Comparative Analysis of Control Methods on a Cart-Inverted Pendulum Model

Cuong Quoc Le[1], Vu Thanh Nguyen[1], Thanh Dat Truong[2], Tan-Hanh Pham[3] and Tri Quang Truong[1*]

[1] HCMC University of Technology and Education, Ho Chi Minh City, Vietnam
[2] Department of Intelligent Robotics, Sungkyunkwan University, South Korea
[2] Department of Mechanical and Civil Engineering, Florida Institute of Technology, Melbourne, FL 32901, USA

[*]E-mail: tri.truongquang@hcmute.edu.vn

Abstract. The inverted pendulum problem remains a fundamental challenge in control theory and serves as a pivotal case study for evaluating various control methods. Beyond validating mathematical model equations and designing controllers, the inverted pendulum provides a conceptual foundation for addressing balance control in robotics. To assess the performance of different controllers, we designed and constructed a mechanical model of the inverted pendulum, enabling the validation of the dynamics model and comparison of control strategies. In this paper, we apply control algorithms, including PID (Proportional-Integral-Derivative Control), LQR (Linear Quadratic Regulator), and MPC (Model Predictive Control), to a Cart-Inverted Pendulum system driven by a gear mechanism. The results indicate that the LQR controller is simple to design and delivers good dynamic performance with reduced oscillations; however, accurately modeling the system presents significant challenges. PID, while effective for simpler control tasks, struggles with oscillations and instability, particularly when tracking nonlinear behaviors. In contrast, the MPC controller excels in handling the nonlinear dynamics of the inverted pendulum and performs effectively across the system's full range, including both swing-up and balancing tasks. Furthermore, the real-time capabilities of CasADi provide a substantial advantage over traditional MPC implementations, ensuring smooth and stable control in real-time environments. These findings demonstrate MPC's superior performance, particularly in managing complex nonlinearities and achieving global control objectives.

1. Introduction

The inverted pendulum system was introduced in the 1960s [1,2] and remains a significant topic in the field of dynamics and control. The system exists in two primary forms: the rotary inverted pendulum and the cart inverted pendulum, both of which present two fundamental control challenges.

The first challenge is the swing-up problem, which involves moving the pendulum from its stable downward position to the unstable upright position. Over the years, various approaches to address this issue have been proposed [3, 4]. The second challenge is maintaining balance at the upright position, which has been shown to provide significant benefits for research purposes.

One of the key advantages of the inverted pendulum system, and a major reason for its popularity in control research, is that stabilizing the pendulum provides an ideal platform for developing and testing more complex models. For example, balance control techniques derived from the inverted pendulum have been successfully applied to self-balancing robots (e.g., Segway), unmanned aerial vehicles (UAVs), and rockets. Moreover, the system has been used to simulate and analyze human movement, particularly in standing [5] and walking [6], and it plays a crucial role in advancing bipedal robot development [7].

It is well understood that not all systems can be controlled using the same control law. The choice of a suitable controller depends on the system's nature—whether it is linear or nonlinear. While the inverted pendulum is fundamentally a nonlinear system, it behaves almost linearly within a small region near the upright position. This characteristic makes it an ideal testbed for experimenting with a wide range of control algorithms, from traditional techniques such as PID and LQR to modern methods like sliding mode control (SMC), model predictive control (MPC), and artificial intelligence-based approaches, including neural networks and reinforcement learning. Consequently, the inverted pendulum is widely used to evaluate controller feasibility before deployment in real-world systems.

Moreover, selecting a suitable controller depends on various factors, such as noise rejection, computational cost, and implementation complexity, making the selection process quite challenging. Consequently, many studies have utilized the inverted pendulum to compare control strategies, including LQR, LQ, and MPC [8], as well as LQR versus SMC variants [9]. These studies highlight significant differences in control strategy characteristics, with each offering distinct strengths and weaknesses.

Given the above, this paper aims to implement and compare PID, LQR, and MPC controllers on a common inverted pendulum model. This approach enables a comprehensive evaluation of the controllers' performance, helping to identify key differences and to assist in choosing the most suitable controller for each application. It also provides deeper insight into the fundamental characteristics of each method, supporting both real world deployment and future research.

The paper is organized as follows: Section 2 presents the system modelling, while Section 3 focuses on the design of PID, LQR, and MPC controllers. Section 4 presents simulation results and experimental validation. Finally, Section 5 provides the conclusions.

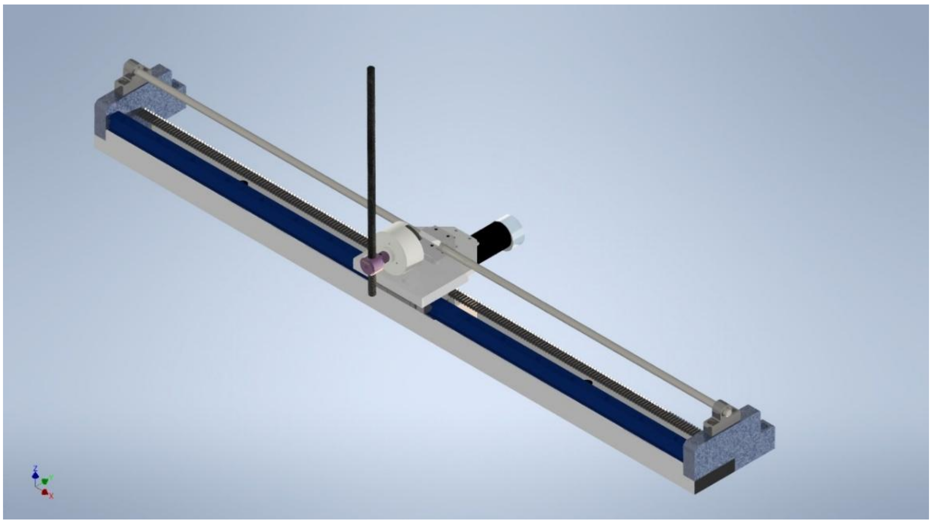

Figure 1. Inverted Pendulum Mechanical Design.

2. Modeling and linearization

2.1 Mathematical modeling of the inverted pendulum system

The inverted pendulum system consists of a cart that moves horizontally and a slender metal rod attached to the cart via a pivot joint at one end. The pendulum is free to swing in a vertical plane, as illustrated in Fig. 2. The motion of the cart is actuated via a torque generated by a DC motor, which changes the cart's position and indirectly influences the pendulum's dynamics.

In this study, the system model is developed using the Euler–Lagrange method [10], which allows a complete description of the interaction between the cart and the pendulum under the influence of control forces. The total energy of the system [11] consists of kinetic energy K and potential energy P, expressed as:

$$K = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m\left[\left(\dot{x} + l\dot{\theta}\cos\theta\right)^2 + \left(l\dot{\theta}\sin\theta\right)^2\right] \tag{1a}$$

$$P = -mgl\cos(\theta) \tag{1b}$$

Considering the effect of the gear train and rotor dynamics on the horizontal motion of the cart, the equivalent mass M is expressed as:

$$M = m_c + m + \frac{I_1}{{r_1}^2} + \frac{I_2}{{r_2}^2} + \frac{I_M}{\left(N_g r_1\right)^2} \tag{2}$$

Where, I_M moment of inertia of the DC motor rotor, I_1, I_2 is moments of inertia of the gear components and N_g is gear ratio of the connected gearbox.

According to [11], the Lagrangian and Euler-Lagrange equation are defined as follows:

$$L = K - P \tag{3a}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i \tag{3b}$$

For $i = 1, 2$ with $(q_1, q_2) = (x, \theta)$, and Q_i representing the generalized forces acting on the system. Substituting equations (1) into (4), we obtain the following nonlinear equations of motion for the inverted pendulum:

$$M\ddot{x} + ml\cos(\theta)\,\ddot{\theta} + b_c\dot{x} - ml\sin(\theta)\,\dot{\theta}^2 = F \tag{4a}$$

$$ml\cos(\theta)\,\ddot{x} + (I + ml^2)\ddot{\theta} + b_p\dot{\theta} + mgl\sin(\theta) = 0 \tag{4b}$$

In practice, the force F applied to the cart is generated by the motor torque and is related to the motor voltage u by the following linear approximation [12]:

$$F = \alpha u - \beta\dot{x} \tag{5}$$

Substituting (5) into ($4a$) leads to a dynamic system where the control input is the motor voltage:

$$M\ddot{x} + ml\cos(\theta)\,\ddot{\theta} + (b_c + \alpha)\dot{x} - ml\sin(\theta)\,\dot{\theta}^2 = \alpha u \tag{6a}$$

$$ml\cos(\theta)\,\ddot{x} + (I + ml^2)\ddot{\theta} + b_p\dot{\theta} + mgl\sin(\theta) = 0 \tag{6b}$$

2.2 System Linearization at a Fixed Operating Point

Since the differential equation (6) illustrates the nonlinear characteristics of the inverted pendulum system [13], to apply linear control methods effectively, we need to linearize the system around the upright equilibrium point ($\theta = \pm\pi$). In this paper, we use the Taylor series approximation method [14] to linearize the system around the equilibrium point.

Let $f(\boldsymbol{x})$ be the vector of state equations for the nonlinear system, where $\boldsymbol{x}$ is the state vector. Then, we obtain the linearized system in the form of equation (7) by evaluating the Jacobian matrix of $f(\boldsymbol{x})$ at the upright equilibrium point.

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} \end{aligned} \tag{7}$$

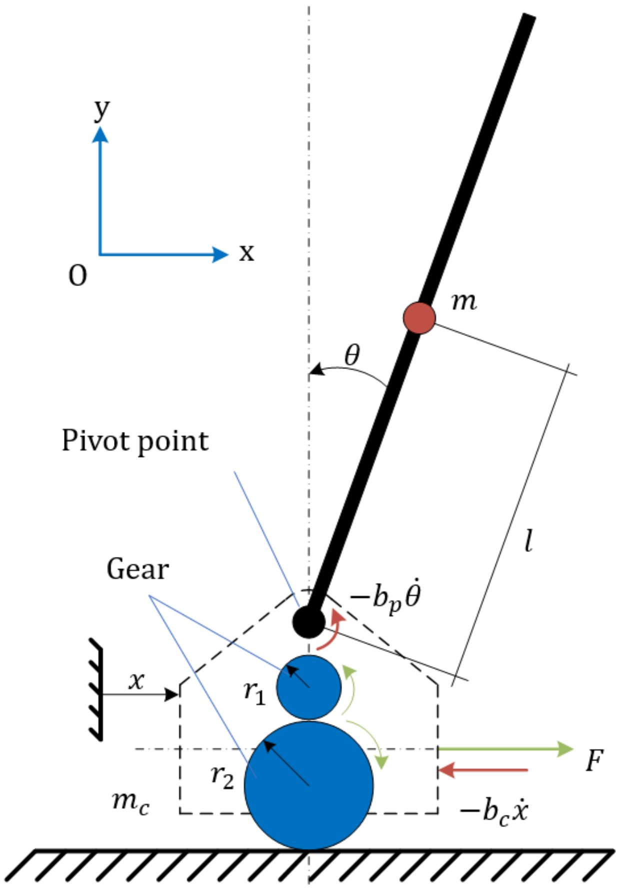

Figure 2. Inverted Pendulum Model.

3. Controller design

3.1. PID (Proportional-Integral-Derivative)

The inverted pendulum system is inherently nonlinear. However, when analyzed around the upright position of the pendulum (an unstable equilibrium point), the system can be linearized, allowing the application of classical linear control methods. As a result, PID control can be effectively applied to control the cart's position and keep the pendulum balanced in the upright position. The control law of the PID controller is defined as:

$$u(t) = K_p e(t) + K_i \int e(t)\,dt + K_d \frac{de(t)}{dt} \quad (8)$$

Where, K_p, K_i and K_d are the proportional, integral, and derivative gains, respectively, $e(t)$ is the error between the desired and actual outputs.

To apply PID effectively, we should derive the transfer function of the system. The transfer function serves as the foundation for analyzing the system's response and designing the PID parameters. Common methods for selecting control parameters include: Ziegler-Nichols, Cohen-Coon, Root Locus, or trial-and-error tuning.

From the linearized model (7) of the system, we transform the system into the Laplace domain [15]:

$$\begin{aligned} s\boldsymbol{X}(s) &= \boldsymbol{A}\boldsymbol{X}(s) + \boldsymbol{B}\boldsymbol{U}(s) \\ \boldsymbol{Y}(s) &= \boldsymbol{C}\boldsymbol{X}(s) + \boldsymbol{D}\boldsymbol{U}(s) \end{aligned} \quad (9)$$

From which the system's transfer function matrix is derived:

$$\boldsymbol{G}(s) = \frac{\boldsymbol{Y}(s)}{\boldsymbol{U}(s)} = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D} \quad (10)$$

The transfer function $\boldsymbol{G}(s)$ serves as the foundation for analyzing system response and designing PID parameters. Several popular tuning methods for PID controllers include Ziegler-Nichols, Cohen-Coon, and Root Locus, or manual tuning through simulation before implementation on the real system.

3.2. LQR (Linear Quadratic Regulator)

The Linear Quadratic Regulator (LQR) is an optimal control method based on full-state feedback. The objective of the LQR controller is to design a control signal that stabilizes the system while minimizing a cost function associated with both the state and control effort. This cost function is commonly known as the Bryson performance index [16] and is defined as:

$$J = \frac{1}{2}\int_0^\infty (\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u})\,dt \quad (11)$$

Where, $\boldsymbol{Q}$ is the state weighting matrix, $\boldsymbol{R}$ is the control input weighting matrix.

The optimal control law is found by solving the above optimization problem, which leads to the Algebraic Riccati Equation (ARE), given by:

$$\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P} + \boldsymbol{Q} = \boldsymbol{0} \quad (12)$$

The goal of solving the ARE is to find the matrix $\boldsymbol{P}$ is obtained, the state feedback gain matrix $\boldsymbol{K}$ can be calculated, resulting in the following control law:

$$\boldsymbol{u} = -\underbrace{\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}}_{\boldsymbol{K}}\boldsymbol{x} \quad (13)$$

Before applying the LQR controller, it is necessary to verify whether the system is controllable. In other words, there must exist a control signal $\boldsymbol{u}$ that can influence all state variables of the system. Controllability can be verified using the condition $(14b)$ involving the controllability matrix, as shown below:

$$\boldsymbol{C} = [\boldsymbol{B} \quad \boldsymbol{A}^1\boldsymbol{B} \quad \boldsymbol{A}^2\boldsymbol{B} \quad \cdots \quad \boldsymbol{A}^{n-1}\boldsymbol{B}] \quad (14a)$$

$$rank(\boldsymbol{C}) = n \quad (14b)$$

In practice, it is often difficult to eliminate the steady-state error using state feedback alone, especially when there are multiple outputs or non-zero reference values. To address this, an integral action can be incorporated into the LQR controller. The idea is to augment the system by introducing an additional state variable $\boldsymbol{z}$, which is the integral of the error between the output and the reference [16]. The augmented system is defined as follows:

$$\frac{d}{dt}\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{z}\end{bmatrix} = \begin{bmatrix}\boldsymbol{Ax}+\boldsymbol{Bu}\\ \boldsymbol{y}-\boldsymbol{r}\end{bmatrix} = \begin{bmatrix}\boldsymbol{Ax}+\boldsymbol{Bu}\\ \boldsymbol{Cx}-\boldsymbol{r}\end{bmatrix} \tag{15}$$

With the augmented state $\widetilde{\boldsymbol{x}} = [\boldsymbol{x} \quad \boldsymbol{z}]^T$, the new cost function for the LQR becomes:

$$J = \frac{1}{2}\int_0^{\infty} (\widetilde{\boldsymbol{x}}^T\boldsymbol{Q}\widetilde{\boldsymbol{x}} + \boldsymbol{u}^T\boldsymbol{Ru})\, dt \tag{16}$$

After solving the Riccati equation corresponding to the augmented system, the updated control law (13) becomes:

$$\boldsymbol{u} = -\boldsymbol{K}_x\boldsymbol{x} - \boldsymbol{K}_z\boldsymbol{z} \tag{17}$$

Where, $\boldsymbol{K}_x$ is the state feedback gain matrix for the original states, $\boldsymbol{K}_z$ is the feedback gain matrix for the integral states. The resulting control law (17)ensures that the system not only remains stable but also eliminates the undesirable steady-state error.

3.3. MPC (Model Predictive Control)

MPC (Model Predictive Control) is an optimal control strategy in which the control input at each time step is computed by solving an optimization problem over a finite prediction horizon. Unlike traditional control methods such as PID or LQR, MPC continuously updates the control signal by solving an optimization problem at each time step, based on the current state and future predictions of the system. The goal of MPC is to minimize a cost function over the prediction horizon N at every sampling instant. Let $\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{x}_{ref}$ be the deviation between the system state and the desired state, then the cost function that MPC seeks to minimize is typically formulated as follows.

$$J = \frac{1}{2}\boldsymbol{e}_N^T\boldsymbol{Q}_N\boldsymbol{e}_N + \frac{1}{2}\sum_{j=0}^{N-1}\left(\boldsymbol{e}_j^T\boldsymbol{Q}_k\boldsymbol{e}_j + \boldsymbol{u}_j^T\boldsymbol{R}_k\boldsymbol{u}_j\right) \tag{18}$$

Moreover, a key advantage of MPC is its ability to control both linear and nonlinear systems. At the same time, it can handle physical constraints of the system (such as state limits $\boldsymbol{x}_{min} \le \boldsymbol{u} \le \boldsymbol{x}_{max}$) and constraints on the control input ($\boldsymbol{u}_{min} \le \boldsymbol{u} \le \boldsymbol{u}_{max}$).

However, MPC operates based on the discrete-time model of the system to predict and optimize future behavior. If the model derived from equation (7) is used, one can utilize MATLAB or various other methods to discretize the system. However, such discretization only provides control within the neighborhood of the linearized operating point. To extend the control range to the full nonlinear system, the fourth-order Runge-Kutta method [14] can be applied. The purpose of this approach is to enable control over the entire nonlinear behavior of the system, covering both the swing-up and balancing phases of the pendulum.

Since MPC requires solving an optimization problem at every time step, it leads to high computational costs and demands powerful hardware, making the basic form of MPC impractical for real-time implementation on embedded systems. To address this issue, some studies have adopted the explicit MPC approach [17]. In this method, the control law is precomputed and stored as a function of the system state, enabling fast real-time retrieval without solving an optimization problem at each step.

$$\boldsymbol{u}_k = \begin{cases} \boldsymbol{F}_1\boldsymbol{x}_k + g_1 & if\ \boldsymbol{H}_1\boldsymbol{x} \le K_1 \\ \vdots & \vdots \\ \boldsymbol{F}_N\boldsymbol{x}_k + g_N & if\ \boldsymbol{H}_N\boldsymbol{x} \le K_N \end{cases} \tag{19}$$

However, explicit MPC also encounters difficulties when the number of control regions increases in complex systems, leading to a significant drop in performance due to the large memory required to store matrices $\boldsymbol{F}$, $\boldsymbol{H}$ and the gain coefficients K. Therefore, in this study, the authors employ CasADi, a powerful optimization library that supports symbolic modeling and the solution of nonlinear programming (NLP) problems. CasADi is a specialized tool for solving real-time optimization problems. As a result, it is well-suited for implementing real-time control of the inverted pendulum system using a nonlinear MPC framework.

4. Simulation and experimental validation

The experimental validation process is performed on a self-built inverted pendulum system with the mechanical design shown in Fig 1. The system includes a DC motor with an encoder, a cart mechanism, and a control board using the STM32F4 Discovery board. UART is used for communication between the PC and the system to record and analyze data. All controllers are computed using MATLAB.

To evaluate the performance of the PID, LQR, and MPC controllers, simulations were conducted in MATLAB using a nonlinear model of the cart-pendulum system. Since the system has two states to be controlled but only one control input, the PID controller in this study was implemented as a parallel dual-loop PID, allowing both states of the inverted pendulum to be regulated. The controller parameters were then tuned using a trial-and-error method to optimize the transient response.

For the LQR controller, MATLAB's built-in lqr() function was used to compute the state feedback gain matrix efficiently. As for the MPC controller, the system was discretized using the fourth-order Runge-Kutta method, and the resulting optimization problem was solved using the CasADi solver integrated within the MATLAB environment.

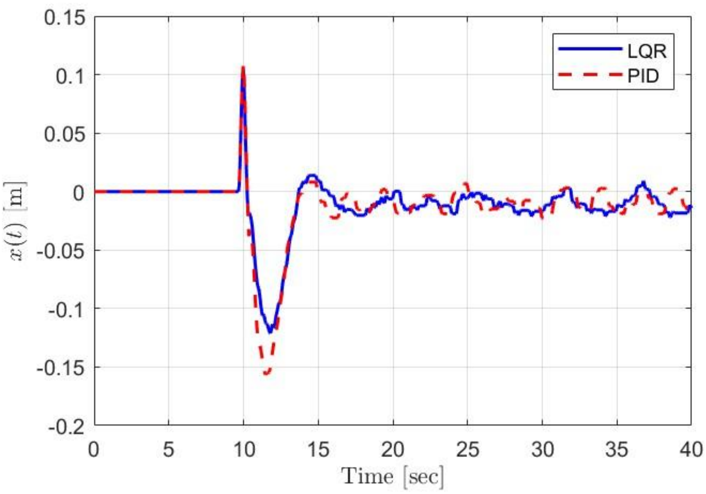

Figure 3. Experimental results of PID and LQR – the cart position when initiating pendulum stabilization.

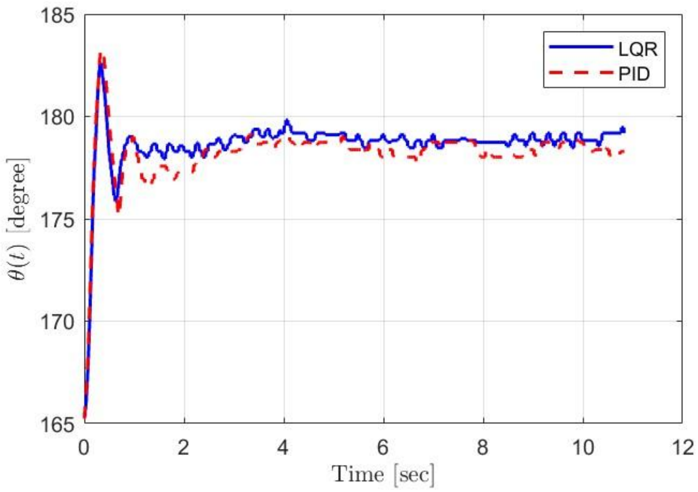

Figure 4. Experimental results of PID and LQR – the pendulum angle values when initiating pendulum stabilization.

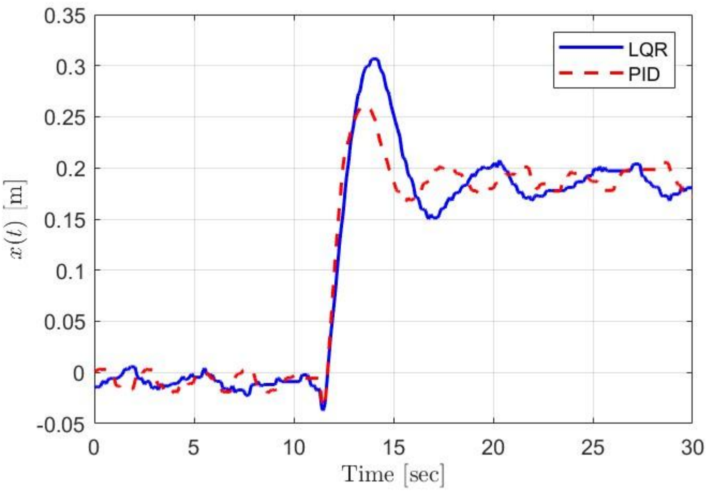

Figure 5. Experimental results of PID and LQR – the cart follows the trajectory $x = 0.2$m.

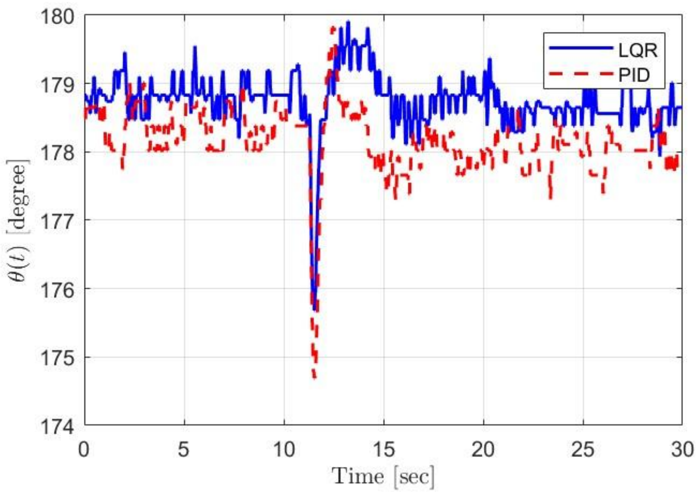

Figure 6. Experimental results of PID and LQR – the pendulum angle value when the cart follows the trajectory $x = 0.2$m.

Based on the results shown in Figure 3 and Figure 4, both LQR and PID controllers are capable of maintaining balance for the inverted pendulum from the initial state. However, LQR exhibits less oscillation, faster response, and greater stability compared to PID. Even when using the controllers to track the trajectory at the cart position $x = 0.2$m in Figure 5 and Figure 6, LQR still demonstrates better control, with less oscillation and quicker stabilization in both position and pendulum angle. Meanwhile, PID experiences strong initial oscillations and slight instability in the pendulum angle. Notably, both PID and LQR controllers are unable to perform the swing-up maneuver, as both are linear controllers that only function within the stable region of the pendulum. In contrast, MPC can control the entire process of the system Figure. 7, including both swing-up and balance, demonstrating superior capability in handling nonlinear conditions and global control. The MPC simulation shows smoother and more stable motion, especially in complex system dynamics.

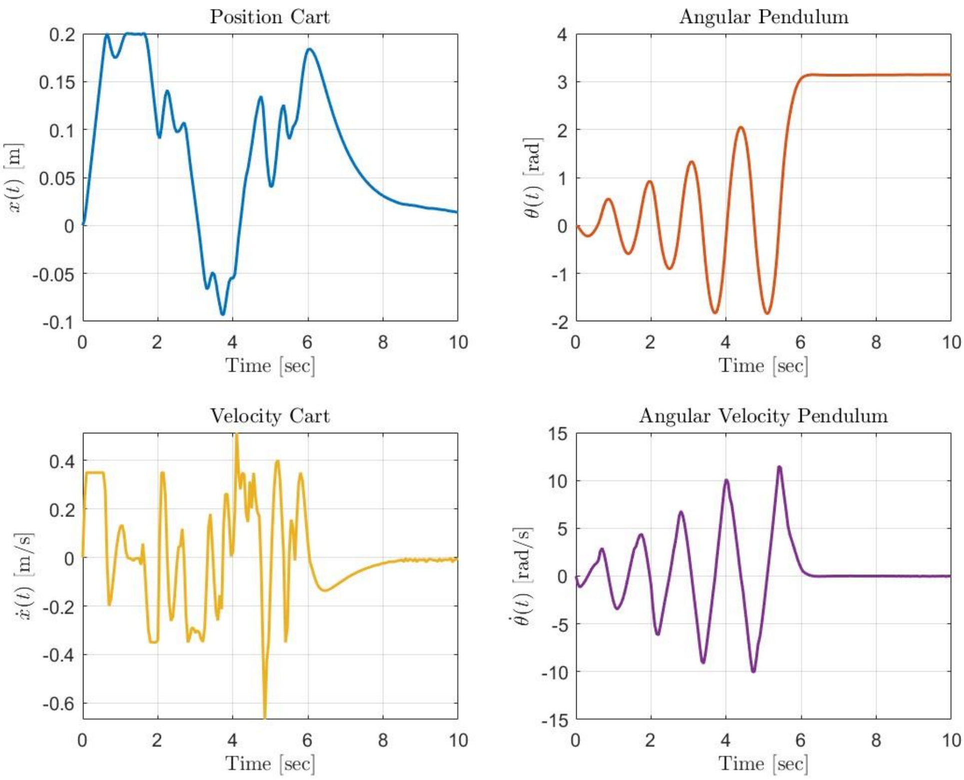

Figure 7. Simulation results of MPC – controlling the entire process including swing-up and balance.

5. Conclusion

This paper presented the design and comparison of three controllers—PID, LQR, and MPC—applied to a common inverted pendulum model. Through simulation and experimental validation, it was demonstrated that each controller has distinct advantages and limitations, depending on specific control objectives and operating conditions.

The PID controller is simple and easy to implement but is most effective for linear or near-linear systems. It performs poorly in noisy environments or under constraints, and tuning PID gains often requires considerable time and expertise. In contrast, the LQR controller offers better control performance for linear systems, featuring reduced oscillations and fast response times. However, both PID and LQR face challenges when applied to real-world systems with constraints. In this context, MPC exhibits superior capability due to its ability to effectively manage output constraints and boundary conditions. Nevertheless, its high computational demand and implementation complexity necessitate powerful hardware.

The results underscore the importance of selecting a suitable controller based on the system's characteristics, performance requirements, and computational cost. The inverted pendulum remains an ideal platform for testing and evaluating diverse control strategies, particularly in the exploration of advanced modern control algorithms.

Looking forward, this research will expand beyond controllers relying on predefined dynamic models. Future work will focus on comparing modern control algorithms that do not depend on mathematical models, especially techniques related to Reinforcement Learning, guiding the development of a truly robust foundation for control systems.

References

[1] Phelps F M and Hunter J H 1965 *An Analytical Solution of the Inverted Pendulum* Am. J. Phys. **33** 285.

[2] Blitzer L 1965 *Inverted Pendulum* Am. J. Phys. **33** 1076–1078.

[3] Furuta K, Yamakita M and Kobayashi S 1991 *Swing Up Control of Inverted Pendulum* IECON '91 2193.

[4] Wiklund M, Kristenson A, Astrom K J 1993 *A new strategy for swing up an inverted pendulum*. IFAC Proceedings Volumes **26** 757–760.

[5] Gage W H, Winter D A, Frank J S, Adkin A L 2004 *Kinematic and kinetic validity of the inverted pendulum model in quiet standing*. Gait Posture **19** 124–132.

[6] Morasso P, Cherif A and Zenzeri J 2019 *Quiet standing: The Single Inverted Pendulum model is not so bad after all*. PLoS ONE 14 e0213870.

[7] Kuo A D 2007 *The six determinants of gait and the inverted pendulum analogy: A dynamic walking perspective*. Human Movement Science **26** 617–656.

[8] Bakarac P, Klauco M, Fikar M 2018 *Comparison of inverted pendulum stabilization with PID, LQ, and MPC control*. Proceedings of the 29th International Conference on Cybernetics & Informatics (K&I), Lazy pod Makytou, Slovakia, 31 January–3 February 2018, pp. 1–6.

[9] Irfan S, Mehmood A, Razzaq M T and Iqbal J 2018 *Advanced sliding mode control techniques for inverted pendulum: Modelling and simulation*. Engineering Science and Technology **21** 753–759.

[10] Nguyen V K 2007 *Động lực học hệ nhiều vật* (Hà Nội: Nhà xuất bản Khoa học và Kỹ thuật).

[11] Hibbeler R C 2024 *Engineering mechanics dynamics*, 15th ed (Harlow: Pearson Education Limited).

[12] Nakamura M, Goto S and Kyura N 2004 *Mechatronic Servo System Control: Problems in Industries and Their Theoretical Solutions*, transl. from Japanese by Zhang T (Berlin: Springer-Verlag).

[13] Niku S B 2020 *Introduction to Robotics: Analysis, Control, Applications*, 3rd ed. (Hoboken, NJ: John Wiley & Sons)

[14] Burden R L and Faires J D 2011 *Numerical Analysis*, 9th ed. (Boston, MA: Brooks/Cole, Cengage Learning)

[15] O'Neil P V 2012 *Advanced engineerging mathematics*, 7th ed. (Stamford, CT: Cengage Learning).

[16] Ogata K 2010 *Modern Control Engineering*, 5th ed. (Upper Saddle River, NJ: Prentice Hall).

[17] Bemporad A 2021 *Explicit Model Predictive Control*. In J. Baillieul & T. Samad (Eds.), *Encyclopedia of Systems and Control* (pp. 744–751). Springer