A SURVEY OF LINEAR CONTROL FOR EXPERIMENTAL BALL AND BEAM WITH MIDDLE AXIS

DUY-KHOA HUYNH, DUC-ANH-QUAN NGUYEN*, DUC-HIEN TRAN, XUAN-HUY NGUYEN, TUAN-KIET LE, GIA-LONG PHAM, CHI-DANH LE, NGUYEN-DUC-HUY BUI, GIA-HUY DANG, MINH-TAM NGUYEN

Ho Chi Minh City University of Technology and Education, Ho Chi Minh City, Vietnam

**Corresponding author: 20151408@student.hcmute.edu.vn*

(Received: 26 November 2024; Accepted: 3 December 2024; Published on-line: 1 March2025)

ABSTRACT: This paper presents an experimental study of linear control algorithms applied to a Ball and Beam system with a central axis. The focus is on evaluating the ball's ability to remain balanced around the central axis and assessing the stability of linear control strategies in real-world applications. The system is controlled using an STM32F4 microcontroller, which manages a DC motor to adjust the beam's angle in response to the ball's position. Through a series of experiments and data analysis, the study explores the effectiveness of linear control in addressing the system's nonlinear dynamics and discusses the practical challenges faced during implementation. The results contribute to a deeper understanding of advanced control techniques and their potential applications in engineering.

KEY WORDS: LQR control, PID Control, Pole placement, Ball and Beam, Linear Control.

1. INTRODUCTION

The ball-and-beam system, a classic example in control theory, offers the unique challenge of balancing a ball along a beam by adjusting its tilt. This system, often used in educational and research environments, provides valuable insights into feedback control and dynamic stability. Specifically, in a middle-axis configuration, the ball must be controlled to remain at the center of the beam, requiring precise adjustments to the beam's angle. In such systems, linear control methods are employed to effectively manage the forces and maintain stability.

This article delves into applying linear control techniques to balance the ball along the ball's middle axis of the Ball and Beam system using the STM32F4 microcontroller. Renowned for its robust processing capabilities, the STM32F4 is ideal for real-time control applications [16], enabling high-precision feedback to adjust the beam's angle with great accuracy. By utilizing a linear control approach, such as Proportional-Derivative (PD) [13] or Proportional-Integral-Derivative (PID) controllers [3], this system demonstrates the power of microcontroller-based solutions in solving complex dynamic problems. Through this work, we explore how the STM32F4 chip enhances the efficiency and responsiveness of the control system, ensuring optimal performance in the balancing task.

We use the STM32F4 microcontroller to control a DC motor for balancing the ball with linear control algorithms [2]. In this paper, we implement three control methods: linear quadratic regulator (LQR), Proportional Derivative (PD), and Pole Placement. After developing these algorithms, we test them on a real-life model to evaluate their adaptability and performance in a practical system.

Despite the effectiveness of established educational models, their high costs can be prohibitive. Consequently, researchers have explored developing cost-effective, real-time models to improve accessibility for educational purposes. Previous studies have shown that while self-made models can provide valuable insights, the control hardware, such as the DSP TMS320F28335, often remains expensive and less adaptable for students. In contrast, Arduino platforms, while affordable and supported by a strong community, mainly support basic control algorithms like PID [3] and linear control. Their limitations become evident when tackling more complex systems. Arduino struggles with the high-velocity operations required for intricate systems like tower cranes, which are often classified as MIMO (multi-input, multi-output) under-actuated models.

The STM32F4 microcontroller, however, emerges as a more suitable alternative, capable of executing sophisticated control algorithms through MATLAB embedding. This platform has proven effective for managing high-order systems, including inverted pendulums, and has successfully implemented PID control [3] in MIMO systems like tower cranes. Compared to DSP boards, the STM32F4 is more affordable, making it a practical choice for educational use. In contrast, its ability to support advanced control strategies makes it a valuable tool for simulation and experimental applications.

This study applies linear control algorithms to a Ball and Beam system using the STM32F4 [10] to control a DC motor. The goal is to investigate the ball's balancing capabilities around the central axis and assess the stability and performance of linear control in real-time applications [8]. Through this study, we aim to comprehensively evaluate linear control's effectiveness in a practical Ball and Beam setup, contributing to the broader field of control engineering.

2. MODELING OF BALL AND BEAM SYSTEM

The Ball and Beam system is an engaging and dynamic setup that consists of two straight sticks forming the beam, a smooth metal ball that rolls along its length, and a DC motor responsible for adjusting the beam's angle. A position sensor made from a resistive wire runs along the beam to track the ball's position, generating a variable voltage signal that changes as the ball moves. An encoder attached to the DC motor reads the angle of the beam, allowing for precise adjustments [5].

As the DC motor pivots the beam, it creates a moment that influences the ball's position, enabling real-time control and stability. This intricate interaction between the components not only showcases the principles of feedback control but also offers hands-on experience in managing dynamic systems. The ball-and-beam system is not just a project; it's a fascinating exploration of engineering concepts in action [7].

2.1. LQR algorithm

We can determine the direction of rotation and the movement of the ball and beam, distinguishing between positive and negative values. It is illustrated in Fig. 1 from [1]. Choose the potential energy reference point at point O, which is located at the center of the horizontal beam. Since the reference point O coincides with the center of mass of the beam, the potential energy of the beam remains zero throughout the motion.

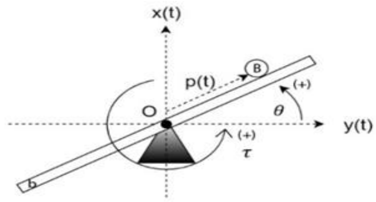

Fig. 1. Determine the positive and negative directions for the beam's rotation and the ball's movement

Table 1. Oversee elements in the system

Parameter	Value
L_b	0.34 (m)
m_b	0.35 (kg)
m_B	0.045 (kg)
R	0.011 (m)
θ	N/a (rad)
p	N/a (m)
τ	N/a (Nm)
e	N/a (V)
$p=v_b$	N/a (m/s)

Based on the reference from source [2], we have derived the equation to describe the system.

Linearizing the ball and beam system to analyze controllability:

$$x = [p \quad \dot{p} \quad \theta \quad \dot{\theta}]^T = [x_1 \quad x_2 \quad x_3 \quad x_4]^T \tag{1}$$

The mathematical equation of the system becomes:

$$\begin{cases} \dot{x}_1 = x_2 = f_1(x,e) \\ \dot{x}_2 = \dfrac{m_B x_1 x_4^2 - m_B g \sin x_3}{m_B + \dfrac{J_B}{R^2}} = f_2(x,e) \\ \dot{x}_3 = x_4 = f_3(x,e) \\ \dot{x}_4 = \dfrac{K_t e - K_t K_b x_4 - 2 m_B R_m x_1 x_2 x_4 - R_m m_B g x_1 \cos x_3}{\left(m_B x_1^2 + J_b\right) R_m} = f_3(x,e) \end{cases} \tag{2}$$

The system is in the form of:

$$\dot{x} = f(x,e) \tag{3}$$

With:

$$f = [f_1 \quad f_2 \quad f_3 \quad f_4] \tag{4}$$

Choose the equilibrium position as point O, located at the center of the horizontal beam, where the ball can remain balanced at point 0, p=0 the ball is located at the center of the horizontal beam, $\dot{p}$=0 is when the ball is stable, e=0 the horizontal beam is positioned horizontally, θ=0 the horizontal beam is stationary and not moving, with the voltage supplied to the motor equal to zero: p=0; $\dot{p}$=0; θ=0; $\dot{\theta}$=0.

The ball and beam system can be approximated as linear:

$$\dot{x} = Ax + Be \quad (5)$$

$$u_{LQR} = \text{-}Ku \quad (6)$$

We used a command in Matlab to find K:

$$K = dlqr(A, B, Q, R) \quad (7)$$

With:

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} \end{bmatrix}_{x=0,e=0} ; B = \begin{bmatrix} \frac{\partial f_1}{\partial e} \\ \frac{\partial f_2}{\partial e} \\ \frac{\partial f_3}{\partial e} \\ \frac{\partial f_4}{\partial e} \end{bmatrix}_{x=0,e=0} \quad \mathrm{Q} = \begin{bmatrix} \mathrm{Q}_1 & 0 & 0 & 0 \\ 0 & \mathrm{Q}_2 & 0 & 0 \\ 0 & 0 & \mathrm{Q}_3 & 0 \\ 0 & 0 & 0 & \mathrm{Q}_4 \end{bmatrix}, \mathrm{R} = \mathrm{R}_1 \quad (8)$$

2.2. Pole placement algorithm

The Pole Placement Algorithm [15] is a fundamental technique [11] used in control theory for designing state feedback controllers. It involves selecting desired locations (poles) for the closed-loop system's characteristic equation, which determines the system's stability and dynamic behavior. The idea is to adjust the feedback gains such that the poles of the closed-loop system are placed at specific locations in the complex plane, ensuring desired system performance such as stability, transient response, and damping [12].

Pole placement control uses the gain matrix K, similar to LQR, but determines K through a different method.

$$\dot{x} = Ax + Be \quad (9)$$

we have:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ c &= Cx + Du \end{aligned} \quad (10)$$

The control system uses voltage value to keep the ball balanced around 0 points, so we have u bellow:

$$u = -Kx \quad (11)$$

To find K, we use the Matlab command to calculate the open-loop eigenvalues:

$$E = eig(A) \quad (12)$$

After using the command, we select the complex value and negative value to use as close-loop eigenvalue:

$P = [a \quad b \quad c \quad d]$; with a, b <0 and b, c is a complex number.

To find K, we use the Matlab command:

$K = place(A, B, P)$; with A and B is the matrix from:

$$Q = \begin{bmatrix} Q_1 & 0 & 0 & 0 \\ 0 & Q_2 & 0 & 0 \\ 0 & 0 & Q_3 & 0 \\ 0 & 0 & 0 & Q_4 \end{bmatrix}, R = R_1$$

2.3. PID algorithm

The PID (Proportional-Integral-Derivative) controller [3] is one of the most widely used control algorithms in industrial and engineering applications. It continuously adjusts the control input to a system based on the error between the desired and actual outputs. The algorithm combines three key terms—proportional, integral, and derivative—to determine the control action.

In this experiment, we use only the proportional (P) and derivative (D) elements to balance the ball [13].

The general form of the PID control law is:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (13)$$

Where:

u(t) is the control signal (input to the system).

e(t) is the error signal, defined as the difference between the desired output (setpoint) and the actual output (measured value).

K_p, K_i, K_d are the **proportional**, **integral**, and **derivative** gains, respectively?

We rely on the LQR value of Eq. 13 to adjust the element values P and D.

2.4. Discussion and Hardware

We utilize MATLAB/SIMULINK software for simulation validation. The mathematical model is computed and identified to perfectly reflect the real-world model, enabling us to accurately evaluate the process using these linear algorithms to assess adaptability through objective functions. Additionally, we integrate this LQR algorithm with the optimized parameters into the experimental model to conclude the smart swarm's search capabilities. Simulations are conducted for 10 seconds, with a system sampling time of 0.01 seconds. The main parameters utilized in this paper are presented below:

https://doi.org/10.51630/ijes.v6i1.149

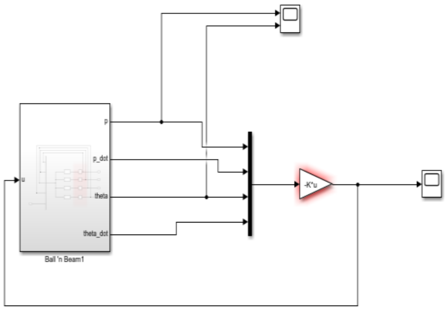

Fig. 2. Ball and Beam using LQR in Simulink

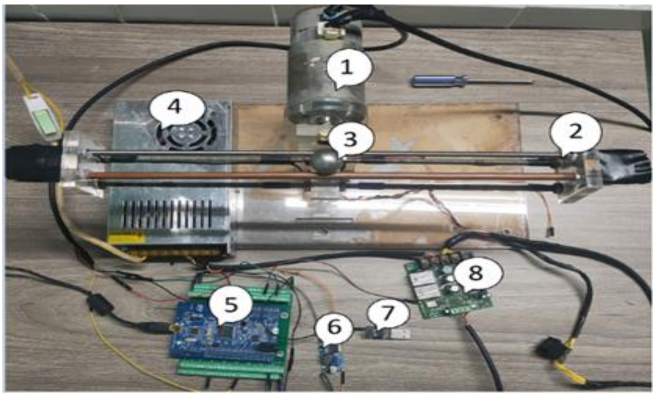

Fig. 3. Real-life ball and beam system

Table 2. Elements of ball and beam system

Station	Parameter	Value	Unit
1	m_B	0.045	kg
2	m_b	0.35	kg
3	L_B	0.34	m
4	K_t	0.06494	N
5	K_b	0.06494	V.s/rad
6	R_m	6.83572	Ω
7	R	0.011	m

Table 3. Parts of a real-life system

Station	Unit
1	DC motor detach with encoder
2	The beam is wrapped with a resistive wire.
3	Iron ball
4	Voltage supplier
5	STM32F4 Discovery
6	Voltage regulator circuit
7	Output reader
8	H-bridge

The system consists of a horizontal beam, a ball, a DC motor, a sensor to read the ball's position, and an encoder attached to the motor shaft to measure the rotational angle of the motor, which corresponds to the tilt angle of the beam. The sensor determines the ball's position on the beam by wrapping the beam with a resistive wire and applying a voltage. This voltage produces an ADC signal, which is then processed to determine the position of the ball based on the ADC reading. The beam can rotate around its central axis thanks to the torque generated by the motor, which is applied to the beam.

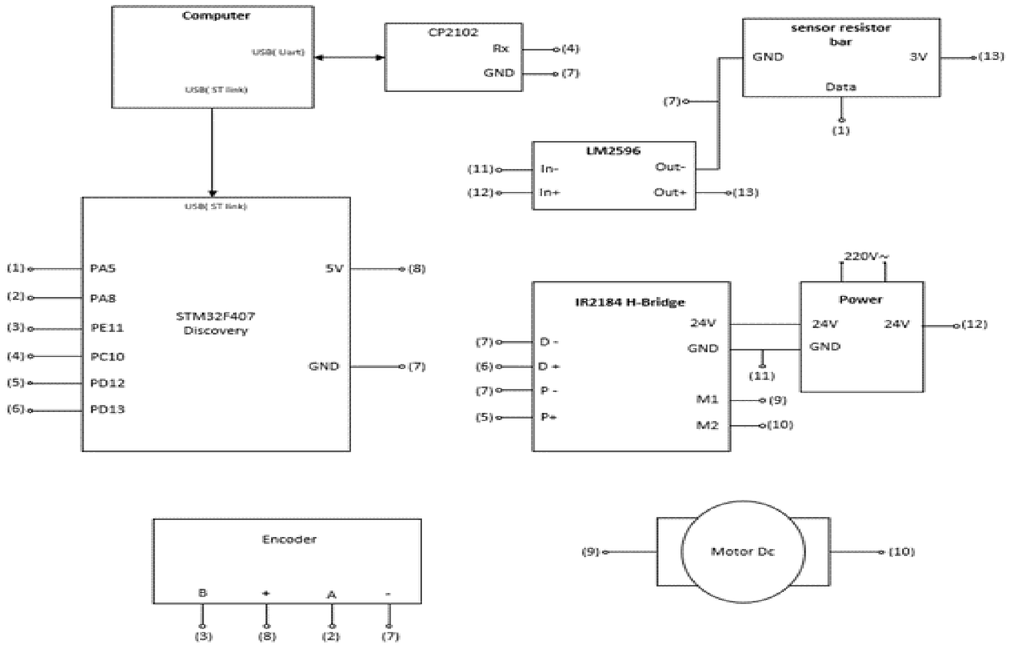

Fig. 4. Connection of Real-life system

3. RESULTS AND DISCUSSION

3.1. Results of LQR Algorithm Method in Real-Life Model

Based on the reference from source [2], we have matrix Q as:

$$Q = \begin{bmatrix} Q_1 & 0 & 0 & 0 \\ 0 & Q_2 & 0 & 0 \\ 0 & 0 & Q_3 & 0 \\ 0 & 0 & 0 & Q_4 \end{bmatrix}; R = R_1 \tag{14}$$

So we got the first result after embedding the code in the real-life hardware with:

$$Q = \begin{bmatrix} 180 & 0 & 0 & 0 \\ 0 & 4.2 & 0 & 0 \\ 0 & 0 & 110 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix} \tag{15}$$

K_1, K_2, K_3, K_4 are described as the system's position, velocity, rotation angle, and angular velocity. So, we have to modify and adjust these elements to stabilize the model.

In this article, we focus on adjusting K_1 and K_4. Position is crucial for balancing the ball, while angular velocity is used to determine the velocity needed to maintain balance.

Table 4. Adjust elements of matrix Q

Parameter	K_1	K_4	**Results**
LQR-1	180	0.5	Fig. 5
LQR-2	180	0.3	Fig. 6
LQR-3	350	0.3	Fig. 7
LQR-4	400	0.3	Fig. 8

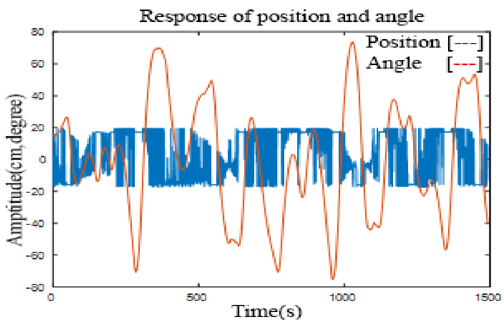

Fig. 5. LQR-1's result

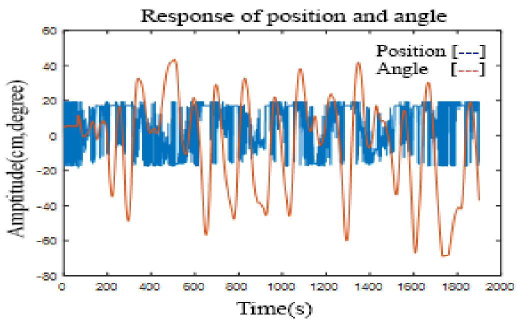

Fig. 6. LQR-2's result

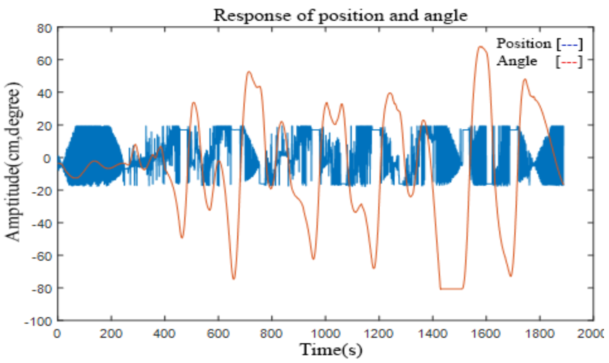

Fig. 7. LQR-3's result

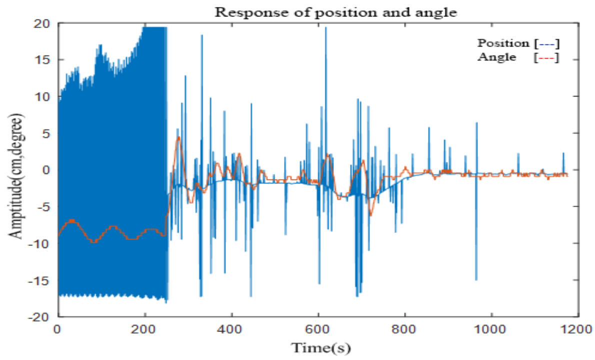

Fig. 8. *LQR-4's result*

The results from the initial changes to matrix Q show that the system attempts to position the ball at zero. However, the DC motor remains unstable when rotating rapidly Fig. 55. Reducing the value of K_4 resulted in the motor rotating in a more controlled manner Fig. 66. Increasing the value of K_1, which has the most significant impact on the ball's position, yielded results showing progress in centering the ball Fig. 7. The final configuration of matrix Q yielded stable results for the real-life system, significantly enhancing its performance and reliability Fig. 8.

3.2. Result of Pole Placement Method in Real-Life Model

After calculating K by command in Matlab, we have the first result:

$$P = [-5.3186 \quad \begin{pmatrix} -0.0386 \\ +5.2753i \end{pmatrix} \quad \begin{pmatrix} -0.0386 \\ -5.2753i \end{pmatrix} \quad -5.2375]$$

Similar to LQR, the elements of P are related to the Q matrix from LQR, with the main elements being the first and last ones in the P matrix. This article focuses on adjusting the main elements to observe how the system responds to the ball's position. (With P_1 and P_4 are negative values).

Table 5. Adjust elements of matrix P

Parameter	P_1	P_4	Result
PP-1	-5.3186	-5.2375	Fig. 9
PP-2	-5.6	-5.2375	**Error! Reference source not found.**
PP-3	-5.6	-5.1	**Error! Reference source not found.**
PP-4	-5.6	-5.12	**Error! Reference source not found.**

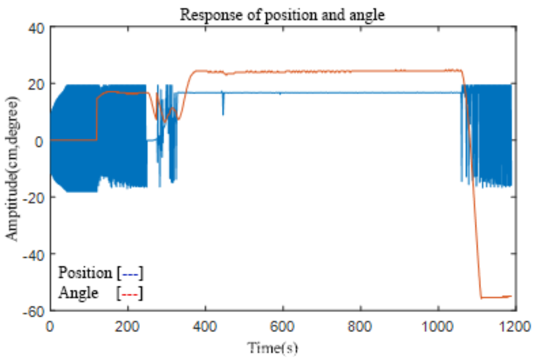

Fig. 9. PP-1's result

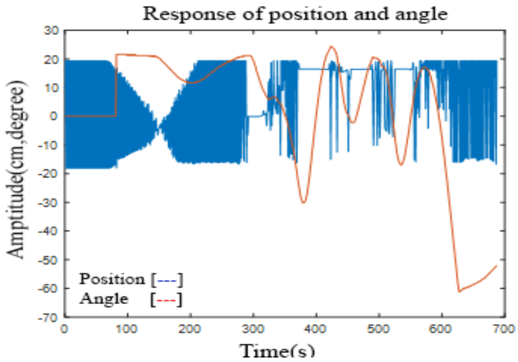

Fig. 10. PP-2's result

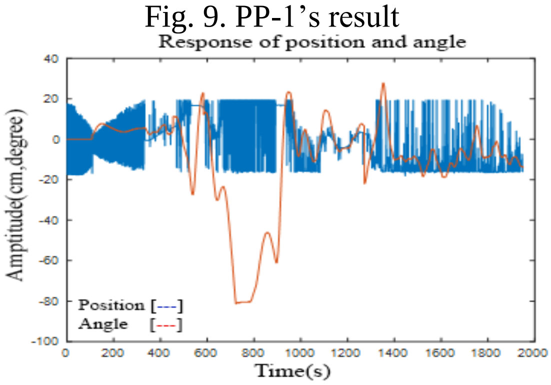

Fig. 11. PP-3's result

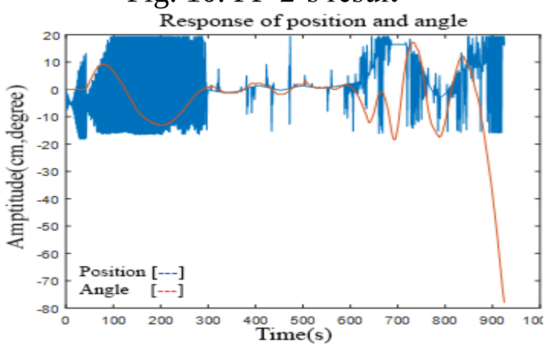

Fig. 12. PP-4's result

After finding P, we obtain the first desired closed-loop eigenvalue. Still, the result is insufficient for the desired performance Fig. 9. Changing P_1 makes the system respond to the position more effectively Fig. 10. We adjusted P_4 to stabilize the rotation of the DC motor. It worked as expected Fig. 11. We made slight adjustments and achieved a stable, temporarily stable result Fig. 12.

https://doi.org/10.51630/ijes.v6i1.149

3.3. Real-Life PD Control Result in Real-Life Model

Based on the matrix Q of LQR. We are using K_1, K_2, K_3, K_4 $K_{p1}, K_{d1}, K_{p2}, K_{d2}$ PD control elements.

$$K_{p1} = 230, K_{d1} = 90, K_{p2} = 150, K_{d2} = 1$$

This article uses P and D control instead of PID control. Our primary focus is adjusting the PD control's angular velocity element to observe how the system responds to the ball's position.

Table 6. Adjust elements of PD control

Parameter	K_{p1}	K_{d2}	Result
PD-1	230	1	Fig. 13
PD-2	230	5	Fig. 14
PD-3	230	8	Fig. 15
PD-4	230	10	Fig. 16

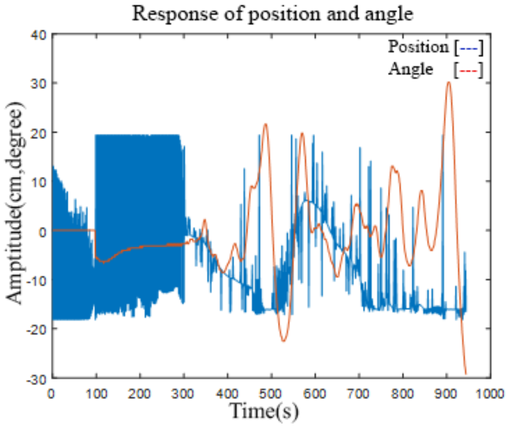

Fig. 13. PD-1's result

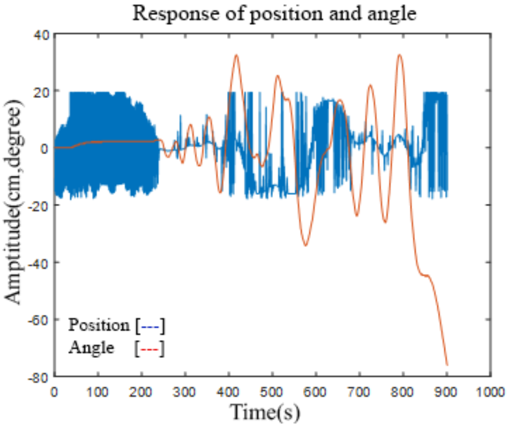

Fig. 14. PD-2's result

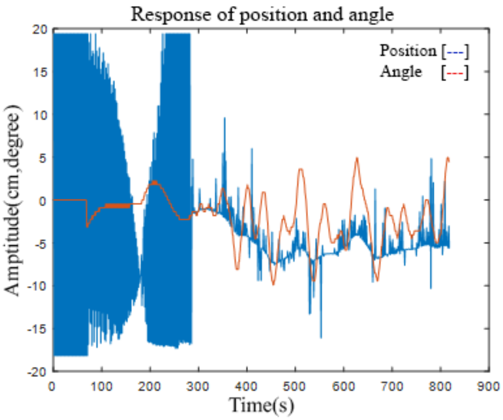

Fig. 15. PD-3's result

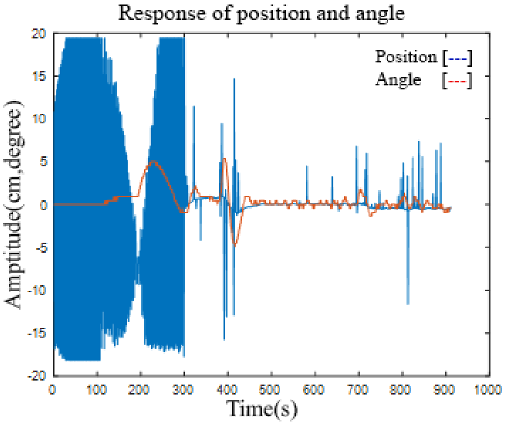

Fig. 16. PD-4's result

After changing K_{p1}, we observed an improvement in the ball's position response when K_{p1} was increased. However, the motor's rotational response was still not satisfactory Fig. 13). Increasing K_{d2} resulted in an outstanding response to the motor's rotation with respect to the ball's position, as shown in Fig. 14 to Fig. 15. Adjusting a little bit of K_{d2} 's value, we achieved very good system stability Fig. 16.

3.4. Evaluation About Effectiveness

To objectively evaluate the controllers presented in Fig. 5 – 16, Table 7 compares the stability and optimality criteria between the controllers. It helps highlight the performance and the improvements made by developing a linear controller for the ball and beam model. The criteria are presented in Table 7 below:

Table 7. Overview comparison of stability and optimality criteria between the controllers

Controller	LQR		Pole Placement		PD	
Target	Beam	Ball	Beam	Ball	Beam	Ball
Unit	Angle(rad)	Position(m)	Angle(rad)	Position(m)	Angle(rad)	Position(m)
Rise time(s)	232	230	300	290	380	370
Settling time(s)	820	830	360	350	470	465
Peak Value	0.0222π	-0.015	0.0111π	-0.02	0.0389π	0.01

Based on the performance comparison of the three controllers (LQR, Pole Placement, and PD), each controller has advantages and disadvantages. The LQR controller has the lowest rise time (232s for beam and 230s for ball), indicating a faster response capability, but it also has the longest settling time (820s and 830s), showing that the system takes longer to stabilize. In contrast, the Pole Placement controller has the shortest settling time (360s for beam and 350s for ball), making it suitable for applications requiring quick stabilization. Regarding the peak value, LQR and Pole Placement achieve smaller values than PD, particularly for the beam's angle 0.0222π and 0.0111π compared to 0.0389π, demonstrating better oscillation minimization. Meanwhile, the PD controller exhibits the highest rise time and peak value, reflecting slower response speed and more significant oscillation. Overall, LQR suits systems requiring high precision, Pole Placement is optimal for scenarios demanding rapid stabilization, and PD can be used in simpler systems with less stringent requirements.

4. CONCLUSION

This study explored linear control methods for balancing the ball in a Ball and Beam system. Linear control techniques, such as LQR, PD, and pole placement, offer distinct advantages and trade-offs regarding stability, complexity, and performance. LQR proved to be the most effective control strategy, offering superior performance compared to PD and pole placement. It provided better stability and faster response in balancing the ball on the beam, making it the most reliable approach for managing the system's dynamics. However, the complexity of LQR, particularly in terms of tuning and computation, makes it more challenging to implement than more straightforward methods like PD and pole placement.

While less complex and easier to implement, PD control did not perform as well as LQR in terms of system stability and responsiveness Fig. 15). It is well-suited for simpler systems or when computational resources are limited, but its performance can fall short for more demanding applications, especially when dealing with nonlinearities or more complex dynamic behaviors.

Pole placement, on the other hand, offers a simpler alternative to LQR [14] with an intuitive approach to control design. However, it did not provide the same level of stability as

LQR. Fig. 7 to 9. While it can be effective in systems where control simplicity is prioritized, it may not be the best choice for applications requiring precise and robust control [15].

In conclusion, while LQR is more complex than PD and pole placement, its effectiveness in balancing the ball in the ball-and-beam system makes it the preferred choice for applications where performance is critical. For simpler systems or educational purposes, PD or pole placement may still be viable alternatives, with pole placement being a good balance between simplicity and control performance, though not as stable as LQR Fig. 8.

The Ball and Beam middle-axis system provides an excellent platform for exploring the challenges and solutions related to nonlinear dynamics and control. While linearization offers a practical way to simplify the design of control algorithms, it also comes with the trade-off of ignoring the nonlinearities that exist in the system. As the system operates further from equilibrium, these nonlinear effects become more significant and can degrade the performance of linear controllers. Alternative control strategies that account for nonlinear behavior, or hybrid approaches combining linear and nonlinear techniques, should be considered to overcome these challenges. By addressing the nonlinear aspects of the Ball and Beam system, it is possible to achieve more robust and adaptive control solutions that improve system performance across a broader range of operating conditions.

ACKNOWLEDGEMENT

This paper is the product of Duy-Khoa Huynh's bachelor thesis at HCMUTE. We thank Dr. Van-Dong-Hai Nguyen for his invaluable supervision and support throughout our research. His guidance has been instrumental in helping us successfully complete this work.

REFERENCES

[1] Nguyen Minh Quan et al, “Fuzzy Controller from Experts’ Rules for Middle Axis Ball and Beam”, Journal of fuzzy systems and control, Vol. 1, No 3, 2023. doi: 10.59247/jfsc.v1i3.94.

[2] Nguyen Dang Khoa Tran, Mai Ha Truong, Huu Dat Nguyen, Van Tu Le, “Research and using LQR control method for Ball and beam system”, Binh Duong University Journal of Science, 2024. doi: 10.56097/binhduonguniversityjournalofscienceandtechnology.v6i3.176.

[3] Ang K.H, Chong G., Li Y.: “PID control system analysis, design, and technology”, in IEEE Transactions on Control Systems Technology, vol.13, no.4, July 2005.doi: 10.1109/TCST.2005.847331.

[4] Tu Dinh Nguyen, Chi-Ngon Nguyen, Le Hoang Dang, Tran Chi Cuong, “Adaptive control ball and beam using Neuron network RBF”, Can Tho University Journal of Science, 2017. doi: 10.22144/ctu.jvn.2017.064.

[5] Shah M. , Rahat Ali, Fahad Mumtaz: "Control of Ball and Beam with LQR Control Scheme using Flatness Based Approach", International Conference on Computing, Electronic and Electrical Engineering (ICE Cube), 2018. DOI: 10.1109/ICECUBE.2018.8610968.

[6] Huynh Thai Hoang " Intelligent control system ", The National University of Ho Chi Minh City Publishing House, 2014. Link: https://cuudongthancong.com.

[7] Nguyen Thi Phuong Ha, Huynh Thai Hoang, “Automatic Control Theory”, The National University of Ho Chi Minh City Publishing House, 2002. Link: https://cuudongthancong.com.

[8] Dang Ngoc Trung, Tran Thi Thanh Hai, “Research LQR control method for Ball and Beam system”,Thai Nguyen University of Technology Journal of Science, no. 72, page 90 – 93. Link: https://vjol.info.vn.

[9] Van-Chinh Pham et al, ―GA-PID Control for Ball and Beam: Simulation and Experiment‖, International Journal, ISSN-L: 1453- 2069; Print ISSN: 1453-2069; Online ISSN: 2359-9855, Vol. 27, No. 2, pp. 29-35 December 2022. doi: https://doi.org/10.24193/rm.2022.2.5.

[10] S. Latif, E. Muhammad and U. Naeem, "Implementation of ball and beam system using classical and advanced control techniques," 2019 International Conference on Applied and

Engineering Mathematics (ICAEM), Taxila, Pakistan, 2019, pp. 74-79. doi: 10.1109/ICAEM.2019.8853822.
[11] T. F. E. Lin S. Tung, "A Study of Pole Placement in Linear Control Design," IEEE Xplore, 1984 American Control Conference. San Diego, CA, USA, 2009. doi: 10.23919/ACC.1984.4788433
[12] F. Gunnarsson, F. Gustafsson, J. Blom, “Pole placement design of power control algorithms”, 1999 IEEE 49th Vehicular Technology Conference (Cat. No.99CH36363), Houston, TX, USA, 2002.doi: 10.1109/VETEC.1999.778433.
[13] Jason J. R. Liu, Nachuan Yang, Ka-Wai Kwok, James Lan, “Proportional-derivative controller design of continuous-time positive linear systems” 2021 International Journal of Robust and Nonlinear Control 32, University of Hong Kong 2021.doi: 10.1002/rnc.5809.
[14] U. Holmberg, P. Myszkorowski, D. Bovin,”A pole-placement approach that matches identification and control objectives”, 1997 European Control Conference (ECC), Brussels, Belgium, 2015.doi: 10.23919/ECC.1997.7082214.
[15] Anjali Rai, Santosh Kumar Suman, Awadhesh Kumar, Shekhar Yadav “Impact of Control Stability using LQR and Pole-placement for Ball and Beam System”, 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021.doi: 10.1109/ICICCS51141.2021.9432281.
[16] Ali Hmidet, Othman Hasnaoui, “Waijung Blockset-STM32F4 Environment for Real Time Induction Motor Speed Control”, IEEE 5th International Congress on Information Science and Technology (CiSt), 2018.doi: 10.1109/CIST.2018.8596416

APPENDIX

Assemble number for Ball and Beam system in Simulink

```
%%Author: Huynh Duy Khoa
clc;
syms x1 x2 x3 x4 u
syms Kt Kb Rm mB mb R Lb g Jb JB

p_init=0.08;
p_dot_init=0.01;
theta_init=0.02;
theta_dot_init=-0.02;

f1 = x2;
f2 = (mB*x1*x4^2-mB*g*sin(x3))/(mB+(JB/R^2));
f3 = x4;
f4= (((Kt*u-Kt*Kb*x4)/Rm)-2*mB*x1*x2*x4-mB*g*x1*cos(x3))/(Jb+mB*x1^2);

A = [diff(f1,x1) diff(f1,x2) diff(f1,x3) diff(f1,x4);
    diff(f2,x1) diff(f2,x2) diff(f2,x3) diff(f2,x4);
    diff(f3,x1) diff(f3,x2) diff(f3,x3) diff(f3,x4);
    diff(f4,x1) diff(f4,x2) diff(f4,x3) diff(f4,x4)];

B = [0;0;0; diff(f4,u)];

%assemble

x1=0;x2=0;x3=0;x4=0;u=0;

Kt = 0.06494;
Kb = 0.06494;
Rm = 6.83572;
```

```
mB = 0.045;
mb = 0.35;
R = 0.011;
Lb = 0.37;
g =  9.81;

JB = (2*mB*R^2)/5; %moment of ball
Jb = (mb*Lb^2)/12; %moment of beam

A =[                                                    0,                  1,
0,                          0;
                                            (mB*x4^2)/(mB + JB/R^2),                  0,    -
(g*mB*cos(x3))/(mB + JB/R^2),             (2*mB*x1*x4)/(mB + JB/R^2);
                                                          0,                  0,
0,                          1;
 (2*mB*x1*(2*mB*x1*x2*x4 - (Kt*u - Kb*Kt*x4)/Rm + g*mB*x1*cos(x3)))/(mB*x1^2 + Jb)^2 - (g*mB*cos(x3) +
2*mB*x2*x4)/(mB*x1^2 + Jb), -(2*mB*x1*x4)/(mB*x1^2 + Jb), (g*mB*x1*sin(x3))/(mB*x1^2 + Jb), -((Kb*Kt)/Rm
+ 2*mB*x1*x2)/(mB*x1^2 + Jb)];

B =[          0;
              0;
              0;
 Kt/(Rm*(mB*x1^2 + Jb))];

f1 = x2;
f2 = (mB*x1*x4^2-mB*g*sin(x3))/(mB+(JB/R^2));
f3 = x4;
f4= (((Kt*u-Kt*Kb*x4)/Rm)-2*mB*x1*x2*x4-mB*g*x1*cos(x3))/(Jb+mB*x1^2);
```

Finding K for LQR control method

```
%% Huynh Duy Khoa
%% Finding K for LQR control
clc;
clear;
x1=0;x2=0;x3=0;x4=0;
u=0;

A =  [       0    1.0000         0         0;
         0         0   -7.0071         0;
         0         0         0    1.0000;
 -110.5583         0         0   -0.1545];
B=[0;0;0;2.3792];

Q=[400 0 0 0;
    0 4.2 0 0;
    0 0 110 0;
    0 0 0 0.3];
[Ad,Bd] = c2d(A,B,0.01);

R=0.01;
K=dlqr(Ad,Bd,Q,R)
```

Finding K for Pole placement control method

```
%% Huynh Duy Khoa
clc;

syms k1 k2 k3 k4 s muy omega gtd

K = [k1 k2 k3 k4];

I = [1 0 0 0;
    0 1 0 0;
    0 0 1 0;
    0 0 0 1];

A =  [        0    1.0000         0         0;
         0         0   -7.0071         0;
         0         0         0    1.0000;
 -110.5583         0         0   -0.1545];

B=[0;0;0;2.3792];

C=[1 0 0 0;
    0 1 0 0;
    0 0 1 0;
    0 0 0 1];

D=[0;0;0;0];

s0=A-B*K;

s1=s*I-A+B*K;
s2=det(s1);

p=[-5.6 -0.0386+5.2753i -0.0386-5.2753i -5.12];

K=place(A,B,p)
acl = A-B*K;
                              ecl = eig(acl);
```