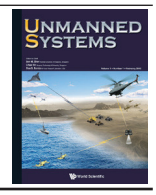
Unmanned Systems, Vol. 14, No. 3 (2026) 573–587

DOI: 10.1142/S2301385026410074

UNMANNED SYSTEMS
https://www.worldscientific.com/worldscinet/us



Cart-Pole System: Practical Deployment of Model-Based Control Versus Reinforcement Learning

Qihua Chen [*,†], Mingxiang Liu [*,†,§], Minyue Fu [‡,¶]

[*]*School of Automation and Intelligent Manufacturing, Southern University of Science and Technology, Shenzhen, P. R. China*

[†]*Guangdong Provincial Key Laboratory of Fully Actuated System Control Theory and Technology, Shenzhen, P. R. China*

[‡]*School of Automation, Guangdong University of Technology, Guangzhou, P. R. China*

This paper compares the classical model-based approach and reinforcement learning (RL) approach for nonlinear systems with known structures but unknown parameters, which represents a wide class of engineered control systems. The study is based on a benchmark cart-pole system involving both swing-up and balance control. Although the model-based approach for the cart-pole system has been studied extensively in the literature, we propose a new pseudo-energy function with a logarithmic barrier function (LBF) for smoother swing-up action. A simple feedforward compensation is used to reduce the cart friction. The subsequent balance control employs a linear quadratic regulator. The model-based approach is complemented with a real-time parameter identification scheme. For the RL approach, we adopt the deep deterministic policy gradient (DDPG) algorithm in conjunction with several simulation techniques to mitigate the simulation-to-reality (sim-to-real) gap. Experimental results on the Googoltech platform demonstrate that both design approaches are effective. The model-based approach does not require training and has higher computational efficiency, but it requires substantial system modeling and controller design knowledge. In contrast, the RL approach has higher flexibility and adaptability, but requires massive training. The results reflect the trade-offs and complementarity of the two methods in practical control tasks.

Keywords: Model-based control; reinforcement learning control; cart-pole system.

US

1. Introduction

The cart-pole system is a classical benchmark in control engineering, characterized by instability, nonlinearity, underactuation and nonminimum phase properties [1]. It consists of a pole attached to a cart moving on a horizontal slider with the goals of swinging the pole up and balancing it upright. Real-world factors such as friction, sensor noise, actuator saturation and slider limits make it ideal for testing control strategies' effectiveness.

Beyond its academic significance, the cart-pole problem embodies essential dynamic features found in numerous practical engineering systems. In robotics, analogous dynamics appear in the study of bipedal walking and balance control mechanisms [2]. In aerospace engineering, the pitch control of thrust-vectored rockets closely resembles the cart-pole dynamics [3]. In addition, transportation systems involving self-balancing vehicles utilize control strategies analogous to those studied within this framework [4,5]. Hence, advances in controlling the cart-pole system not only contribute to theoretical development but also have direct practical implications for engineering disciplines.

Historically, extensive research has been devoted to addressing the control challenges of the cart-pole system through a variety of model-based methodologies. Swing-up control is often tackled using energy-based approaches,

Received 15 October 2025; Accepted 6 January 2026; Published 6 February 2026. This paper was recommended for publication in its revised form by Special Issues Editors, Jie Chen, Ben M Chen and Zongli Lin.
Email Addresses: [§]*liumx@ieee.org*, [¶]*fumy@sustech.edu.cn*
[§]Corresponding author.

which provide intuitive physical insights and have proven effective in practice [6,7]. For stabilization, both classical linear controllers such as proportional integral derivative (PID) [8], the pole placement and LQR [9], as well as advanced nonlinear techniques such as sliding mode control [10–12] and fuzzy logic control [13–15], have been widely explored. However, real-world implementation of these model-based strategies typically requires accurate knowledge of the system parameters. In particular, unmodeled static and kinetic friction in the cart mechanism can significantly affect controller performance or even destabilize the system [16]. Furthermore, the limited length of the slider restricts the range of motion of the cart, which can result in control failure when the cart reaches physical boundaries [17]. To enhance the practicality of these control techniques, this work explicitly incorporates friction compensation strategies, parameter identification via least-squares estimation and introduces logarithmic barrier functions (LBFs) to ensure the cart's position remains safely within operational limits.

Despite the effectiveness of model-based methods, their dependence on precise models limits their robustness to uncertainties. In recent years, RL has emerged as a compelling alternative approach capable of synthesizing controllers without explicit reliance on a detailed system model [18–21]. Among RL methods, the DDPG algorithm is particularly promising due to its effectiveness in handling continuous-state and continuous-action control tasks [22, 23], including the cart-pole system [24]. However, deploying RL methods directly on physical systems introduces substantial challenges, including long training durations, hardware wear and the nonstationarity of the physical system during prolonged learning, thus violating the RL assumption of a time-invariant environment [25]. In addition, there is a performance discrepancy between simulation training environments and real-world applications, known as the ''sim-to-real gap'' [26–28]. This work addresses these difficulties by employing specialized sim-to-real techniques to enable policy transfer from simulation to real hardware.

The primary objective of this paper is to present a comparative evaluation of two distinct control paradigms: (i) a model-based strategy that employs an initial phase of friction compensation and parameter identification, followed by the application of an improved energy-based method and LQR stabilization and (ii) a RL approach utilizing the DDPG algorithm in conjunction with sim-to-real transfer techniques. Both methodologies are rigorously validated through experiments conducted on a physical cart-pole system, enabling an assessment of their performance under realistic operating conditions. The main contributions of this work are summarized as follows:

First, a friction modeling and compensation strategy is implemented based on a parameter identification approach. By combining transfer function estimation with time-domain curve fitting, the Coulomb and viscous friction coefficients are estimated using least squares. The resulting parameters are used in a feedforward compensator to minimize the effect of friction. Second, we extend previous work by implementing a system identification framework on a real cart-pole platform, using the partial state variable filtering technique proposed in [7]. Physical parameters, i.e. $M + m$, ml and $J + ml^2$ of continuous-time dynamics, are identified from real-world sensor data under measurement noise. Third, to prevent boundary violations during swing-up on limited guide rail, a penalization term based on LBFs is incorporated into the energy-based control law. This modification improves swing-up safety in physical systems by enforcing soft constraints without sacrificing convergence. Fourth, a task-decoupled reinforcement learning (RL) framework is developed, where swing-up and balancing are treated as separate subtasks, each handled by an individual DDPG agent. A zero-shot sim-to-real transfer strategy is integrated, involving system identification, domain randomization, sensor noise modeling and subtask-wise training, enabling deployment on real hardware without additional tuning. Last, to enable a comparison between model-based and learning-based control, a consistent experimental setup is implemented on a real cart-pole system. Both strategies are tested under identical conditions to evaluate performance and implementation cost, providing actionable insights for control strategy selection.

The rest of this paper is organized as follows. Section 2 presents the description and mathematical modeling of the cart-pole system. Section 3 describes the model-based control method, including friction modeling, parameter estimation and swing-up/balancing control design enhanced by LBFs. Section 4 introduces the RL approach using DDPG and discusses sim-to-real transfer techniques. Experimental results comparing both approaches on a real cart-pole platform are shown in Sec. 5. Finally, conclusions are drawn in Sec. 6.

2. System Modeling and Identification

2.1. *Dynamic modeling of the cart-pole system*

The cart-pole system, as shown in Fig. 1, can be analyzed by neglecting air resistance and pole joint friction. We assume that both the cart and the pole are rigid bodies and that the pole has a uniform distribution of mass. All the parameters can be seen in Table 1. We can obtain the mathematical model of the system using the Euler–Lagrange equations, which are given by

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right) - \frac{\partial T}{\partial q} + \frac{\partial V}{\partial q} = F, \tag{1}$$

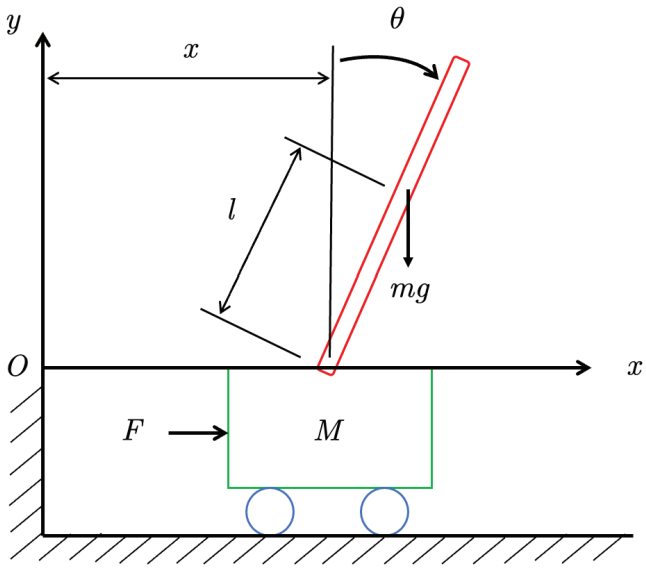


Fig. 1. The cart-pole system.

Table 1. Symbols and their physical interpretations.

Symbol	Unit	Physical interpretations
M	kg	Mass of the cart
m	kg	Mass of the pole
l	m	Distance from the pivot to the pole's center of mass
L	m	Half length of the slider
J	kg· m^2	Moment of inertia of the pole
g	m/s^2	Acceleration of gravity
F	N	Input force applied to the cart
x	m	Displacement of the cart
$\dot{x}$	m/s	Velocity of the cart
θ	rad	Angle of the pole
$\dot{\theta}$	rad/s	Angular velocity of the pole

where q is the generalized coordinate of the system, F is the generalized force, T is the kinetic energy and V is the potential energy. For the cart-pole system, the displacement x of the cart and the angle θ of the pole are chosen as q, and the input force is F. The potential energy V is given by

$$V = mgl\cos\theta. \tag{2}$$

The kinetic energy contains the translational and rotational kinetic energy of both the cart and the pole, which is given by

$$\begin{aligned} T &= \frac{1}{2}M\dot{x}^2 + \frac{1}{2}J\dot{\theta}^2 \\ &\quad + \frac{1}{2}m\left[\left(\frac{d(x + l\sin\theta)}{dt}\right)^2 + \left(\frac{d(l\cos\theta)}{dt}\right)^2\right] \\ &= \frac{1}{2}(M+m)\dot{x}^2 + \frac{1}{2}(J+ml^2)\dot{\theta}^2 + ml\dot{\theta}\dot{x}\cos\theta, \end{aligned} \tag{3}$$

where J is the moment of inertia of the pole about its center of mass.

Substituting Eqs. (2) and (3) into Eq. (1), the nonlinear dynamics of the cart-pole system can be obtained as

$$\begin{bmatrix} M+m & ml\cos\theta \\ ml\cos\theta & J+ml^2 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} F + ml\dot{\theta}^2\sin\theta \\ mgl\sin\theta \end{bmatrix}, \tag{4}$$

where the LHS represents the mass-inertia coupling of the cart-pole, and the RHS represents the input and gravitational terms.

2.2. *Friction compensation*

In this section, we address the problem posed by cart friction on system parameter identification and control performance. To address these problems, we propose a feedforward compensation method. This method focuses on the compensation of cart friction while ignoring joint friction. Using state variables such as position or velocity, a friction model is developed to estimate the friction torque. This estimated torque is then incorporated into the control input to effectively counteract friction. The structure of the proposed friction compensation method is illustrated in Fig. 2.

The friction model used in this work combines Coulomb friction and viscous friction, represented mathematically as

$$F_f = \begin{cases} F_{\text{in}} & \text{if } \dot{x} = 0 \text{ and } |F_{\text{in}}| \le F_c, \\ k\dot{x} + F_c \cdot \text{sign}(\dot{x}) & \text{otherwise}, \end{cases} \tag{5}$$

where $\dot{x}$ is the velocity of the cart, F_{in} is the input force and F_c and k represent Coulomb friction and the viscous coefficient, respectively.

When the pole remains stationary relative to the cart, after ignoring the motion of the pole and applying a constant input, the system behaves as a cart with a total mass $M+m$, governed by the dynamics.

$$F_{\text{in}} - F_f = F_{\text{in}} - (F_c + k\dot{x}) = (M+m)\ddot{x}. \tag{6}$$

To identify F_f, we define $F = F_{\text{in}} - F_c$, resulting in

$$F = k\dot{x} + (M+m)\ddot{x}. \tag{7}$$

The transfer function between F and x is given as

$$\frac{X(s)}{F(s)} = \frac{\frac{1}{M+m}}{s^2 + \frac{k}{M+m}s} \tag{8}$$

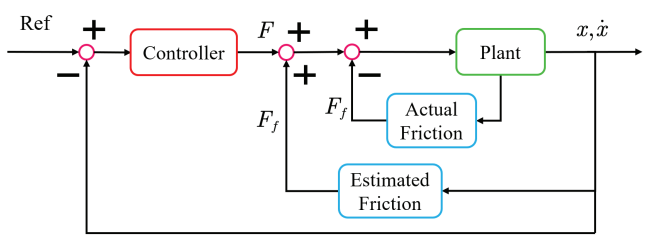


Fig. 2. The friction compensation.

and using the input force F_{in} and the response of the cart position, we can identify the parameters of the transfer function:

$$p_1 = \frac{F_{\text{in}} - F_c}{F_{\text{in}}(M+m)}, \quad p_2 = \frac{k}{M+m}. \tag{9}$$

Additionally, we can solve the time-domain expression of $x(t)$ as

$$x(t) = (F_{\text{in}} - F_c)\left(\frac{1}{k}t - \frac{M+m}{k^2} + \frac{M+m}{k^2}e^{-\frac{k}{M+m}t}\right) \tag{10}$$

and by applying a curve fitting method to the $x(t)$ response, we can identify

$$p_3 = \frac{F_{\text{in}} - F_c}{k}, \quad p_4 = \frac{k}{M+m}. \tag{11}$$

Combining the two approaches, we can verify the consistency of the identified parameters. However, knowing (p_1, p_2) or (p_3, p_4) alone does not directly yield the values of k and F_c. To address this problem, we consider the least squares method.

Given a set of input forces $\{F_{\text{in}}^i\}_{i=1}^N$, and using the above identification, we can obtain a corresponding set of parameters $\{p_1^i\}_{i=1}^N$ and $\{p_2^i\}_{i=1}^N$. The linear equations corresponding to each p_1^i, p_2^i, and the parameter vector $k_p = [M + m, F_c, k]^\top$ are given by

$$\Theta^i k_p = \Phi^i, \tag{12}$$

where

$$\Theta^i = \begin{bmatrix} p_1^i F_{\text{in}}^i & 1 & 0 \\ p_2^i & 0 & -1 \end{bmatrix}, \quad \Phi^i = \begin{bmatrix} F_{\text{in}}^i \\ 0 \end{bmatrix}. \tag{13}$$

Then, Eq. (12) can be solved using the least squares method:

$$k_p = ((\Theta^i)^\top \Theta^i)^{-1}(\Theta^i)^\top \Phi^i. \tag{14}$$

This procedure can also be applied using the parameter group (p_3, p_4), which can be compared against (p_1, p_2) to further validate the identified parameters.

2.3. *Parameter identification*

In this section, we focus on the parameter identification of the cart-pole system. Our primary objective is to estimate the vector of system parameters $\xi = [M + m, ml, J + ml^2]^\top$, using measurements of state $X = [x, \dot{x}, \theta, \dot{\theta}]^\top$ and input force F. Since the equation is linear in ξ, we employ the least squares method to estimate the system parameters. To mitigate the impact of measurement noise, we adopt the partial state variable filtering technique developed in [7], which enables the construction of a noise-robust regression model. Consider the dynamic equation:

$$A(\theta)\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = b(\theta, \dot{\theta}), \tag{15}$$

where

$$A(\theta) = \begin{bmatrix} M+m & ml\cos\theta \\ ml\cos\theta & J+ml^2 \end{bmatrix}, \quad b(\theta, \dot{\theta}) = \begin{bmatrix} F + ml\dot{\theta}^2\sin\theta \\ mgl\sin\theta \end{bmatrix}.$$

Applying the transfer function of a first-order low-pass filter $G(s) = \frac{1}{s+\lambda}$ to both sides of Eq. (15), we define the filtered output as

$$\zeta = \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix} = G(s)A(\theta)\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix}. \tag{16}$$

Then for any $t_0 < t_1$, we obtain

$$\zeta(t_1) = e^{-\lambda(t_1-t_0)}\zeta(t_0) + \int_{t_0}^{t_1} e^{-\lambda(t_1-\tau)}A(\theta(\tau))\begin{bmatrix} \ddot{x}(\tau) \\ \ddot{\theta}(\tau) \end{bmatrix} \mathrm{d}\tau. \tag{17}$$

Letting $\zeta(t_0) = 0$ and integrating over $[t_0, t_1]$, we have

$$\begin{aligned} \zeta(t_1) = A(\theta(t_1))\begin{bmatrix} \dot{x}(t_1) \\ \dot{\theta}(t_1) \end{bmatrix} - e^{-\lambda(t_1-t_0)}A(\theta(t_0))\begin{bmatrix} \dot{x}(t_0) \\ \dot{\theta}(t_0) \end{bmatrix} \\ - \int_{t_0}^{t_1} e^{-\lambda(t_1-\tau)}c(\theta(\tau), \dot{x}(\tau), \dot{\theta}(\tau))\mathrm{d}\tau \end{aligned} \tag{18}$$

with

$$c(\theta, \dot{x}, \dot{\theta}) = \left(\lambda A(\theta) + \frac{\mathrm{d}A(\theta)}{\mathrm{d}t}\right)\begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix}.$$

Similarly, applying $G(s)$ to the RHS of Eq. (15), we finally obtain

$$\begin{aligned} A(\theta(t_1))\begin{bmatrix} \dot{x}(t_1) \\ \dot{\theta}(t_1) \end{bmatrix} - e^{-\lambda(t_1-t_0)}A(\theta(t_0))\begin{bmatrix} \dot{x}(t_0) \\ \dot{\theta}(t_0) \end{bmatrix} \\ - \int_{t_0}^{t_1} e^{-\lambda(t_1-\tau)}z(\theta(\tau), \dot{x}(\tau), \dot{\theta}(\tau))\mathrm{d}\tau = 0, \end{aligned} \tag{19}$$

where

$$z(\theta, \dot{x}, \dot{\theta}) = b(\theta, \dot{\theta}) + \left(\lambda A(\theta) + \frac{\mathrm{d}A(\theta)}{\mathrm{d}t}\right)\begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix}.$$

Since $\xi = [M + m, ml, J + ml^2]^\top$ are the parameters to be identified, we rewrite relevant expressions as

$$A(\theta)\begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} = \mathcal{A}\xi, \quad z(\theta, \dot{x}, \dot{\theta}) = \mathcal{B}\xi + \mathcal{C}, \tag{20}$$

where

$$\mathcal{A} = \begin{bmatrix} \dot{x} & \dot{\theta}\cos\theta & 0 \\ 0 & \dot{x}\cos\theta & \dot{\theta} \end{bmatrix}, \quad \mathcal{C} = \begin{bmatrix} F \\ 0 \end{bmatrix},$$
$$\mathcal{B} = \begin{bmatrix} \lambda\dot{x} & \lambda\dot{\theta}\cos\theta & 0 \\ 0 & \lambda\dot{x}\cos\theta + g\sin\theta - \dot{x}\dot{\theta}\sin\theta & \lambda\dot{\theta} \end{bmatrix}.$$

Then Eq. (19) can be written as

$$\left(\mathcal{A}(t_1) - e^{-\lambda(t_1-t_0)}\mathcal{A}(t_0) - \int_{t_0}^{t_1} e^{-\lambda(t_1-\tau)}\mathcal{B}(\tau)d\tau\right)\xi = \int_{t_0}^{t_1} e^{-\lambda(t_1-\tau)}\mathcal{C}(\tau)d\tau. \tag{21}$$

For discrete measurements at $t = kh, k = 0, 1, \ldots, K$, i.e. $X(kh)$ is measurable, Eq. (21) becomes

$$\left(\mathcal{A}(k) - e^{-\lambda kh}\mathcal{A}(0) - \frac{e^{\lambda h}-1}{\lambda}\sum_{i=0}^{k-1} e^{-\lambda h(k-1-i)}\mathcal{B}(i)\right)\xi = \frac{e^{\lambda h}-1}{\lambda}\sum_{i=0}^{k-1} e^{-\lambda h(k-1-i)}\mathcal{C}(i). \tag{22}$$

Since Eq. (22) is linear in ξ, the least-squares optimization is as follows:

$$H_k\xi = F_k, \tag{23}$$

where

$$H(k) := \mathcal{A}(k) - e^{-\lambda kh}\mathcal{A}(0) - \frac{e^{\lambda h}-1}{\lambda}\sum_{i=0}^{k-1} e^{-\lambda h(k-1-i)}\mathcal{B}(i),$$
$$F(k) := \frac{e^{\lambda h}-1}{\lambda}\sum_{i=0}^{k-1} e^{-\lambda h(k-1-i)}\mathcal{C}(i) \tag{24}$$

and

$$H_k = \begin{bmatrix} H(1) \\ H(2) \\ \vdots \\ H(K) \end{bmatrix}, \quad F_k = \begin{bmatrix} F(1) \\ F(2) \\ \vdots \\ F(K) \end{bmatrix}. \tag{25}$$

3. Model-Based Control Design

The model-based control design for the physical cart-pole system consists of two main stages: swing-up and balancing. This framework is implemented in conjunction with system identification and friction compensation to achieve real-time control. The general process is illustrated in Fig. 3, which describes the key components of the control architecture.

The following sections describe the swing-up control using an energy-based approach enhanced with LBFs, and the balancing control using an LQR.

3.1. *Improved swing-up control with LBFs*

The goal of swing-up control is to move the pole from its initial downward position ($\theta = \pi$) to the upright position ($\theta = 0$) while ensuring that the angular velocity $\dot{\theta}$ approaches zero. Starting from Eq. (4), the angular acceleration $\ddot{\theta}$ is derived as

$$\ddot{\theta} = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} M+m & ml\cos\theta \\ ml\cos\theta & J+ml^2 \end{bmatrix}^{-1}\begin{bmatrix} F + ml\dot{\theta}^2\sin\theta \\ mgl\sin\theta \end{bmatrix} = \frac{1}{\Delta}\left[mgl\sin\theta - \frac{ml}{M+m}\cos\theta(F + ml\dot{\theta}^2\sin\theta)\right], \tag{26}$$

where

$$\Delta = J + ml^2 - \frac{m^2l^2\cos^2\theta}{M+m}. \tag{27}$$

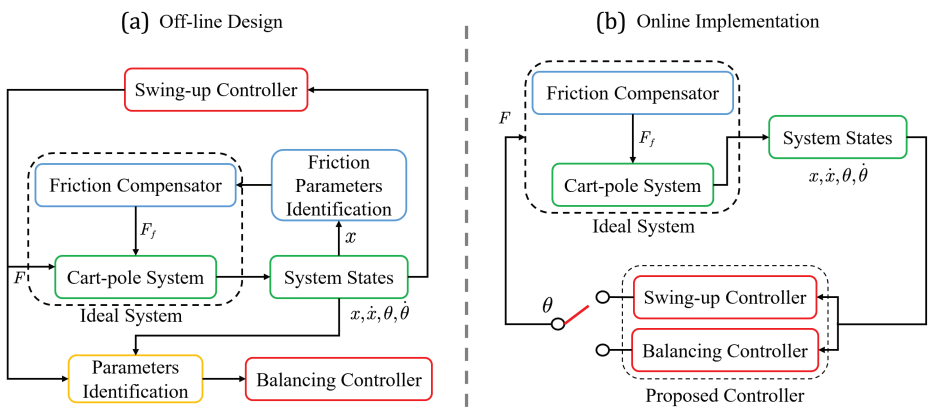


Fig. 3. The block diagram of model-based control design.

To swing up the pole, an energy control strategy is proposed by [7] which will be summarized below. Note that Δ in Eq. (27) can be regarded as the pseudo-moment of inertia. Hence, a pseudo-energy function is defined as

$$E_p = \frac{1}{2}\Delta\dot{\theta}^2 + mgl(\cos\theta - 1). \tag{28}$$

Differentiating E_p with respect to time yields the following.

$$\begin{aligned}\frac{dE_p}{dt} &= \frac{1}{2}\frac{d\Delta}{dt}\dot{\theta}^2 + \Delta\dot{\theta}\ddot{\theta} - mgl\dot{\theta}\sin\theta \\ &= -\frac{mlF}{M+m}\dot{\theta}\cos\theta. \end{aligned}\tag{29}$$

In the initial state ($\theta = \pi$, $\dot{\theta} = 0$), the pseudo-energy is $E_p = -2mgl$. When the pole reaches the upright position with zero angular velocity ($\theta = 0, \dot{\theta} = 0$), the pseudo-energy becomes $E_p = 0$. Therefore, to swing up the pole, E_p must increase from $-2mgl$ to 0. By designing the control input F such that $\frac{dE_p}{dt} > 0$, we ensure that the energy increases, enabling the pole to swing up.

A basic control law to achieve this is given by

$$F = \begin{cases} -\text{sgn}(\dot{\theta}\cos\theta)F_{\text{max}} & \text{if } E_p < 0, \\ 0 & \text{if } E_p = 0, \end{cases} \tag{30}$$

where F_{max} is the maximum control force allowed.

Remark 1. The proof of the effectiveness of (30)is guaranteed by [7]. It is derived from the new pseudo-energy function (28) which considers the full cart pole dynamics rather than the kinetic and potential energy $E = \frac{1}{2}J\dot{\theta} + mgl(\cos\theta - 1)$ of a simplified single pole of [6]. This yields a control law that is more physically accurate and easier to achieve in the hardware. Furthermore, the control input in [6] involves the calculation of the energy E and the adjustment of the gain, while the constant F_{max} is the only variable to adjust in (30).

However, in practical applications, limitations such as the finite length of the slider must be considered. To prevent the cart from reaching the slider boundaries and to penalize excessive velocities, we introduce modifications to the energy control strategy using LBFs.

The improved control input includes additional terms to penalize the cart's position and velocity when they approach their maximum allowable values:

$$F_x = k_x\,\text{sgn}(x)\log\left(1 - \frac{|x|}{x_{\text{max}}}\right), \tag{31}$$

$$F_\nu = k_\nu\,\text{sgn}(\nu)\log\left(1 - \frac{|\nu|}{\nu_{\text{max}}}\right), \tag{32}$$

where x_{max} and ν_{max} are the maximum allowable position and velocity of the cart, respectively, and k_x, k_ν are coefficients determining the strength of the penalties.

Combining the original energy control law with these additional terms, the improved control input becomes

$$F = \begin{cases} -\text{sgn}(\dot{\theta}\cos\theta)F_{\text{max}} + F_x + F_\nu & \text{if } E_p < 0, \\ 0 & \text{if } E_p = 0. \end{cases} \tag{33}$$

Remark 2. Equation (33) does not involve any system parameters, which implies that no prior training is required. The unknown parameters can be identified during the swing-up phase. This is extremely valuable for applications where experiments are costly or failed experiments are not acceptable.

This improved energy control strategy not only ensures the pole swings up by increasing the pseudo-energy but also prevents the cart from exceeding its physical limits by incorporating barrier functions. The additional terms F_x and F_ν act as penalties that become significant as the cart approaches the slider boundaries or reaches high velocities, thus maintaining safe operation within the constraints of the system.

3.2. *Balancing control with LQR*

When the pole is near the upright position (θ and $\dot{\theta}$ are close to zero), we have $\sin\theta \approx \theta$, $\cos\theta \approx 1$, $\dot{\theta}^2\sin\theta \approx 0$. Thus, Eq. (4) can be linearized as follows:

$$\begin{bmatrix} M+m & ml \\ ml & J+ml^2 \end{bmatrix}\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} \approx \begin{bmatrix} F \\ mgl\theta \end{bmatrix}. \tag{34}$$

The state space representation of the linearized system is: $\dot{X} = AX + Bu$, where the state vector $X = [x, \dot{x}, \theta, \dot{\theta}]^\top$, and the matrices A and B are given by

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\dfrac{m^2l^2g}{\Xi} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \dfrac{(M+m)mgl}{\Xi} & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \dfrac{J+ml^2}{\Xi} \\ 0 \\ -\dfrac{ml}{\Xi} \end{bmatrix},$$

where $\Xi = (J+ml^2)(M+m) - m^2l^2$. The control objective is to stabilize the system at $\theta = 0$ and $\dot{\theta} = 0$. To achieve this, the LQR approach is used [29]. The LQR minimizes the cost function $\mathcal{J} = \int_0^\infty (X^\top QX + u^\top Ru)dt$, where Q is a positive semidefinite matrix and R is positive definite. The state feedback control law is $u = F = -KX$, where $K = R^{-1}B^\top P$, and the matrix P is the unique positive semidefinite solution to the algebraic Riccati equation $Q + A^\top P + PA - PBR^{-1}B^\top P = 0$.

4. RL-Based Control Design

While the previous sections introduced a model-based control strategy, we now turn to an RL approach to achieve the control objective. Among various RL algorithms, the DDPG is particularly suitable for environments with continuous state and action spaces. Thus, we employ the DDPG algorithm to learn the controller. The entire policy training and deployment process is shown in Fig. 4.

4.1. *DDPG algorithm*

The DDPG algorithm is a model-free, off-policy RL method designed for continuous action control [22]. It combines the deterministic policy gradient (DPG) framework with deep neural networks to approximate both policy and value functions. DDPG adopts an actor-critic architecture: the actor $\mu(s|\theta^\mu)$ maps states to deterministic actions, while the critic $Q(s,a|\theta^Q)$ estimates the action-value function.

The critic is trained by minimizing the temporal difference (TD) error:

$$L(\theta^Q) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}[(y - Q(s,a|\theta^Q))^2], \tag{35}$$

where the target value is defined as

$$y = r + \gamma Q'(s', \mu'(s'|\theta^{\mu'})|\theta^{Q'}). \tag{36}$$

Here, Q' and μ' are target networks that are updated using a soft update rule:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}, \quad \theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}. \tag{37}$$

The actor is trained via the DPG:

$$\nabla_{\theta^\mu} J \approx \mathbb{E}_{s\sim\mathcal{D}}[\nabla_a Q(s,a|\theta^Q)\big|_{a=\mu(s)} \cdot \nabla_{\theta^\mu}\mu(s|\theta^\mu)]. \tag{38}$$

To encourage exploration during training, temporally correlated noise (e.g. Ornstein–Uhlenbeck process) is added to the actions.

$$a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t. \tag{39}$$

Experience replay is used to store transitions (s,a,r,s') in a buffer $\mathcal{D}$, allowing efficient and decorrelated learning.

4.2. *DDPG framework for swing-up and balancing*

In physical experiments, real-time training encounters several significant challenges.

- The system has safety mechanisms; if the cart reaches the slider boundary, the control is stopped.
- Each episode must begin with the cart centered and the pole pointing downward, which is difficult to repeat manually.
- Large exploratory actions may result in unsafe behaviors or high wear on hardware components.

To address these constraints, we train the DDPG agent in simulation and deploy it directly to the real-world system.

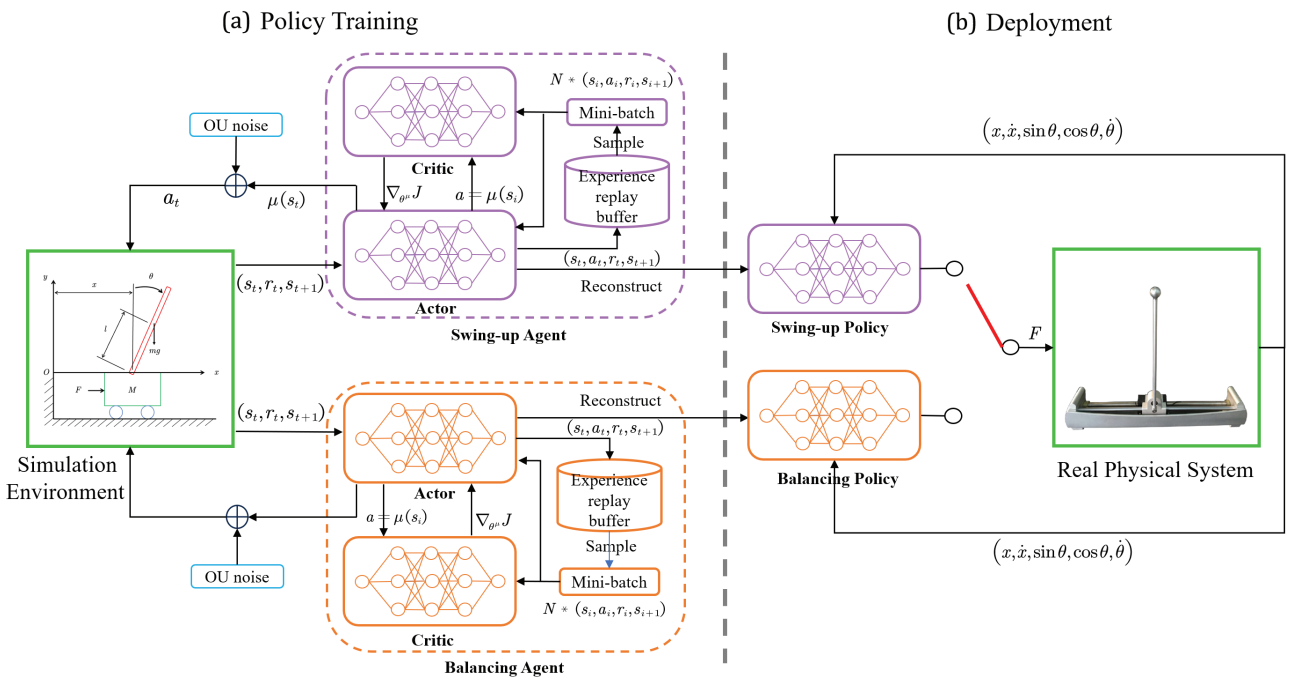


Fig. 4. Policy training and deployment process.

To improve generalization and minimize the sim-to-real gap, specific techniques are adopted, detailed in the following section.

4.2.1. *Simulation environment*

The simulation environment is constructed in MATLAB Simulink with a custom MATLAB Function block. The force is bounded within $[-6\text{N}, 6\text{N}]$. An episode ends when $|x| > L = 0.2$ m or when the episode time exceeds 10 s. The time step is $h = 0.02$ s.

(a) Reward Function Design. To enable efficient swing-up and stabilization of the cart-pole system, we design a reward function that penalizes deviations from the desired upright equilibrium. The reward function consists of four components and is formulated as follows:

$$R_t^{(1)} = -0.1(10\theta_t^2 + x_t^2 + 0.05\dot{\theta}_t^2) - 1000\eta_1, \tag{40}$$

where θ_t is the angle of the pole (in radians), x_t is the position of the cart, $\dot{\theta}_t$ is the angular velocity of the pole and η_1 is a binary flag that indicates boundary violations. Specifically, $\eta_1 = 1$ if $|x_t| > L$, and $\eta_1 = 0$ otherwise.

(b) Explanation of Reward Terms.

- **Pole angle penalty ($10\theta_t^2$):** Encourages the pole to remain upright.
- **Cart position penalty (x_t^2):** Discourages excessive displacement from the center.
- **Angular velocity penalty ($0.05\dot{\theta}_t^2$):** Suppresses abrupt swing behavior.
- **Boundary termination penalty ($-1000\eta_1$):** Enforces strict safety constraints by terminating unstable episodes.

The weights were selected empirically to balance swing-up success and final stabilization.

4.2.2. *Training setup and agent structure*

To improve learning efficiency and adapt to different control phases, the overall task is divided into two subtasks: *swing-up* and *balancing*. Each subtask is managed by a dedicated DDPG agent with customized training settings

(a) Swing-Up Phase. The swing-up agent is trained with initial pole angles near the downward equilibrium. The aim is to transition the system to the vicinity of the upright state. The reward function remains in the original form given in Eq. (40).

(b) Balancing Phase. The balancing agent is trained with initial conditions close to the upright position, i.e. $|\theta_t| < 0.17$ rad, while it is also the switching logic same as model-based control design. The aim is to balance the pole after the swing-up phase. The reward function is similar to the reward function of the swing-up phase but slightly different, which is given by

$$R_t^{(2)} = -0.1(5\theta_t^2 + x_t^2 + 0.05\dot{\theta}_t^2) - 200\eta_2. \tag{41}$$

Different from (40), the termination flag η_2 is redefined to reflect deviation from the upright range, $\eta_2 = 1$ if $|\theta_t| > 0.35$ rad, and $\eta_2 = 0$ otherwise. This incentivizes the agent to stabilize the pole in the upright position and penalizes excessive angular deviations.

4.2.3. *State representation and network architecture*

(b) Observation Vector. The observation vector includes both translational and rotational components:

$$\mathbf{o}_t = [x_t, \dot{x}_t, \sin\theta_t, \cos\theta_t, \dot{\theta}_t]^\top$$

which avoids discontinuities near $\theta_t = \pm\pi$ rad by using trigonometric representations.

(b) Neural Network Design. Both the actor and critic networks adopt two hidden layers with 128 and 200 neurons, respectively. For the critic network, the action input is concatenated in the second hidden layer. All hidden layers use ReLU activation, and actor output uses scaled `tanh` activation to enforce action limits.

4.2.4. *Training hyperparameters*

The training parameters for both the swing-up and balancing agents are summarized in Table 2.

This two-stage DDPG framework, comprising separate agents for swing-up and balancing, along with a tailored reward function and safe training environment, enables robust real-world deployment of cart-pole control under safety and generalization constraints.

4.3. *Sim-to-real transfer*

Deploying policies trained in simulation onto real hardware is challenging due to inevitable discrepancies between simulation models and real-world dynamics, known as the sim-to-real gap. To achieve robust zero-shot transfer, we adopt several techniques.

Table 2. DDPG agent training parameters.

Parameter	Value
Experience buffer length	10^6
Mini-batch size	128
Discount factor γ	0.95
Target smoothing factor τ	10^{-3}
Starting epsilon	1
Ending epsilon	0.01
Epsilon decay rate	0.005

System Identification. Physical parameters, such as cart mass, are accurately identified by experiments in the real world to calibrate the simulation model.

Domain Randomization. To enhance robustness and generalization, parameters such as cart mass M, pole length l and friction coefficient k are randomly sampled within a bounded range around their nominal values at the beginning of each episode.

$$M \in [M_0 - \delta, M_0 + \delta],$$

where M_0 is the identified value of the cart mass and δ is the amplitude of fluctuation, which exposes the agent to various dynamics of the system during training.

Sensor Noise Modeling. To simulate measurement imperfections, zero-mean Gaussian noise with variance 0.01 is added to the observed states.

Subtask Training. As described in Sec. 4.2, we handle swing-up and balancing subtasks with separate DDPG agents to enhance robustness during mode transitions.

By integrating these approaches, the trained policy successfully generalizes to the physical system without additional tuning, achieving zero-shot sim-to-real transfer.

5. Experiment

5.1. *Experimental setup*

The experimental setup used in this study is based on the Googoltech real-time linear motor-driven cart-pole system, illustrated in Fig. 5. The setup consists primarily of a host computer, a drive control box and an inverted pendulum system consisting of a linear motor, cart and pendulum. Its mechanical assembly includes the stator and moving parts of the linear motor, the cart structure, the pendulum arm, the encoders and the supporting components.

The system dynamics and control algorithms are modeled and simulated using MATLAB/Simulink. The control logic, hardware drivers and interfaces are integrated within a Simulink model, then automatically translated into optimized C code using Simulink Embedded Coder, and deployed to the Digital Signal Processor (DSP) hardware for real-time control. Bidirectional data exchange with the host computer allows for real-time monitoring, visualization, analysis and logging of essential system parameters such as cart position, pendulum angle, control actions and performance metrics.

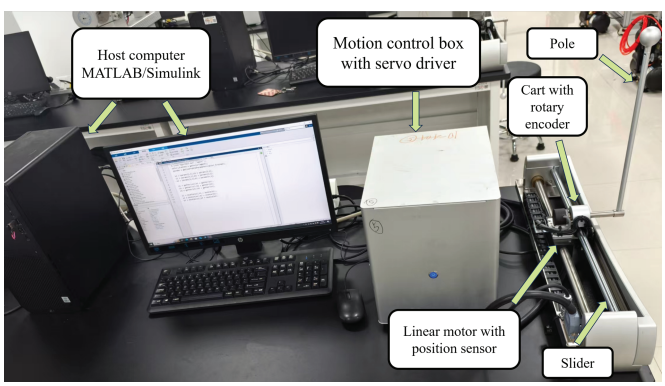


Fig. 5. Cart-Pole system experimental platform.

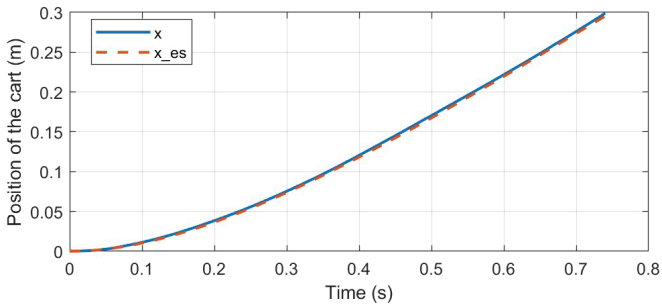
5.2. *Results of model-based control*

The model-based control strategy begins with identifying the friction model and designing a corresponding compensation mechanism. Next, the swing-up controller applies force F, collects offline data (including input forces F and output states $(x, \dot{x}, \theta, \dot{\theta})$), and uses these to identify parameters. During the swing-up phase, an energy-based controller with LBFs drives the pendulum to the upright position while ensuring safety. As the pole nears vertical equilibrium, the system transitions to the LQR for stabilization. This control framework was tested for validation using MATLAB Simulink Desktop Real-Time at Googoltech.

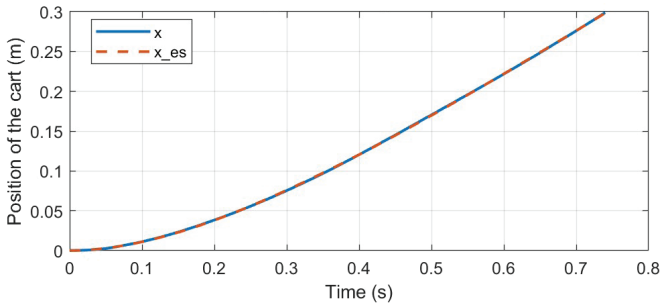
To identify the parameters of the friction model, a group of input forces $\{F_{\text{in}}\}_{i=1}^{N}$ is applied to the system, and the cart positions are collected with respect to time, respectively. The real and estimated positions of the cart by two methods introduced in Sec. 2.2 while the input force is $5.4N$, are shown in Fig. 6. The solid blue curve in Fig. 6 represents the measured position, used to estimate the parameters: p_1, p_2, p_3, p_4. The red dashed curve representing the estimated position confirms the accuracy of the estimation, which allows Eq. (12) to calculate the correct parameters of the friction model: $[M + m, F_c, k]^\top = [1.1814, 2.0417, 4.2638]^\top$, which further guarantees an effective friction compensation design.

Although the system identification process can, in principle, be carried out simultaneously with the swing-up phase in Eq. (23), they are separated in the practical implementation. This is because achieving real-time computation on the experimental hardware is challenging during the swing-up phase. Moreover, the identification procedure requires not a large amount of data, thus, it is entirely feasible to allow the pole to swing for a short duration to collect the necessary samples. Meanwhile, to prevent potential safety risks due to high cart or pole velocities, the input force can be set to a relatively small magnitude during the data collection process. Using Eq. (12), 200 samples ($h = 0.01$) were collected during the $t = [0, 2]$ s data collection process. Estimated parameters are: $\xi = [M + m, ml, J + ml^2]^\top = [1.1796, 0.0261, 0.0075]^\top$. The swing-up controller increases the pseudo-energy E_p, as shown in Fig. 7, moving the pole from $\theta = -\pi$ rad to $\theta = 0$.

As introduced in Sec. 3, the swing-up controller works firstly to drive the pole sufficiently closely to the ideal



(a)



(b)

Fig. 6. Real and estimated positions of the cart by two methods ($F_{\text{in}} = 5.4N$).

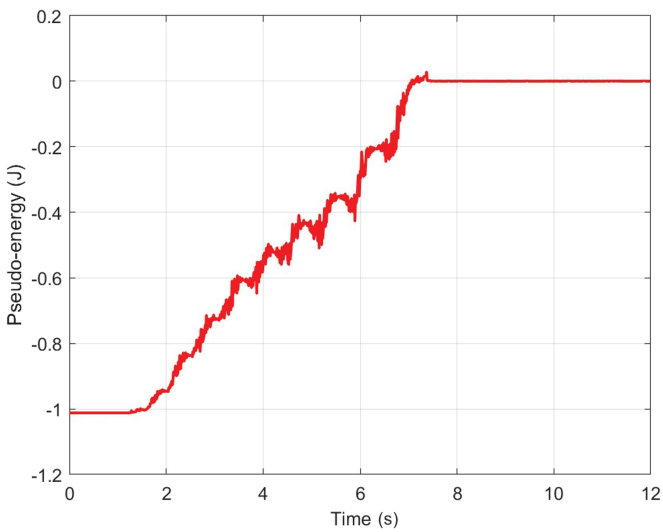


Fig. 7. Pseudo-energy of the system.

upward position, and then the balancing controller starts working (The switching logic is $\theta < 0.17$ rad). At roughly $t = 7.5$ s, the system transitions to the balancing controller as the values of θ and $\dot{\theta}$ converge towards zero, indicating the system's readiness for balance (Fig. 8). To achieve stabilization, the feedback gain of the balance controller K is determined using the LQR algorithm, based on

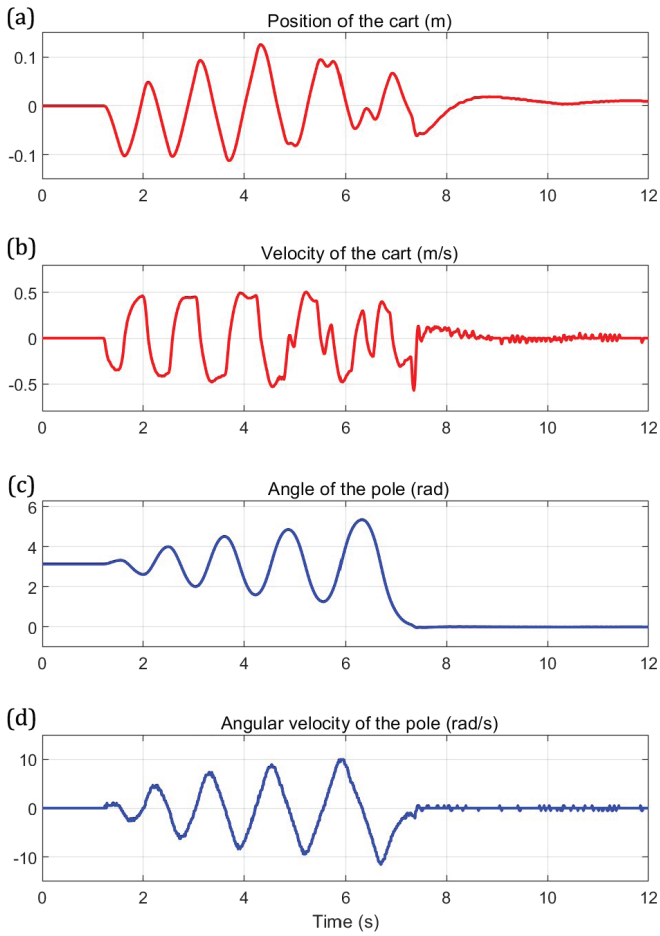


Fig. 8. Cart position and velocity; pole angle and angular velocity of the model-based control ($F_{\text{in}} = 4.2$N).

the parameters of the identified system. By selecting the weight matrices as $Q = \text{diag}(20, 5, 100, 50)$ and $R = 1$ (The pole angle θ is assigned the largest weight to ensure upright stability, followed by angular velocity $\dot{\theta}$ to avoid excessive angular motion. The cart position x is given a smaller weight to keep it at the center, while cart velocity $\dot{x}$ is penalized the least. The input consumption is not the primary factor we consider.), the calculated feedback gain is $K = [-4.4721, -6.3036, -49.6229, -10.8670]$. This gain allows the controller to effectively stabilize the system, as illustrated in Fig. 8.

Remark 3. The value of $x_{\max}$ of the swing-up controller in Eq. (33) can be set as the half length of the slider, guaranteeing that the cart position x never exceeds $x_{\max}$. For the velocity constraint, F_ν serves as a soft penalty so that $\nu_{\max}$ can be chosen sufficiently large guaranteeing that ν does not exceed $\nu_{\max}$. In practice, the velocity barrier serves only as a mild penalty, while the position barrier provides the primary safety guarantee.

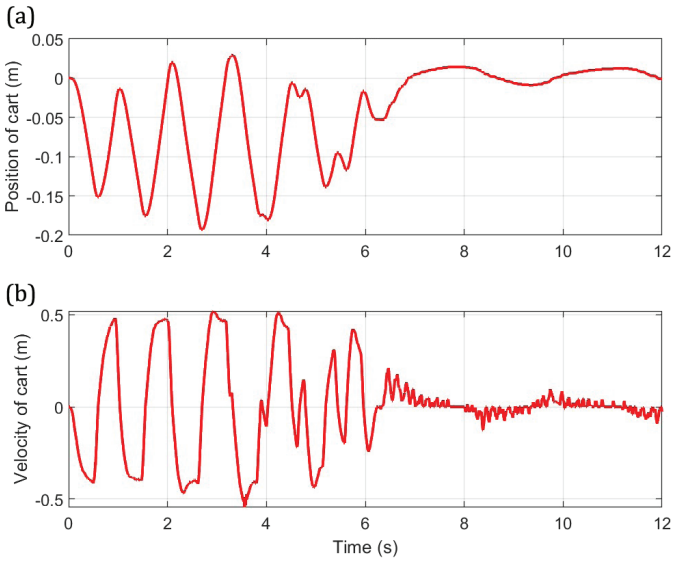


Fig. 9. Cart position and velocity under control without LBFs.

Figure 9 illustrates the performance of the energy control algorithm without LBFs. When comparing Fig. 8(a) with Fig. 9(a), it is evident that the improvement algorithm causes the system to deviate significantly from the center during the swing phase. In contrast, the improved algorithm effectively keeps the system near the center of the orbit, minimizing the risk of collisions and experimental failures. This demonstrates the effectiveness of the improved LBFs-based energy control algorithm.

5.3. *Results of RL-based control*

5.3.1. *Training in simulation*

We train swing-up and balancing DDPG agents for 1000 episodes, 500 episodes each and directly deploy the learned policies on the physical system without fine-tuning. The swing-up policy training curve is shown in Fig. 10(a), while the iteration in which the balancing agent achieved the maximum reward is shown in Fig. 10(b). In the simulation, for the swing-up agent, the parameters of the virtual environment of RL are randomized before each episode. The values of the parameters are selected as shown in Table 3.

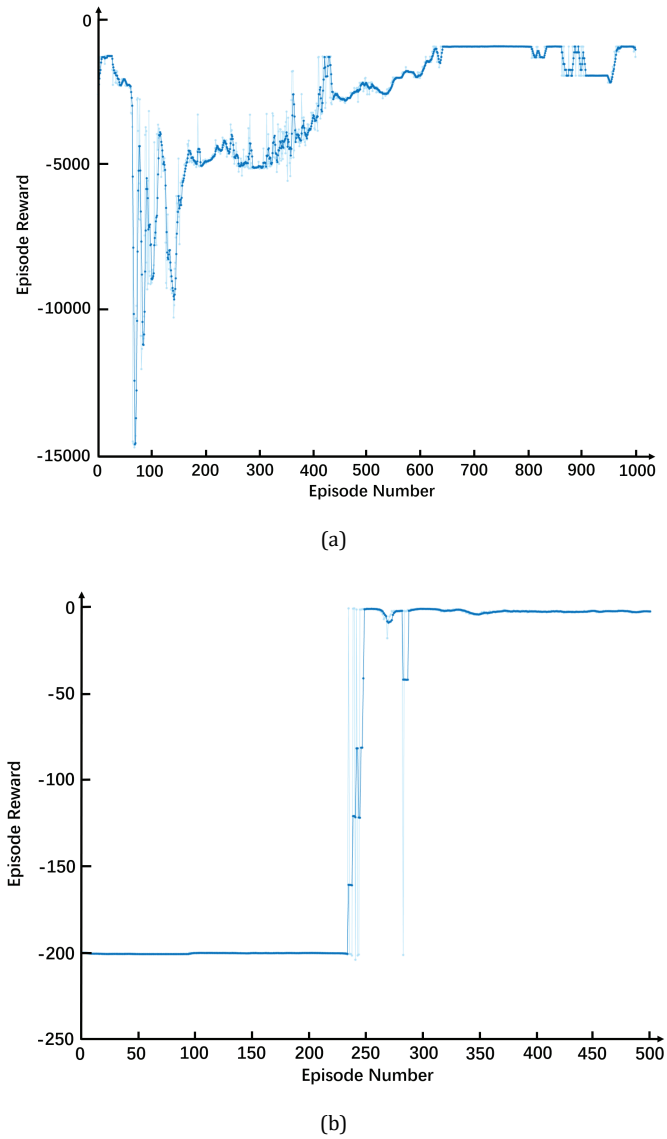


Fig. 10. Training curves in simulation. (a) Total reward per episode of swing-up agent. (b) Total reward per episode of balancing agent.

For the balancing agent, before each episode, in addition to randomizing the system parameters (as is done for the swing-up agent), we also randomize the initial states. Specifically, the initial pole angle is sampled from $(-0.17, 0.17)$ rad, and the angular velocity is sampled from $(-4, 4)$ rad/s. The

Table 3. Physical parameters and their fluctuation ranges.

Symbol	Physical interpretation	Value range
M	Mass of the cart	$[1.06-0.2, 1.06+0.2]$
m	Mass of the pole	$[0.12-0.03, 0.12+0.03]$
l	Distance from the pivot to the pole's center of mass	$[0.22-0.04, 0.22+0.04]$
F_c	Coulomb friction between the cart and slider	$[2.04-0.6, 2.04+0.6]$
k	Viscous friction coefficient between the cart and slider	$[4.26-1, 4.26+1]$

initial position and velocity of the cart are also randomized, with the position sampled from $(-0.03, 0.03)$ m and the velocity from $(-0.2, 0.2)$ m/s. This approach enables the policy to switch flexibly.

5.3.2. *Real-world experiments*

After training in the simulation for several hours, the average total rewards per episode for the swing-up and balancing agents reach and maintain the maximums of -958 and -2.71, respectively, while they can achieve swing-up and balancing control. So, the trained policy or controller is deployed in the real physical system. All the parameters of each layer of the DDPG actor network are extracted, used to reconstruct the controller in the MATLAB Simulink Desktop Real-Time of GoogolTech.

And Fig. 11 illustrates the performance of the DDPG algorithm. At t = 3.5 s, the swing-up policy allows the pole to reach a close proximity to the upward position with the angular velocity almost equal to zero. Then the controller switches to the balancing policy, and it successfully balances the pole.

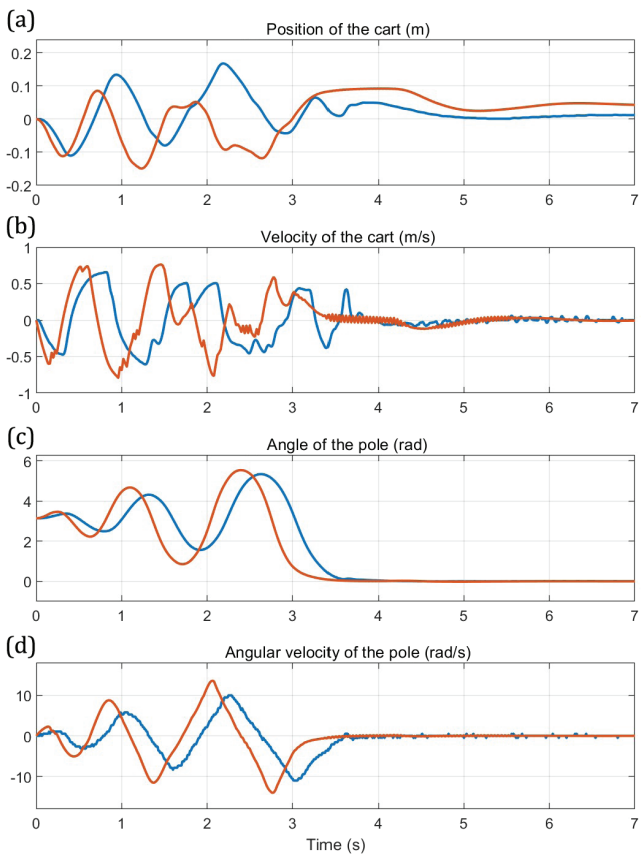


Fig. 11. Cart position and velocity, pole angle and angular velocity of the RL-based approach. The red curve denotes the simulation results, while the blue curve represents the real-world experimental results.

The real system trajectory exhibits a delay relative to the simulation, as illustrated in Fig. 11. This phenomenon may be attributed to factors such as actuator inertia and execution latency in real hardware, coupled with delays arising from sensor sampling and filtering. Furthermore, the simulation environment often fails to fully replicate the uncertainties and nonidealities inherent in the physical system, leading to a mismatch between the control rhythm learned during training and the actual dynamics. Furthermore, since RL prioritizes the optimization of cumulative long-term rewards over immediate responsiveness, it can exacerbate the temporal offset between policy output and system-observed behavior.

Although a single policy trained with the unified reward (40) can accomplish both swing-up and balancing in simulation, we find that this approach does not reliably transfer to the physical system. The underlying reason is that the two subtasks exhibit different control characteristics: The swing-up phase represents a large-range nonlinear motion far away from the equilibrium, where the controller must inject energy to drive the pole toward the upright position. In contrast, the balancing phase operates within a small neighborhood of the equilibrium, where the dynamics are approximately linear and stabilization is achieved through small inputs. In the simulation, a single policy can find an ideal trajectory such that angle θ and angular velocity $\dot{\theta}$ are sufficiently close to zero when the pole reaches the upright position. However, in real hardware, friction variations, sensory noise and actuator delays can cause the single policy to fail when approaching to the upright position. Inspired by model-based control, we decouple the task into swing–up and balancing policies, each agent focuses on a much narrower dynamic region with different reward functions. And the balancing policy was trained with random initial states, especially the angle and angular velocity were not close to zero. After training, the learned policy can stabilize the pole even when the angular velocity does not approach zero when switching. This significantly improves sim-to-real transfer performance.

5.4. *Comparative analysis of engineering performance*

To evaluate the engineering performance of the model-based and RL-based controllers, we conducted comparative experiments under varying physical conditions. In practice, we observed that the friction coefficients k and F_c of the slider change over time due to environmental factors such as temperature and wear. To assess sensitivity to these variations, we performed two sets of experiments conducted one week apart.

For the model-based controller, system parameters $M+m, ml, J + ml^2, k, F_c$ were first identified and used to

perform ten swing-up and balancing tests. After one week, two types of evaluations were performed under different physical conditions: firstly, ten trials were conducted using the original parameters to test sensitivity to parameter mismatch, including experiments with different values of $F_{\max}$; Second, 10 similar trials were performed but using newly re-identified parameters. For the RL-based controller, the trained policy was deployed directly without any on-site fine-tuning. Ten trials were conducted in the first session and another 10 were performed one week later without retraining under changed friction conditions.

The experimental results are summarized in Tables 4 and 5. Table 4 shows that the model-based controller exhibits a strong dependence on accurate parameters. When the parameters of first week were reused in the second week, the success rate dropped to zero, indicating that friction variations significantly affected the dynamics of the system. When the controller reused previously identified parameters, even slight friction mismatch resulted in inaccurate control input, causing the success rate to drop to zero. In contrast, the RL-based controller achieved a consistent 100% success rate across both weeks, despite environmental changes and did so without any retraining.

Furthermore, the effectiveness of the designed control law depended on the selection of $F_{\max}$ where it was not always successful under different values of $F_{\max}$. In theory, when the pseudo-energy E_p reaches zero, the control input becomes zero so that no extra energy is introduced into the system. Then, when the pole reaches very close to the upright position, both the angle and angular velocity should approach zero. However, this ideal behavior does not hold in the physical system. Due to hardware-related factors such as actuator delays and sensing latency, certain values of $F_{\max}$ cause the pole arriving at the switching region with a nonnegligible residual angular velocity. As a result, when the controller switches to the LQR controller, the stabilizing control effort required to counteract this residual velocity exceeds the admissible input range. The LQR input saturates, preventing the controller from generating the corrective force needed to stabilize the pole, ultimately leading to failure.

Table 4. Control success rate of model-based approach.

Model-based	Success rate	Success rate (reidentified parameters)
Exp1	50%	
Exp2	0%	60%

Table 5. Control success rate of RL-based approach.

RL-based	Success rate
Exp1	100%
Exp2	100%

It can be seen that both approaches ensure safety: the model-based approach uses LBFs to avoid limits and high speeds, while RL learns safety through rewards and penalties. However, RL requires a lot of upfront computation for training, which lasts for hours. Model-based control requires accurate offline system identification and friction compensation, but an accurate initial setup. RL adapts well to a variety of conditions and uncertainties. At the same time, the model-based approach is less flexible and relies on adjusting the control law (the term $F_{\max}$) of the phase transition.

Remark 4. Our switching logic is based on a hard threshold of $|\theta| < 0.17$ rad. Theoretically, such a hard switching logic can induce oscillations during the transition between controllers. In practice, the observed oscillations mainly manifest in the cart motion, specifically, small overshoots in cart position and velocity after switching as shown in Fig. 8. However, no noticeable oscillation occurs in the pole angle or angular velocity, which are the primary states that determine whether the pole can be stabilized upright. Since the weight matrix Q of LQR controller places greater emphasis on the pole angle and angular velocity, they converge smoothly toward zero, even though the cart exhibits minor transient adjustments.

6. Conclusion

This work experimentally compared model-based and RL-based controllers for cart-pole swing-up and stabilization. The model-based approach, combining friction compensation, an improved energy-based swing-up law and LQR balancing, performs efficiently when system parameters are accurate but is sensitive to friction variation and model mismatch. In contrast, the task-decoupled DDPG controller trained with domain randomization and noise modeling demonstrated stronger robustness without retraining across changing conditions. The model-based method is simple and efficient when the system is well understood but depends on prior modeling expertise. In contrast, the RL method is more flexible for complex or uncertain dynamics, although it requires longer training and a careful reward design. Importantly, the RL design was directly inspired by the model-based phase decomposition, showing that principled control structures can significantly improve sim-to-real reliability while it is a central issue when applying RL to real-world control tasks. In this case, a promising future direction is to integrate model-based structure into learning frameworks to improve the reliability of sim-to-real transfer. In future work, we aim to study how to combine the advantages of both approaches when the system model

structure is not precisely known, or there are unmodeled dynamics.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 62350710214, 62121004, 62550003 and U23A20325, and in part by the Shenzhen Key Laboratory of Control Theory and Intelligent Systems under Grant No. ZDSYS20220330161800001.

ORCID

Qihua Chen
https://orcid.org/0009-0007-0141-6656
Mingxiang Liu
https://orcid.org/0000-0003-3655-418X
Minyue Fu
https://orcid.org/0000-0002-4659-601X

References

[1] J. Brownlee, The pole balancing problem: A benchmark control theory problem, Technical Report No. 7-01, Swinburne University of Technology, Australia (2005).
[2] H. Chen, B. Wang, Z. Hong, C. Shen, P. M. Wensing and W. Zhang, Underactuated motion planning and control for jumping with wheeled-bipedal robots, *IEEE Robot. Autom. Lett.* **6**(2) (2020) 747–754.
[3] J. Pei and P. Rothhaar, Demonstration of the space launch system augmenting adaptive control algorithm on pole-cart platform, in *2018 AIAA Guidance, Navigation, and Control Conf.* (American Institute of Aeronautics and Astronautics, 2018), p. 0608.
[4] H. Fukushima, M. Kakue, K. Kon and F. Matsuno, Transformation control to an inverted pendulum for a mobile robot with wheel-arms using partial linearization and polytopic model set, *IEEE Trans. Robot.* **29**(3) (2013) 774–783.
[5] T. Takei, R. Imamura and S. Yuta, Baggage transportation and navigation by a wheeled inverted pendulum mobile robot, *IEEE Trans. Ind. Electron.* **56**(10) (2009) 3985–3994.
[6] K. J. Åström and K. Furuta, Swinging up a pendulum by energy control, *Automatica* **36**(2) (2000) 287–295.
[7] M. Fu, Control of the cart-pole system: Model-based vs. model-free learning, *IFAC-PapersOnLine* **56**(2) (2023) 11847–11852.
[8] T.-B. Dang *et al.*, PID control for cart and pole system: Simulation and experiment, *J. Fuzzy Syst. Control* **2**(1) (2024) 29–35.
[9] R. Banerjee and A. Pal, Stabilization of inverted pendulum on cart based on pole placement and LQR, in *2018 Int. Conf. Circuits and Systems in Digital Enterprise Technology* (*ICCSDET*) (IEEE, 2018), pp. 1–5.
[10] R.-J. Wai and L.-J. Chang, Adaptive stabilizing and tracking control for a nonlinear inverted-pendulum system via sliding-mode technique, *IEEE Trans. Ind. Electron.* **53**(2) (2006) 674–692.
[11] M.-S. Park and D. Chwa, Swing-up and stabilization control of inverted-pendulum systems via coupled sliding-mode control method, *IEEE Trans. Ind. Electron.* **56**(9) (2009) 3541–3555.
[12] M. Mahmoud, R. Saleh and A. Ma'arif, Stabilizing of inverted pendulum system using robust sliding mode control, *Int. J. Robot. Control Syst.* **2**(2) (2022) 230–239.
[13] S. Yurkovich and M. Widjaja, Fuzzy controller synthesis for an inverted pendulum system, *Control Eng. Pract.* **4**(4) (1996) 455–469.
[14] C.-S. Chen and W.-L. Chen, Robust adaptive sliding-mode control using fuzzy modeling for an inverted-pendulum system, *IEEE Trans. Ind. Electron.* **45**(2) (1998) 297–306.
[15] M. I. El-Hawwary, A.-L. Elshafei, H. M. Emara and H. A. Abdel Fattah, Adaptive fuzzy control of the inverted pendulum problem, *IEEE Trans. Control Syst. Technol.* **14**(6) (2006) 1135–1144.
[16] H. Olsson, K. J. Åström, C. Canudas de Wit, M. Gäfvert and P. Lischinsky, Friction models and friction compensation, *Eur. J. Control* **4**(3) (1998) 176–195.
[17] Q. Chen, M. Liu and M. Fu, System identification based control design for the cart-pole system, in *2024 Int. Conf. Intelligent Robotics and Automatic Control* (*IRAC*) (IEEE, 2024), pp. 603–606.
[18] F. L. Lewis and D. Vrabie, Reinforcement learning and adaptive dynamic programming for feedback control, *IEEE Circuits Syst. Mag.* **9**(3) (2009) 32–50.
[19] B. Singh, R. Kumar and V. P. Singh, Reinforcement learning in robotic applications: A comprehensive survey, *Artif. Intell. Rev.* **55**(2) (2022) 945–990.
[20] Y. Song, A. Romero, M. Müller, V. Koltun and D. Scaramuzza, Reaching the limit in autonomous racing: Optimal control versus reinforcement learning, *Sci. Robot.* **8**(82) (2023) eadg1462.
[21] G. Zhu, R. Zhou, W. Ji and S. Zhao, LAMARL: LLM-aided multi-agent reinforcement learning for cooperative policy generation, *IEEE Robot. Autom. Lett.* **10** (2025) 7476–7483.
[22] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra, Continuous control with deep reinforcement learning (2015), arXiv:1509.02971.
[23] S. Zhao, *Mathematical Foundations of Reinforcement Learning* (Springer-Verlag, Singapore, 2025).
[24] S. Kumar, Balancing a Cart-Pole system with reinforcement learning — A tutorial (2020), arXiv:2006.04938.
[25] G. Dulac-Arnold, N. Levine, D. J. Mankowitz, J. Li, C. Paduraru, S. Gowal and T. Hester, Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis, *Mach. Learn.* **110**(9) (2021) 2419–2468.
[26] W. Zhao, J. Peña Queralta and T. Westerlund, Sim-to-real transfer in deep reinforcement learning for robotics: A survey, in *2020 IEEE Symp. Series on Computational Intelligence* (*SSCI*) (IEEE, 2020), pp. 737–744.
[27] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova and D. Batra, Sim2real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robot. Autom. Lett.* **5**(4) (2020) 6670–6677.
[28] Z. Wang and S. Zhao, Sim-to-real transfer in reinforcement learning for maneuver control of a variable-pitch MAV, *IEEE Trans. Ind. Electron.* **72** (2025) 10445–10454.
[29] B. D. O. Anderson and J. B. Moore, *Optimal Control*: *Linear Quadratic Methods* (Courier Corporation, 2007).

Qihua Chen received his B.S. degree in Automation from the Shandong University, Jinan, China, in 2023. He is currently pursuing the M.S. degree at the School of Automation and Intelligent Manufacturing, the Southern University of Science and Technology, Shenzhen, China. His research interests include optimal control, reinforcement learning and their applications.

Mingxiang Liu received his Ph.D. degree in control science and engineering from the Guangdong University of Technology, Guangzhou, China, in 2023.

Currently, he is Postdoctoral Researcher with the School of Automation and Intelligent Manufacturing, Southern University of Science and Technology, Shenzhen, China. His research interests include optimal control, adaptive control, dynamic games, reinforcement learning and their applications.

Minyue Fu (Life Fellow, IEEE) received his Bachelor degree in Electrical Engineering from the University of Science and Technology of China, in 1982, and master and Ph.D. degrees in Electrical Engineering from the University of Wisconsin-Madison, USA in 1983 and 1987, respectively. From 1987 to 1989, he served as Assistant Professor in the Department of Electrical and Computer Engineering, Wayne State University, USA. He joined the University of Newcastle, Australia, in 1989 and became Chair Professor in electrical engineering in 2003. His current research interests include networked control systems, distributed estimation and control, high-precision control and reinforcement learning.

Prof. Fu has been Associate Editor for the *IEEE Transactions on Automatic Control*, *IEEE Transactions on Signal Processing*, *Automatica* and *Journal of Optimization and Engineering*. He is a Fellow of IEEE, Fellow of IFAC, Fellow of Engineers Australia and Fellow of Chinese Association of Automation.