Reinforcement learning control of servo actuated centrally pivoted ball on a beam

Archana Ganesh
Dept. of Instrumentation and Control
National Institute of Technology
Tiruchirappalli, India
archana.2098@gmail.com

Banu Sundareswari M.
Dept. of Instrumentation and Control
National Institute of Technology
Tiruchirappalli, India
monalisapanda236@gmail.com

Monalisa Panda
Dept. of Instrumentation and Control
National Institute of Technology
Tiruchirappalli, India
jjbanu78@gmail.com

Then Mozhi G.
Dept. of Instrumentation and Control
National Institute of Technology
Tiruchirappalli, India
thenmozhi.gp@gmail.com

Dhanalakshmi K., *IEEE Senior Member*
Dept. of Instrumentation and Control
National Institute of Technology
Tiruchirappalli, India
dhanlak@nitt.edu

Abstract— **The objective of this work is to devise a controller using Reinforcement Learning (RL) agents, for unstable and complex control systems like the ball beam system. The reinforcement learning agent's job is to keep the ball's position as close as possible to a set point. The Reinforcement Learning agent learns through rewards. Every action is taken such that the reward value is maximized. The reward becomes maximum if setpoint and the current ball position are as close as possible. So, a ball position from the sensor, in terms of reward is taken as feedback to predict the next action. The predicted action is the angle of the beam which needs to be turned by the motor. The action space considered is of a continuous domain, and the Reinforcement Learning algorithms that have been used are Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG). Once the environment dynamics are defined, hyper-parameters of the reinforcement learning algorithms pertaining to this environment are tuned, and the model is trained. Servo motor is used as the actuation mechanism.**

Key words-- Reinforcement Learning, Ball and beam system, servo motor, Proximal Policy Optimization, Deep Deterministic Policy Gradient.

I. INTRODUCTION

Deep learning or machine learning have recently received more attention in the research community and has become a highly successful and widespread research topic due to their outstanding performance in many nonlinear problems and applications. Deep learning is the process of nonlinear mapping between the inputs and outputs along with the extraction of features of data using deep neural networks with many hidden layers [1]. Reinforcement learning (RL) is one of the subclasses of machine learning which learns through communication with an environment by taking different actions and experiencing many failures and successes while trying to maximize the received rewards [2]. The learning system affects its future inputs and hence they are closed-loop problems. The algorithms used in RL includes Q learning which has a lookup table format of state and actions; trust region policy optimization (TRPO) complex to implement; Proximal Policy Optimization (PPO) improved version of TRPO which is simple and easy to implement with the discrete state; and Deep Deterministic Policy Gradient (DDPG) which improves the learning rate in a straight actor-critic approach with the continuous state. The application of RL includes many areas like gaming, Speech and image processing, robotics, and industrial automation. In robotic applications of RL, it aids the system with the self-learning behavior to form an adaptive control. One of the robotic manipulators called the ball balancing system which has nonlinear dynamics is chosen as a real-time problem to implement the RL control in a simulation.

The ball and beam system can be actuated with conventional servo motor and non-conventional smart material-based actuators [9-10]. Various control strategies have been explored to stabilize the ball and beam system. The dynamics of the ball on beam system is complex and nonlinear and hence the beam angle and ball position are controlled by the non-model based control strategies such as Neural Network [11], and Fuzzy Logic [12]. This strategy applies even when there is a variation in the parameter, and it need not necessitate the mathematical equations. Based on the control input, the model-based control strategies for the system fall into two categories; one approach uses the torque of the beam; the other uses the excitation of the motor as the regulated signals of the PID controller. State-space, compensator, and Linear Quadratic Regulator (LQR) to control the underactuated ball on beam system are an example of model-based approach in which torque of the beam acts as the input of the system. Various drawbacks in the usual control methods like system constraints (PID), performance- robustness tradeoff (FLC) [8], can be addressed using an unsupervised learning alternative. The reinforcement learning algorithm for control can prove to be much more efficient in terms of robustness, rate of convergence, and adaptability without compromising on the performance. The actor-critic neural control called Reinforcement Learning along with PD control has been designed for underactuated ball and beam system [3] and used with a deep deterministic policy gradient in [4] to stabilize the ball position. The authors [6] derived adjustable policy learning rate (APLR) for the cart-pole inverted pendulum which integrated concept of fuzzy Inference system (FIS) and Lyapunov stability provides the wide range of reward system that promotes the learning stability of the proximal policy optimization (PPO) algorithm [6]. The focus of this work is to implement Reinforcement Learning algorithms: PPO and DDPG to achieve the control objective. The system dynamics of the underactuated, nonlinear ball and beam system were simulated, and the algorithms were used to guide the ball to the required set point.

II. System Description

The ball and beam system is a nonlinear, underactuated two-degree system driven by a single actuator that controls the position of the beam which rotates its central axis, and position of the ball rolling along the beam. The objective of the system is to position the beam and hence the ball to a set point.

A. *Experimental Setup*

The ball balancing system is actuated by the servo motor MG995 which is a DC Motor with low torque and high speed. It also has a metal gear system to vary the speed, a position sensor to measure the angular position of the shaft and a control circuit for closed-loop control. The metal gear is externally connected with a long shaft to carry the polycarbonate beam which is supported in the middle of the two vertical structures and rotates against its central axis. The position of the beam is proportional to the position of the shaft can be obtained from the position sensor. The beam is connected with two resistive wires inside and makes a track to allow the metal ball rolling along with it. This configuration acts as a linear potentiometer to measure the ball position. The positions of the beam and the ball are given to the control circuit. The position control beam is obtained by the signal from the control circuit that controls the direction of rotation of the DC motor.

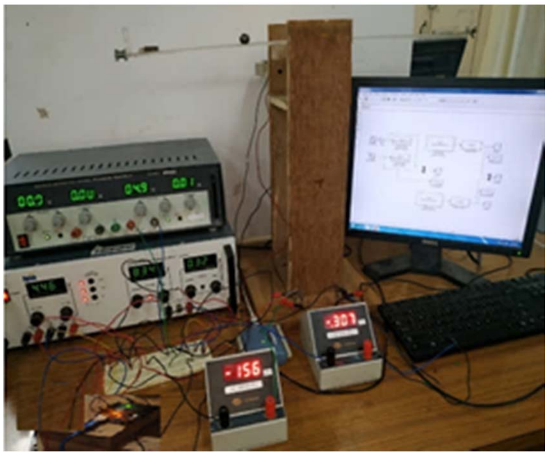


Fig. 1. Experimental arrangement of the ball and beam

TABLE I. SPECIFICATIONS OF THE BEAM AND THE BALL

Parameter	Symbol	Unit	Value
Beam tilt angle	θ	rad	(-π/6, π/6)
Mass of the ball	M_{ball}	kg	0.023
Radius of the ball	R_b	m	0.008
Control Voltage	V	V	(0, 5)
Position of ball	X	m	(-0.5, 0.5)
Acceleration of gravity	G	m/s^2	9.81
Length of the beam	L_{beam}	M	1
Mass of the beam	M_{beam}	kg	0.103

The PWM signal from the Arduino acts as an activation signal for the servo motor, i.e., the on-time of the pulse varies the angle of rotation of the motor. The sensor signals from linear potentiometer along beam which provides the ball position and position sensor of servo motor which provides the beam angle are feedback to the Arduino and then to the reinforcement learning control architecture in PYTHON program environment to calculate the PWM control signal for the required beam angle to the servo for positioning the beam intern the ball at the pre-set value. Fig. 1 shows the total experimental arrangement for the system.

B. *Mathematical Modelling*

To implement the control in any system it is necessary to derive the relation between the input and outputs of the system which is the mathematical model of the system. Basically, it is the differential equation and the elements of it are obtained by Newton's second law to balance the force/torque acting on the system.

Torque on the beam = Torque of motor + torque on the ball

$$T_{ball} = -xM_b g\cos\theta \tag{1}$$

$$T_{motor} = KI - J_m \frac{d^2x}{dt^2} - \mathrm{b}\frac{\mathrm{dx}}{\mathrm{dt}} \tag{2}$$

$$\frac{d^2x}{dt^2} = \frac{KI - xM_b g\cos\theta - b\frac{dx}{dt}}{Jm} \tag{3}$$

where $\frac{d^2x}{dt^2}$ is the angular acceleration of the ball, K is emf coefficient, I is current, θ is the angle of the beam from the reference position, M_b is the mass of the ball, g is acceleration of gravity, x is the distance travelled by the ball, $\frac{dx}{dt}$ is the velocity of the ball, J_m is the moment of inertia of the motor, b is the damping coefficient system. The specifications of the ball and beam system are listed in Table 1. The value of x is based on the length of the beam with 0 being the midpoint.

III. Reinforcement learning

Reinforcement Learning (RL) is a subclass of a machine learning technique that aids a software agent who is the decision-maker in a communicative environment with the help of responses of its self-learning to maximize the reward. [1]. It is also an algorithm to find the optimal reward for the observed state S through an action A. Optimal means the actions which yield the highest feedback called reinforcements (R) for the current action and also for the upcoming actions. Reinforcement learning aims to develop a control algorithm, called a policy that chooses optimal actions (A) for each observed state (S) based on reinforcement feedback to maximize the reward to achieve the objective. An important idea in reinforcement learning is establishing the value function which is the predictable sum of future reinforcement signals that the agent receives from the associated state of the environment. Thus V (s) is the value of state S and selecting optimal actions (A) in the future [2].

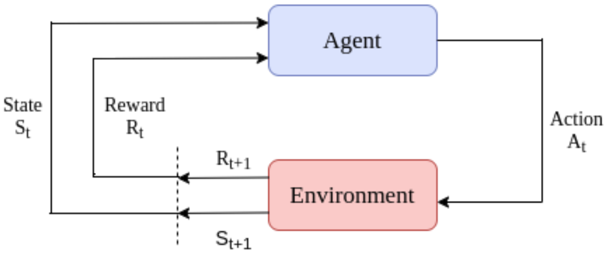


Fig. 2. Reinforcement learning control

Algorithms such as PPO, DDPG are continuous control based on ‘on-policy’ and ‘off-policy’ interaction with the environment, respectively. In this work, both policy algorithms are verified for positioning the ball on the beam. Fig. 2 shows the general schematic of RL control.

A. On policy and Off policy

An on policy uses the estimation method to calculate the value of the policy which is deterministic whereas the off-policy method generates the policy value by the behavior that can sample all the action space and it is not related to the on-policy value. Off policy, the learner learns the policy independently without any exploration, whereas in, on policy the policy learning is performed with the agent including exploration steps so that it can be iteratively improved by minimizing the loss.

B. Proximal Policy Optimization (PPO)

PPO is one of the most popular deep reinforcement uses on policy learning and the probability ratio between the two policies uses clipped probability ratios. This algorithm alters the objective function at every step when there is a difference in the present and past values of policy. PPO algorithm can also be implemented with GPU called PPO2 to increase the speed of operation by 3 times [4]. Fig. 4 shows the schematic of RL with the PPO algorithm.

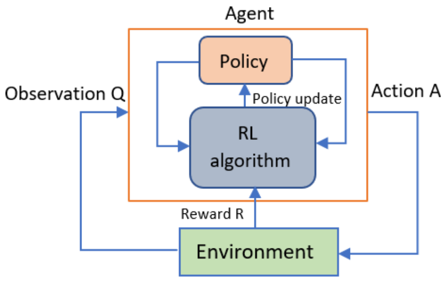


Fig. 3. Reinforcement learning with PPO

C. Deep deterministic Policy Gradient (DDPG)

Deep Deterministic Policy Gradient (DDPG) combines the advantage of the Deterministic Policy Gradient (DPG) of the model-free, off-policy algorithm, and Deep Q learning (DPQ). The advantage of DPG is an actor-critic for a system having a continuous action space environment. The advantage of DPQ is a replay (memory previous learning) and slow learning. In this algorithm, instead of a probabilistic method, the straight-forward approach is used, i.e., the actor directly maps the states to action. It has a target value network which is the delayed version of the original network to track the learning thereby improving the stability of the system. Fig. 4 shows the schematic of RL with the DDPG algorithm.

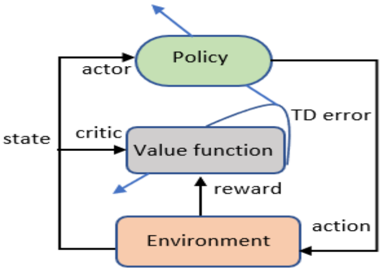


Fig. 4. Reinforcement learning with DDPG

This paper aims to justify the adaptability of the PPO and DDPG algorithm in reinforcement learning for servo actuation mechanism of ball and beam system. In this section, the implementation details such as defining the environment dynamics, simulating the environment, and establishing the reinforcement learning control, is discussed.

IV. Implementation

A. Renderer

For simulation purposes, OpenAI Gym, an open-source interface for reinforcement learning tasks, was utilized to render the Ball – Beam system and define the environment dynamics. It also comprises a set of APIs which are used to make the reinforcement learning task more computationally efficient. The sample coding and simulated results for ball and beam configuration in the OpenAI Gym environment are shown in Fig. 5 and Fig. 6, respectively.

Simulated system description

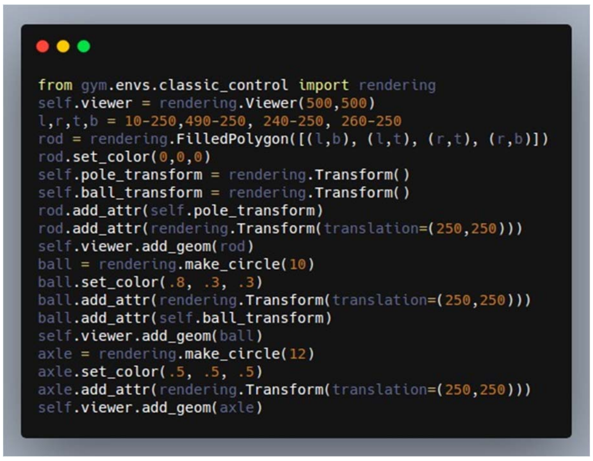
- This environment requires a continuous action space which is the set of angles in case of motor actuation.
- The total length of the beam is assumed to be 1.0m where 0.0 is the midpoint of the beam and -0.5 and +0.5 are the left and right extremes of the beam respectively.

The main objective of reinforcement learning is to maximize the reward points to keep the position of the ball at the desired value. The calculation of reward points is based on the reward functions are represented in Eq. 4

$$Reward = \frac{1-|Setpoint-ball\ position|^2}{Beam\ length^2} \quad (4)$$

B. RL control of ball position

The ball location/position on the beam is incorporated into the reward function and the reward is taken as a feedback input for taking an optimal action (voltage input/set angle) in the subsequent timesteps. The agent tries to maximize this reward function and takes the corresponding action for each timestep. Thus, the agent achieves the control objective of manipulating the ball position at the setpoint, by obtaining an optimal state, action, and reward trajectory for every action, which is the angle change and tries to maximize the reward by bringing ‘x’ near the setpoint. Once the environment dynamics are defined, reinforcement learning algorithms pertaining to this environment are implemented, and the model is trained.



```
from gym.envs.classic_control import rendering
self.viewer = rendering.Viewer(500,500)
l,r,t,b = 10-250,490-250, 240-250, 260-250
rod = rendering.FilledPolygon([(l,b), (l,t), (r,t), (r,b)])
rod.set_color(0,0,0)
self.pole_transform = rendering.Transform( )
self.ball_transform = rendering.Transform( )
rod.add_attr(self.pole_transform)
rod.add_attr(rendering.Transform(translation=(250,250)))
self.viewer.add_geom(rod)
ball = rendering.make_circle(10)
ball.set_color(.8, .3, .3)
ball.add_attr(rendering.Transform(translation=(250,250)))
ball.add_attr(self.ball_transform)
self.viewer.add_geom(ball)
axle = rendering.make_circle(12)
axle.set_color(.5, .5, .5)
axle.add_attr(rendering.Transform(translation=(250,250)))
self.viewer.add_geom(axle)
```

Fig. 5. Sample coding for ball and beam in OpenAI Gym

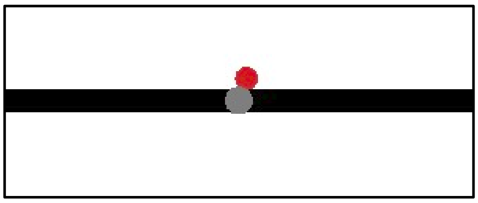


Fig. 6. Simulated ball on beam in OpenAI Gym

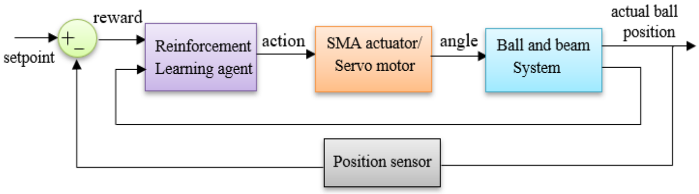


Fig. 7. Schematic flow diagram of the control loop.

The schematic diagram of closed control of ball and beam using RL control algorithm is shown in Fig. 7. The RL agent was trained using the PPO2 and DDPG algorithms with different learning rates, time steps, and epochs. The policy used was MlpPolicy that implements actor-critic, using an MLP (2 layers of 64). The trained model was able to successfully balance the ball near the required set point. Thus, the appropriate algorithm is selected, and the parameters are tuned accordingly to achieve satisfactory results. The hyperparameters of RL learning for PPO and DDPG algorithms are tabulated in Table II.

TABLE II. HYPERPARAMETERS OF LEARNING

Hyperparameters/ Algorithms	PPO	DDPG
Learning rate	0.00025	0.00001
Total time steps	30000	30000
Epochs	250	500
Average reward	0.93	0.94

V. RESULTS AND DISCUSSIONS

To visually infer the performance of the reinforcement learning control, the reward, assigned for the iterative actions taken by the agent, is considered for the evaluation. Furthermore, to verify the basic controller characteristics such as disturbance rejection, robustness, and adaptability, the position of the ball on the beam is taken as a plotting parameter.

The performance of the PPO algorithm for different setpoint positions (0.0, 0.3) of the ball is evaluated with corresponding reward point achievements are shown in Fig.s 8a, 8b, 9a, and 9b. Similarly, the performance of the DDPG algorithm with different set points (0.0, 0.3) of ball positions is obtained with the corresponding reward point achievement as shown in Figs 10a, 10b, 10c and 10d. From the results, it is observed that the DDPG algorithm performs well for positioning the ball at a given setpoint position without much oscillation.

The robustness of the PPO and DDPG algorithms are evaluated for different gain values of the system. The Figs. 11a and 11b show the response of robustness of PPO and DDPG algorithms, respectively. It is inferred that the DDPG algorithm is more robust than the PPO algorithm.

The disturbance rejection capability of the PPO and DDPG algorithms are evaluated by many applied signals of positive and negative directions of ball positions and disturbance with different magnitude at various time steps.

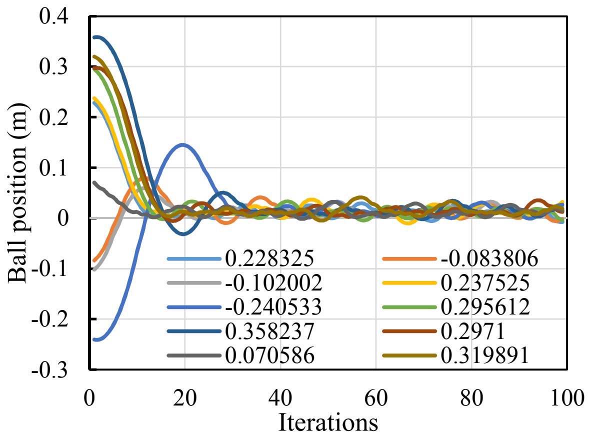


Fig. 8a. Ball position at different initial positions and set point = 0 using PPO algorithm

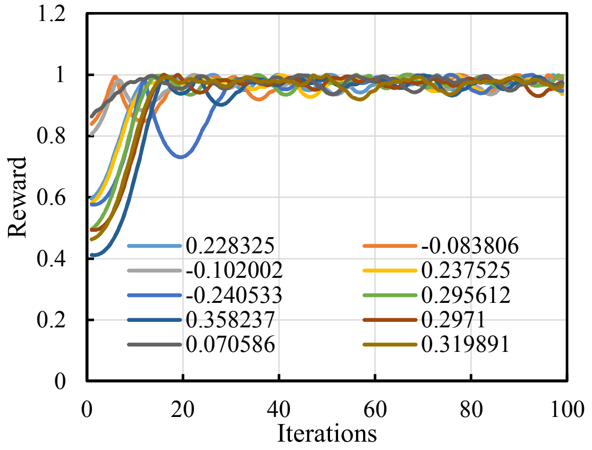


Fig. 8b. Reward achievement of PPO algorithm for set point = 0

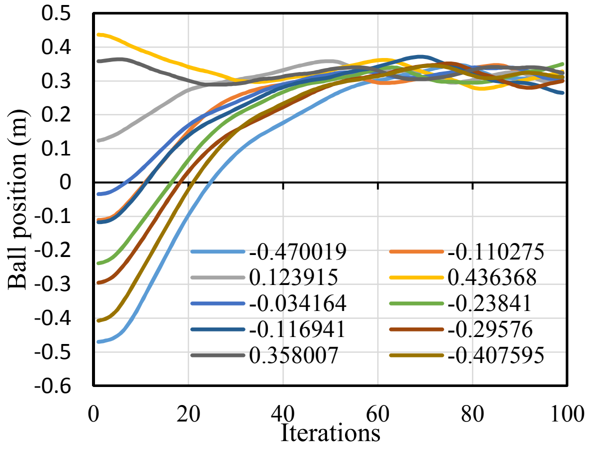


Fig. 9a. Ball position at different initial positions and set point = 0.3 using PPO algorithm

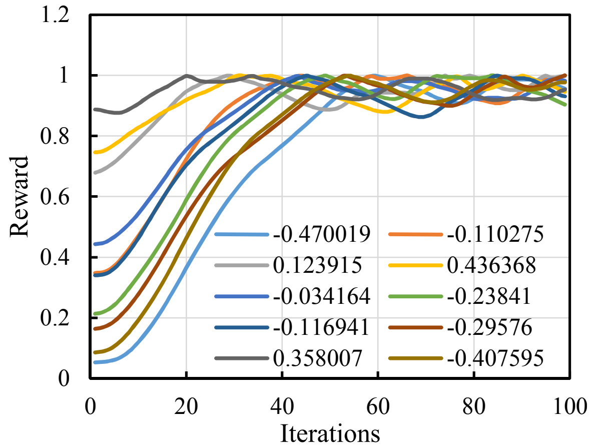


Fig. 9b. Reward achievement of PPO algorithm for set point = 0

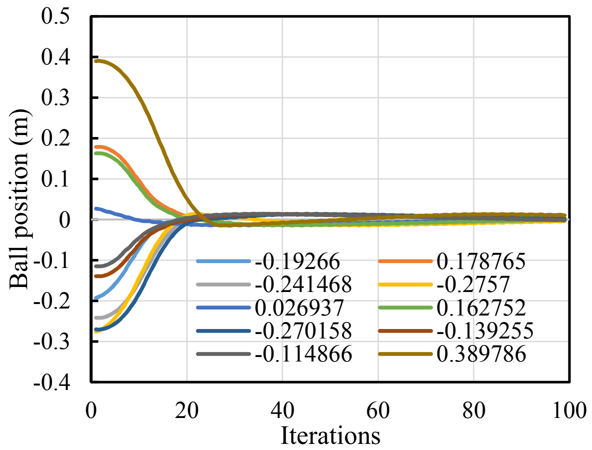


Fig. 10a. Ball position at different initial positions and set point = 0 using DDPG algorithm

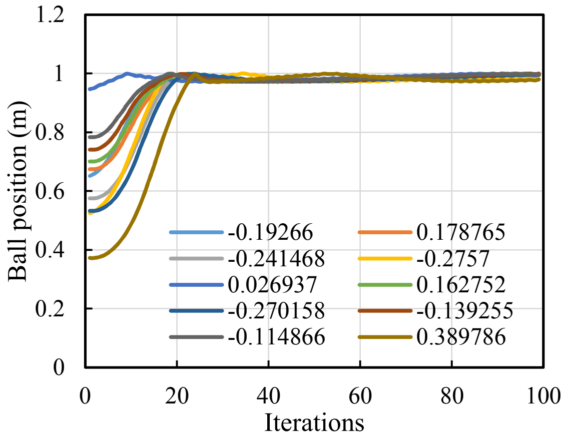


Fig. 10b. Reward achievement of DDPG algorithm for set point = 0

The Figs. 12a and 12b show the response of disturbance of PPO and DDPG algorithms, respectively. It is clearly visible that the DDPG algorithm responds well for rejecting the disturbance than the PPO algorithm.

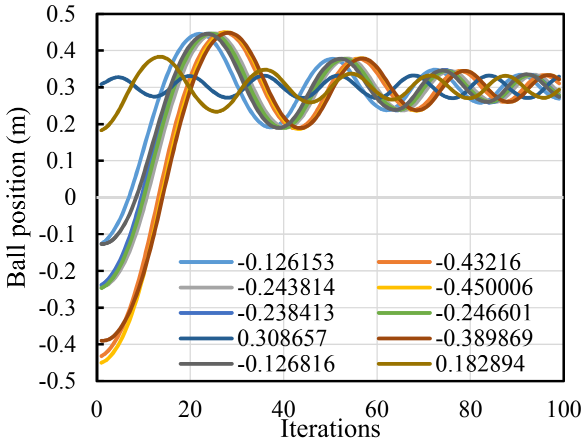


Figure 10c. Ball position at different initial positions and set point = 0.3 using DDPG algorithm

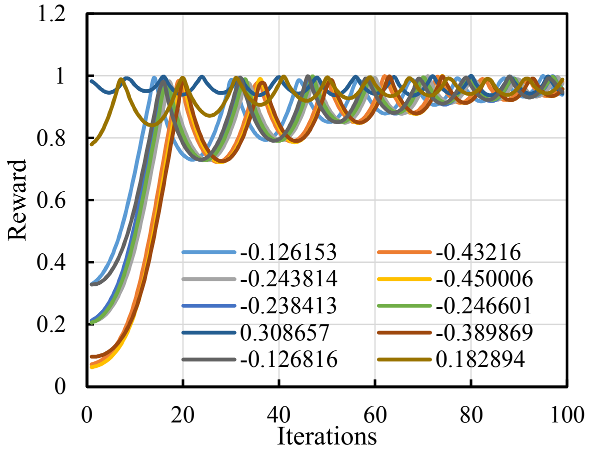


Fig. 10d. Reward achievement of DDPG algorithm for set point = 0.3

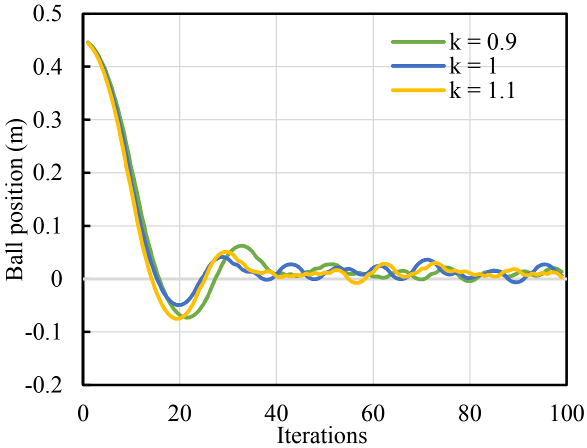


Fig. 11a. Robustness response of PPO algorithm

Finally, the performance of PPO and DDPG of RL are compared with conventional PID controller and depicted in Fig. 13. From the results, it is observed that the PPO of RL has more oscillations and PID controller response has small overshoot and undershoot whereas the DDPG algorithm outperforms the PPO and conventional PID controller.

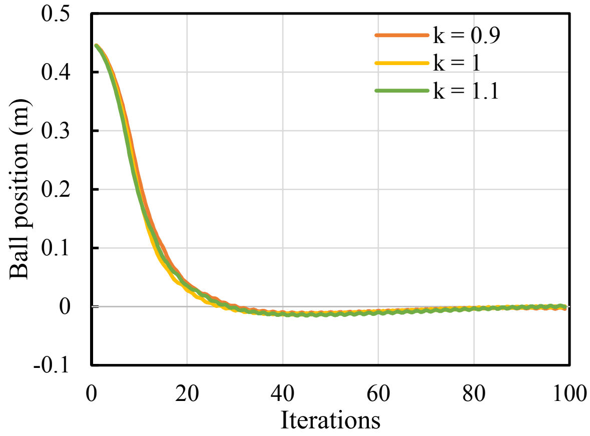


Fig. 11b. Robustness response of DDPG algorithm

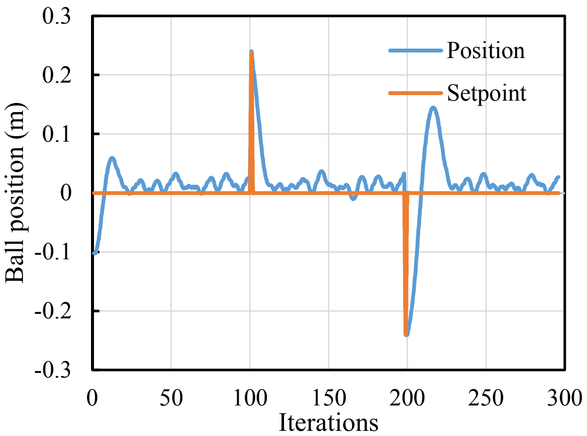


Fig. 12a. Disturbance Rejection response of PPO algorithm

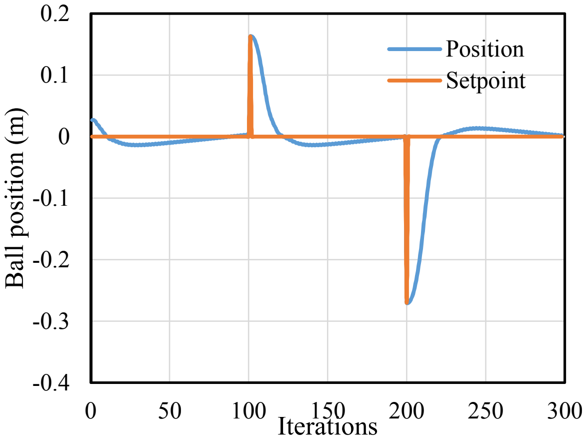


Fig. 12b. Disturbance Rejection response of DDPG algorithm

VI. Conclusion

The Reinforcement learning control for the dynamic ball and beam system is designed and simulated using RL control. The environment is developed in OpenAI Gym open source application to simulate and visualize the dynamics of the ball and beam and then the reinforcement learning agent learns the optimal policy to achieve the control objective. Proximal Policy Optimization (PPO), Deep Deterministic Policy Gradient (DDPG) algorithms were implemented, and its results compared. The reinforcement learning agent's capability of disturbance handling was verified by applying disturbance signals to the system. The initial position of the ball on the beam was also varied and the agent was observed to balance efficiently. From results it is observed that the performance of the DDPG algorithm is better than PPO algorithm and PID controller. The settling time was found to be approximately between 3 to 3.5 second while averaging over various episodes. The trained model proved to be an efficient alternative for conventional controllers like PID. The practical and computational advantages of the adaptability of Reinforcement Learning methods can be extended to balance several unstable complex systems. Accessibility to solutions for control objectives can be improved by considering Reinforcement Learning as an alternative.

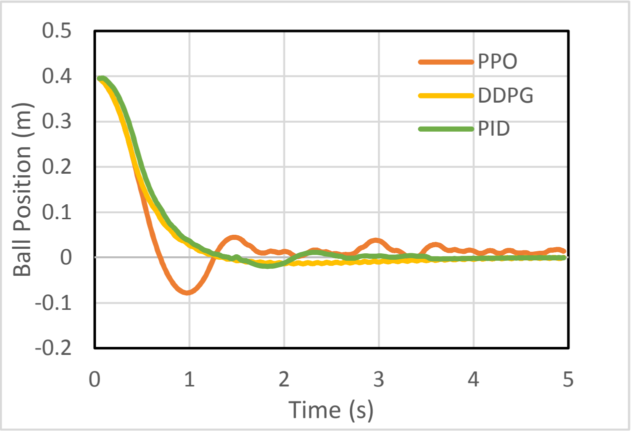


Fig. 13. Performance analysis of positioning of the Ball on the Beam with PPO, DDPG and PID controllers

References

[1] Hammoudeh, Ahmad, (2018). A Concise Introduction to Reinforcement Learning, 10.13140/RG.2.2.31027.53285.

[2] R. S. Sutton and A. G. Barto, Reinforcement learning: An Introduction, 2nd ed. Cambridge, MA: MIT Press, 2017

[3] R. Matthew Kretchmar, "*Synthesis of reinforcement learning and robust control theory*," Ph.D. dissertation, Dept. Comp. Sci., Colorado State University, Fort Collins, 2000.

[4] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O., "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347.

[5] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., and Wierstra, D., "Continuous control with deep reinforcement learning," arXiv preprint arXiv: 1509.02971.

[6] Chen, M., Lam, H. K., Shi, Q., and Xiao, B., "Reinforcement Learning-based Control of Nonlinear Systems using Lyapunov Stability Concept Fuzzy Reward Scheme," IEEE Transactions on Circuits and Systems II: Express Briefs.vol.1, pp. 1-5, 2019.

[7] Hendzel, Z., Burghardt, A., and Szuster, M., "Reinforcement learning in discrete neural control of the underactuated system," *in Proc. 2013, International Conference on Artificial Intelligence and Soft Computing* vol. 37, pp. 425-436, 2007.

[8] Shi, H., Sun, Y., and Li, J., "Dynamical motor control learned with deep deterministic policy gradient," Computational intelligence and neuroscience, vol. 2018, pp. 1-11, 2018.

[9] Sunjai Nakshatharan S., Josephine Selvarani Ruth D. and Dhanalakshmi K.," Servo control of an under actuated system using antagonistic shape memory alloy," Smart Structures and Systems, vol. 14, pp. 643-658, 2014

[10] Josephine Selvarani Ruth D., Sunjai Nakshatharan S. and Dhanalakshmi K., "Interrogation of undersensing for an underactuated dynamic system", IEEE Sensors, vol. 15, pp. 2203-2211, 2015.

[11] Wang, Q., Mi, M., Ma, G., and Spronck, P. I. E. T. E. R., "Evolving a neural controller for a ball-and-beam system," *in Proc. 2004 International Conference on Machine Learning and Cybernetics,* Vol.2, pp. 757-761.

[12] Amjad, M., Kashif, M. I., Abdullah, S. S., and Shareef, Z., "Fuzzy logic control of ball and beam system," *in Proc. 2010 2nd International Conference on Education Technology and Computer,* Vol. 3, pp. 489-493.