Reinforcement Learning and Fuzzy PID Control for Ball-on-plate Systems

Jia-Hsun Lo* , Po-Kai Wang* , and Han-Pang Huang**
Department of Mechanical Engineering, National Taiwan University
**Corresponding Author

Abstract— **Ball-on-plate system (BPs) is a well-known two-dimensional balancing problem that many types of controllers were implemented. The Reinforcement Learning (RL) and Fuzzy PID controllers are proposed to improve the performance of time-domain response. The comparison of the controllers with PD controller shows the high robustness of disturbance rejection and the requirements of performance are satisfied. The application of RL becomes a solution with different characteristics from other traditional controllers, and the performance of Fuzzy PID is confirmed in the BPs control.**

I. Introduction

Ball-on-plate system (BPs) is a two-dimensional non-linear system that many types of controllers were implemented. In 2015, Fabregas et al. [2] implemented a PD controller in Easy Java Simulation, and another research realized the real-time control of BPs by using PID controller in 2016 [4]. Those results showed that the performance of linear controllers can barely balance the ball. Other types of controllers, for example, "Observer integrated backstepping control" [5] and "Approximate feedback linearization and NN-based PID" [6] were designed to pursue high stability or accuracy in real-time control. In 2021, fuzzy logic controller based on chicken swarm optimization was proposed [8] and in 2022 PD tuned fuzzy logic controller was designed to improve the response in time-domain [7]. The performance of the controller in step response and tracking has been studied completely. The objective becomes the robustness of the controller. Although a controller based on H-infinity of robust control was designed for BPs in 2022 [9], the disturbance rejection is seldom discussed in controlling BPs.

In this research, we will focus on the whole designing process of BPs, by using machine learning and a modified PID algorithm based on fuzzy concepts. The performance in time-domain is required and we would like to discuss more about the robustness and disturbance rejection of the controllers.

II. Background

In this section, two main method which is used in our controller design are the reinforcement learning (RL)and fuzzy PID. Both are the non-linear controller but designed in a completely different way.

A. Reinforcement Learning

The main objective of RL is to construct an agent in specific environment and make it complete the task automatically. The design of Q-table is like a list which tells the agent how to act in different situation. This research used Double Deep Q-learning algorithm [3], which took a well-trained Neural Network as Q-table, to solve the controlling problem. It is a classical RL algorithm for discrete action space. There are some points that makes the controller successful. The design of reward leads the performance to different possibility, and the transformation from physical system to mathematical model influence the similarity between virtual and actual environments. The application of Double Deep Q-learning appears in many tasks like double pendulum or rocket simulator. We are interested in the high robustness and learning-based design of RL and make it a new way to solve controlling problems.

B. Fuzzy PID

Fuzzy PID was first designed in 1999 [1] combining traditional proportional-integral-derivative (PID) controller with fuzzy logic. It inherits high robustness of fuzzy logic control and good time-domain performance of PID control. The error is fuzzified through several fuzzy rule and operated similar with PID control. Same as Fuzzy logic control, the membership function and fuzzy rule are designed by experience, which makes Fuzzy PID control have non-linear characteristics. We are interested in the high robustness and linear accuracy of experience-based design to compare it with the learning-based design in the controlling problem of BPs.

III. Method

A. Ball-on-plate System

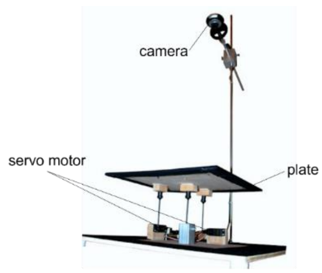
As shown in Figure 1, a ball-on-plate system (BPs) mainly consists of a plate and a ball. The plate is connected to two linkages and motors, above which a camera is served as the observer to receive the position of ball. The control objective is to balance the ball to a specific position by controlling the motors in both XY direction.

Jia-Hsun Lo is with the Department of Mechanical Engineering National Taiwan University, Taipei, Taiwan (e-mail: jason89502@gmail.com).

Han-Pang Huang is with the Department of Mechanical Engineering, National Taiwan University, Taipei, Taiwan (corresponding author, e-mail: hanpang@ntu.edu.tw).

2022 International Automatic Control Conference (CACS) | 978-1-6654-9646-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/CACS55319.2022.9969795

Figure 1. Device of BPs [10]



The mathematic model of BPs is derived below. A ball with mass m and radius $R = 0.04$ contacts with the plate surface with tilting angle α and β in two directions. The inertia is denoted as I, and the gravity constant is denoted as $g = 9.81$. As shown in Figure 2, two forces act on the ball, gravity and centrifugal, causing the translation and rotation of the ball. Assuming the contact surface of ball and plate is pure rolling, the equations of motion can be written as follows:

$$\begin{aligned}\left(m+\frac{I}{R^2}\right)\ddot{x}-m\left(x\dot{\alpha}^2+y\dot{\alpha}\dot{\beta}\right)+mg\sin\alpha &= 0\\ \left(m+\frac{I}{R^2}\right)\ddot{y}-m\left(y\dot{\beta}^2+x\dot{\alpha}\dot{\beta}\right)+mg\sin\beta &= 0\end{aligned} \tag{1}$$

where the gravity is:

$$F_g = -mg\sin\alpha \tag{2}$$

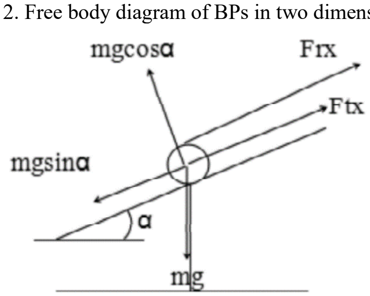
and the centrifugal force is:

$$F_c = m\left(x\dot{\alpha}^2+y\dot{\alpha}\dot{\beta}\right) \tag{3}$$

The translation, rotation, and inertia of ball are:

$$F_{tx} = m\ddot{x}\ ,\ F_{\tau x} = \frac{I\ddot{x}}{R^2}\ ,\ \mathrm{I} = \frac{2}{3}mR^2 \tag{4}$$

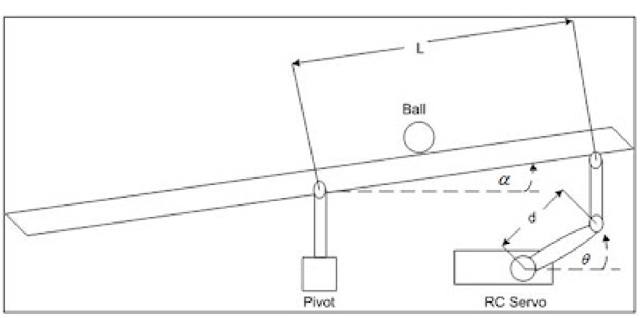
Figure 2. Free body diagram of BPs in two dimensions [4]



The angles of plate with the length $L = 0.25$ in two directions are controlled by the motor angle and the linkage with length $d = 0.08$, as shown in Figure 3. The relation between tilting angle and motor angle is denoted in the following equation:

$$\sin\alpha = \frac{d\sin\theta}{L} \tag{5}$$

Figure 3. Geometrical constraint [10]

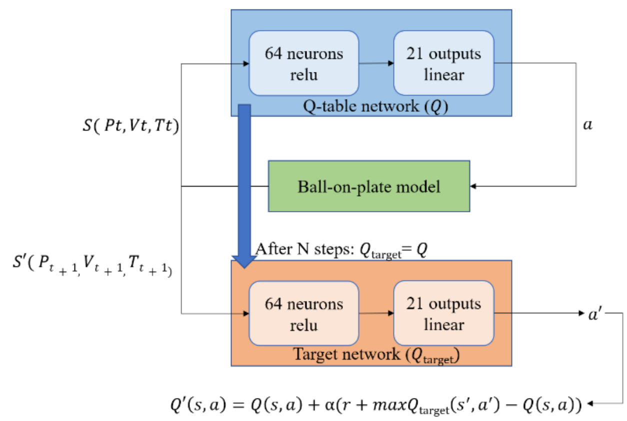


Combining equations (1) and (5), the non-linear model of BPs is constructed. The controllers in the following section are designed based on this mathematical model.

B. *Controller Design with Reinforcement Learning*

The reinforcement learning (RL) controller is designed based on Double Deep Q-learning Network (DDQN), which provides discrete action space by using Neural Network (NN). The RL controller takes actions according to its NN-fitted Q-table. As shown in Figure 4, the NN-fitted Q-table is composed of three layers. The inputs are position, velocity, and targets of the ball. Targets are the position which the ball is asked to reach and maintain stable. The outputs are the scores of 21 actions (±15˚, 1.5˚ increasing per action) at that moment. The hidden layer and output layer are activated by relu and linear function with Adam optimizer. Two controllers are designed for two directions.

Figure 4. Model of DDQN



The core of the learning process is to find out a propriate rule of reward. The error of the control is defined in equation. As shown in Table I, the reward is decided with difference of error, boundary condition, and steady state, making ball reach the target and stay under error requirement. x_n is the position of ball, x_t is the target of ball and E_s is the required steady state error. Each episode has 50 discrete moments for totally 5 seconds. The score of an episode is the sum of the 50 rewards. To explore better strategy, ε-greedy policy is taken as random decision as shown in equation, which decreases in every episode.

$$E_n = |x_n - x_t| \tag{6}$$

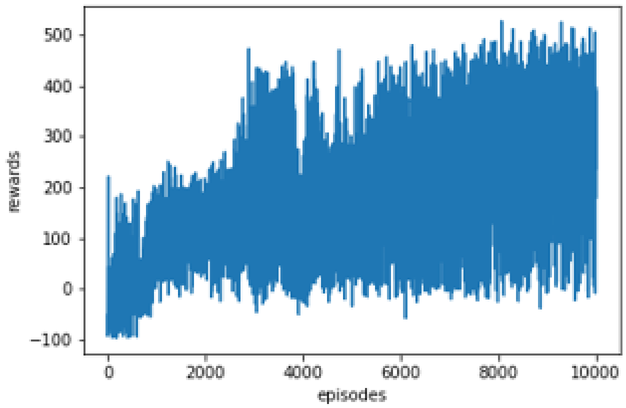
$$\varepsilon = 0.9*(1\text{-}episode/totally\ episodes) \tag{7}$$

TABLE I. DEFINITION OF REWARD

R_n	Conditions
1	$(E_{n-1} - E_n) > 0$
-1	$(E_{n-1} - E_n) < 0$
-1	$\lvert x_n \rvert > \mathrm{L}$
10	$E_n < E_s$ over 1 s

The model is trained after 10000 episodes with 0.1s of updating period. 32 data per curve fitting are shuffled from a queue with the size of 640. As shown in Figure 5, it takes 16s per episode, totally 44.4 hours for training process. The increasing of rewards with oscillation shows that some random factors effect on the rewards, which is mainly the initial condition take place rather than the ε-greedy policy in the later episodes. The score shows the convergence of rewards after 6000 episodes, due to the decreasing of ε.

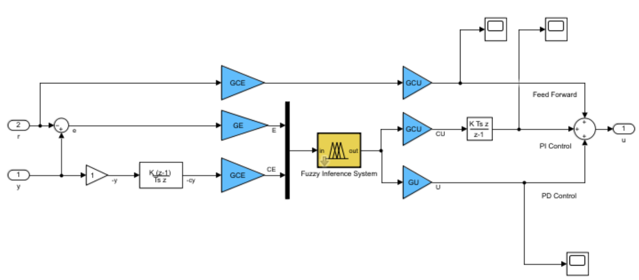
Figure 5. Score of rewards changes in model training



C. Controller Design with Fuzzy PID

The fuzzy PID controller is designed in MATLAB toolbox, which is a modified PID controller by Fuzzy logic with non-linear controlling characteristics. As shown in Figure 6, the error and its derivative are fuzzified. GCU, GCE, GU, GE are the parameters in fuzzy PID. y is the input position. u is the output action. The proportion, integral and differentiation can be derived in the equations (8).

Figure 6. Block diagram of Fuzzy PID controller



$$\begin{aligned} Kp &= GCU * GCE + GU * GE \\ Ki &= GCU * GE \\ Kd &= GU * GCE \end{aligned} \tag{8}$$

The error is set [-0.25, 0] as “Negative,” [0, 0.25] as “Positive.” The derivative of error is set [-0.2, 0] as “Negative,” [0, 0.2] as “Positive.” As shown in Figure 7, the membership function of fuzzy logic is designed by using nonlinear Gaussian function with σ = 0.1. The output u has 3 tags: Max (15), Zero (0), and Min (-15), and its value ranges from -15 to 15 degree. Four Fuzzy logic rule is the main part to decide the controlling action, which is written as follows:

- If E is negative and CE is negative, then u is “Min”.
- If E is negative and CE is positive, then u is “Zero”.
- If E is positive and CE is negative, then u is ”Zero”.
- If E is positive and CE is positive, then u is “Max”.

The action is calculated by the summation of u weighted by its score as written in equation. Combining the output and input, the control surface of the fuzzy PID controller is shown in Figure 8.

$$y = \frac{\Sigma s_i u_i}{\Sigma s_i} \tag{9}$$

Figure 7. Membership function of Fuzzy logic

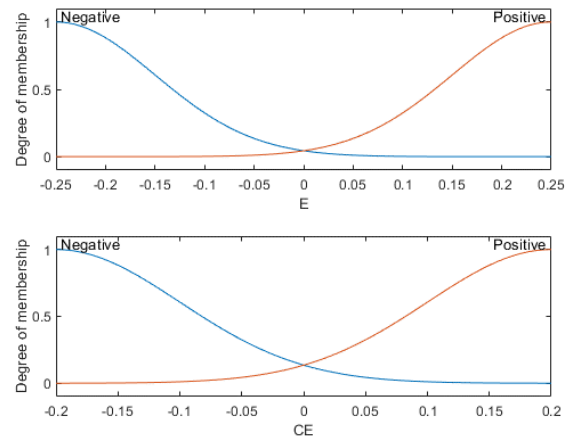
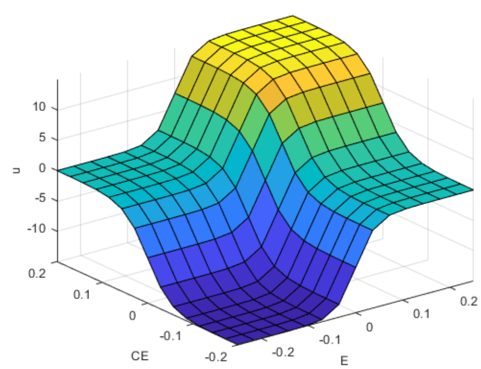


Figure 8. Control surface of Fuzzy PID controller

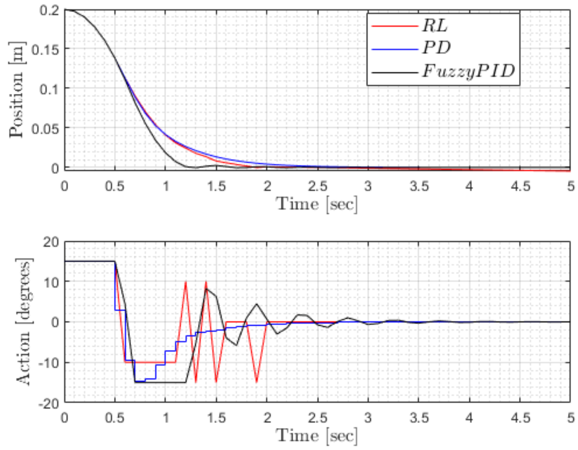


IV. Results

A. Nonlinear System Control

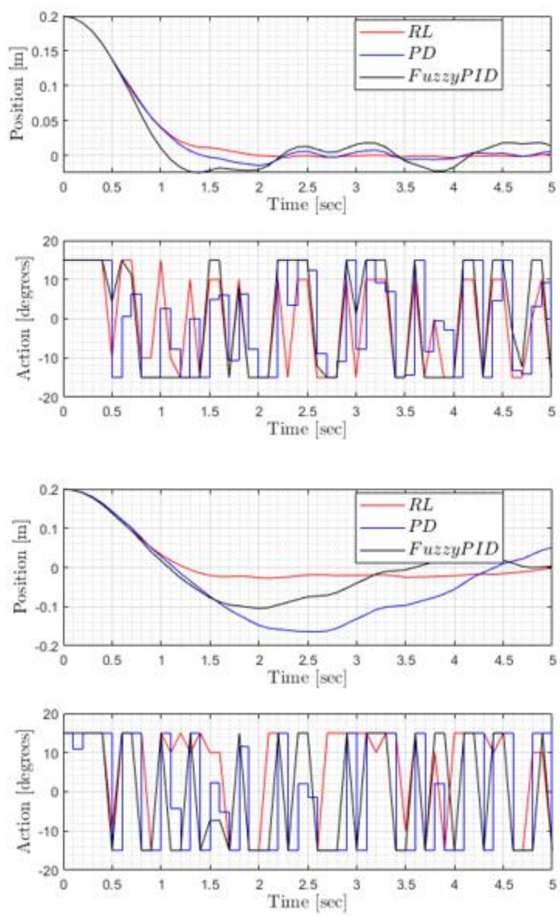
To test the performance of these two controllers, the mathematical nonlinear model is constructed in MATLAB\SIMULINK. The RL controller is built in Keras model and trained under Python environment. The Fuzzy PID controller is designed in MATLAB Fuzzy toolbox (Kp=5, Ki=0.001, Kd = 2.4). The PD controller (Kp=5, Kd = 2.9) is taken as reference to compare with our two controllers, As shown in Figure 9, we input a command of step response to conduct time-domain analysis, in which the error requirement is 0.01 in our design. Three controllers satisfy the error requirement, balancing the ball from point (0.2, 0.2) to the center (0, 0). Fuzzy PID has the fastest response, and the response of RL controller is slightly faster than PD controller.

Figure 9. Step response of BPs with three controllers



To check the robustness, a normal distributed noise is added to the backward signal. As shown in Figure 10, the RL controller shows strong robustness under the disturbance. Fuzzy PID has better robustness than PD controller in large noise but worse in small noise.

Figure 10. Step response of BPs with noise added (up ±0.01, down ±0.05)



Tracking tests are conducted for BPs, making the ball move along a square trajectory. As shown in Figure 11 and 12, three controllers can complete the tracking test. The reference trajectories are denoted as red line and the real trajectory of ball is denoted as blue line. Especially, fuzzy PID controller makes the least error, and the RL controller is not really precise at tracking, probably due to the discrete action space. After the noise added to X direction, RL controller has better disturbance rejection than fuzzy PID controller to make the ball stay in line.

Figure 11. Tracking along square route without noise added

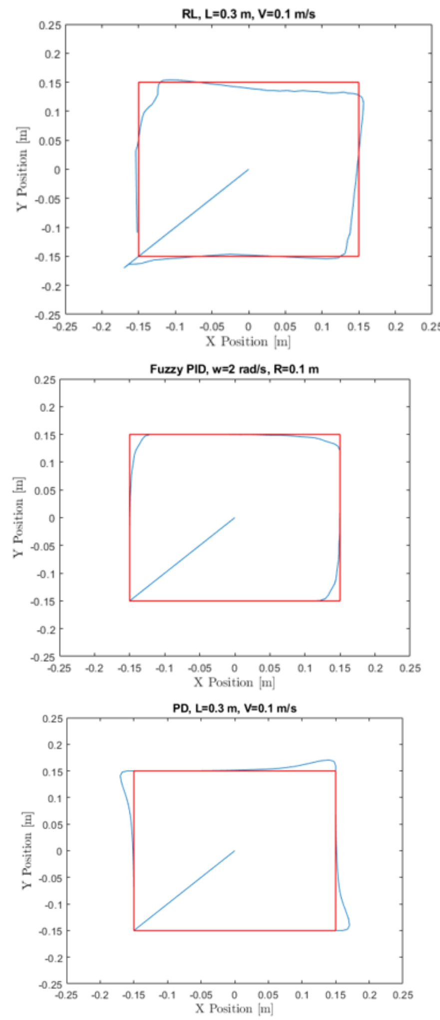
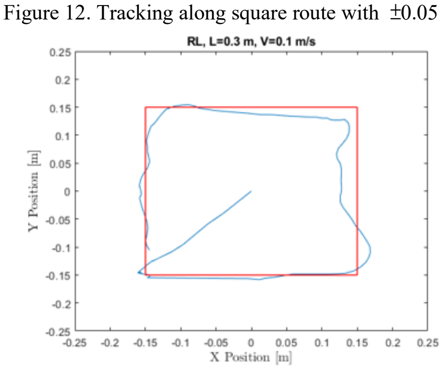
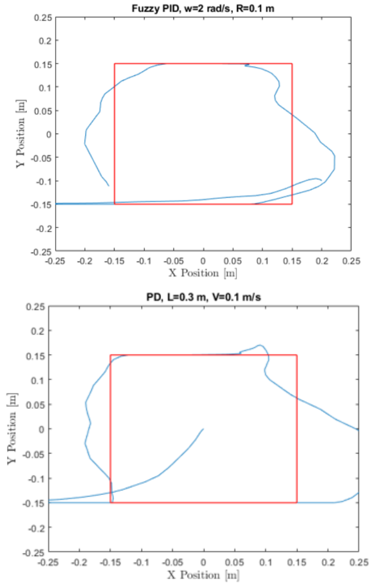


Figure 12. Tracking along square route with ±0.05 m noise

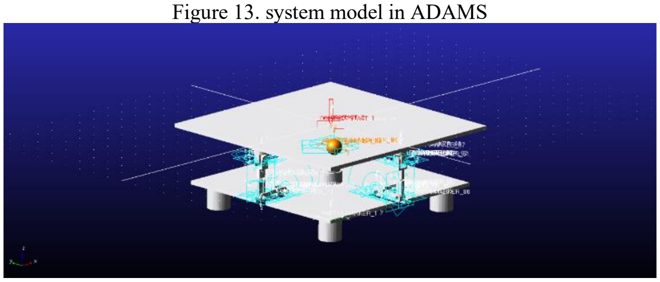




B. ADAMS Control Co-simulation

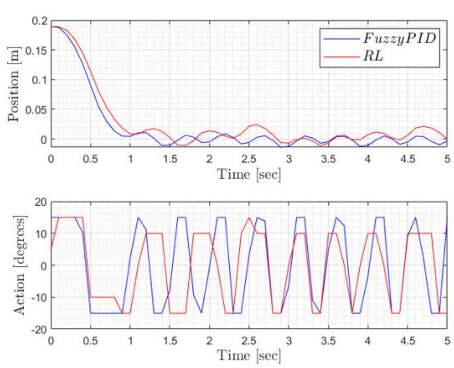
Automatic Dynamic Analysis of Mechanical System (ADAMS) is a software that simulates a dynamic system under computer-aided design (CAD) model. As shown in Figure 13, the CAD model is constructed by Inventor software and imported into ADAMS environment. Some constraints are imposed on the system: revolute joint and surface contact of ball and plate.

Figure 13. system model in ADAMS



The implementation of controllers combining MATLAB with ADAMS is shown in Figure 13. Two controllers can balance the ball to the target with step response, and they have similar actions and oscillation. The error of fuzzy PID is $4 \cdot 10^{-6}$ and the error of RL controller is 0.007. The results show that both controllers reach the requirement of error and complete the ball balancing task.

Figure 14. Step response of BPs in ADAMS environment



C. Discussion

Through step response and tracking task, the result shows that RL controller and fuzzy PID have their advantages against traditional PD controller. As shown in Table II, three controllers take position as input. RL controller has discrete action space, which is the reason for higher steady state error but also easier for digital implementation. In comparison to RL controller, three parameters in fuzzy PID must be tuned and the rule in fuzzy logic is more intuitive to design. The Calculation of RL is slower than other two controllers because of the ability to calculate NN. It can be improved by using better CPU or GPU to reduce the time delay. fuzzy PID and RL controllers show high robustness and higher noise rejection than PD controller. This is the main advantage of the two controllers compared with PD controller.

TABLE II. PERFORMANCE OF THREE CONTROLLERS.

Controllers	PD	Fuzzy PID	RL
Input	Position	Position	Position
Output	Continuous action space	Continuous action space	Discrete action space
Design	Tuned	Complicated tuned	Machine learning
Calculation	Faster	Fast	Slow
Robustness	Low	high	higher
Steady-state error	$4 \cdot 10^{-6}$m	$4 \cdot 10^{-6}$m	0.007 m

V. CONCLUSION

In this research, we design two controllers by using RL and fuzzy PID to satisfy error requirements and high robustness. The RL controller performs high ability of disturbance rejection and fuzzy PID controller performs good balance between steady state error and ability of disturbance rejection. The result shows the potential of Reinforcement learning that it not only operates high-level decision but also becomes a spontaneous controller to solve controlling issue, and the interference of human being experiment can make the traditional linear control more flexible with non-linear characteristics. With the progress of high-speed calculation and virtual environments for dynamics, the disadvantages of Reinforcement learning will be eliminated. It indeed becomes an alternative to solve controlling problems.

ACKNOWLEDGMENT

All members from Robotics Lab who provided insight and expertise are greatly appreciated.

REFERENCES

[1] James Carvajal, Guanrong Chen, Haluk Ogmen, “Fuzzy PID controller: design, performance evaluation, and stability analysis,” *Information Sciences*, Vol. 123, pp. 249-270, 2000.
[2] E. Fabregas, J. Chac´on, S. Dormido-Canto, G. Farias, S. Dormido, “Virtual laboratory of the ball and plate system,” *International Federation of Automatic Control*, Vol. 50, No 1, pp. 152-157, 2015.

[3] Hado van Hasselt, Arthur Guez, David Silver, "Deep reinforcement learning with double Q-learning," *Thirtieth AAAI Conference on Artificial Intelligence*, pp. 2094-2100, 2015.

[4] G. Joselin Retna Kumar, N. Showme and M. Aravind, R. Akshay, "Design and control of ball on plate system," *International Science Press*, Vol. 9, No. 34, pp. 765-778, 2016.

[5] Jie Ma, Hao Tao, Jingwen Huang, "Observer integrated backstepping control for a ball and plate system," *International Journal of Dynamics and Control*, Vol. 9, No. 1, pp. 141-148, 2021.

[6] Amin Mohammadi, Ji-Chul Ryu, "Neural network-based PID compensation for nonlinear systems: ball-on-plate example," *International Journal of Dynamics and Control*, Vol. 8, No. 5, pp. 178-188, 2020.

[7] Nikita Nikita, Bharat Bhushan, "Effect of parameter variation of ball balancer system using PD and Fuzzy control," *IEEE Delhi Section Conference*, New Delhi, India, 2022.

[8] Ahmed A. Oglah, Mohammed Majid Msallam, "Real-time implementation of Fuzzy logic controller based on chicken swarm optimization for the ball and plate system," *International Review of Applied Sciences and Engineering*, 2021.

[9] G. Rigatos, G. Cuccurullo, K. Busawon, et al., "Nonlinear optimal control of the ball and plate dynamical system," *AIP Conference Proceedings*, Crete, Greece, Vol. 2425, 2022.

[10] Ball on Plate control. Electronics & Control Project. https://sites.google.com/site/controlandelectronics/b