectory Tracking of 3-DOF Industrial Robot Manipulator by Sliding Mode Control," *2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics*, Mathura, India, pp. 364-369, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[13] Roshni Maiti, Kaushik Das Sharma, and and Gautam Sarkar, "PSO Based Parameter Estimation and PID Controller Tuning for 2-DOF Nonlinear Twin Rotor MIMO System," *International Journal of Automation and Control*, vol. 12, no. 4, pp. 582-609, 2018. [CrossRef] [Google Scholar] [Publisher Link]