machines

MDPI

Article

Transition Control of a Double-Inverted Pendulum System Using Sim2Real Reinforcement Learning

Taegun Lee, Doyoon Ju and Young Sam Lee *

Department of Electrical and Computer Engineering, Inha University, Incheon 22212, Republic of Korea; dlxorjs815@inha.edu (T.L.); seiko_kr@inha.edu (D.J.)
* Correspondence: lys@inha.ac.kr

Abstract: This study presents a sim2real reinforcement learning-based controller for transition control in a double-inverted pendulum system, addressing the limitations of traditional control methods that rely on precomputed trajectories and lack adaptability to strong external disturbances. By introducing the novel concept of 'transition control', this research expands the scope of inverted pendulum studies to tackle the challenging task of navigating between multiple equilibrium points. To overcome the reality gap—a persistent challenge in sim2real transfer—a hardware-centered approach was employed, aligning the physical system's mechanical design with high-fidelity dynamic equations derived from the Euler–Lagrange equation. This design eliminates the need for software-based corrections, ensuring consistent and robust system performance across simulated and real-world environments. Experimental validation demonstrates the controller's ability to reliably execute all 12 transition scenarios within the double-inverted pendulum system. Additionally, it exhibits recovery characteristics, enabling the system to stabilize and return to equilibrium point even under severe disturbances.

Keywords: reinforcement learning; double-inverted pendulum; sim2real transfer; transition control

Academic Editor: Carmine Maria Pappalardo

Received: 15 December 2024
Revised: 10 February 2025
Accepted: 24 February 2025
Published: 26 February 2025

Citation: Lee, T.; Ju, D.; Lee, Y.S. Transition Control of a Double-Inverted Pendulum System Using Sim2Real Reinforcement Learning. *Machines* **2025**, *13*, 186. https://doi.org/10.3390/machines13030186

1. Introduction

From a control engineering perspective, the inverted pendulum system presents significant challenges due to its instability, nonlinear dynamic equations, and non-minimum phase characteristics. Additionally, it is an underactuated system, where the number of control variables exceeds the number of actuators. These complexities have made the inverted pendulum system a widely used testbed for evaluating various control strategies over an extended period [1,2]. The primary research areas involving this system include 'swing-up control', which aims to transition the pendulum from a downward-hanging position to an upright one, and 'balance control', which focuses on maintaining the pendulum in its inverted state. Among these, swing-up control is generally regarded as the more challenging topic, as it necessitates addressing the system's nonlinearity, instability, and input–output constraints. The difficulty escalates further in the case of a double-inverted pendulum, which involves multiple pendulums. It was not until 2007 that Graichen introduced an effective two-degree-of-freedom swing-up control technique specifically designed for the double-inverted pendulum system [3].

Moreover, the rapid advancements in artificial intelligence (AI) have spurred active research into the application of reinforcement learning, particularly utilizing deep neural networks, in the field of control engineering [4,5]. Traditional control engineering approaches often require highly accurate mathematical models and precise system parameters, which

can be cumbersome and restrictive. In contrast, reinforcement learning-based controller design offers a new paradigm that addresses these limitations by enabling controller development even without prior knowledge of the target system [6]. Reinforcement learning utilizes data obtained through direct system interaction to iteratively refine the control policy, ultimately designing an optimal controller [7]. Within this interdisciplinary domain, inverted pendulum systems including multi-stage variations remain critical testbeds for validating AI-based control methodologies. They are extensively employed in applications such as replacing traditional control methods to solve the swing-up problem [8,9] and evaluating the performance of novel AI-driven techniques [10,11].

However, when reinforcement learning agents directly interact with physical systems during the training phase, several challenges emerge. These include the substantial time required for interaction, increased costs associated with data acquisition, physical system constraints, and potential risks of damage during experimentation [12]. To circumvent these issues, most research on reinforcement learning-based controllers relies on simulation environments. Such environments replicate the dynamic characteristics of physical systems, allowing reinforcement learning algorithms to be applied within these controlled settings. This method, where training is conducted in a simulation environment and the resulting model is transferred to a real-world system, is commonly referred to as 'sim2real transfer' [13]. By leveraging the sim2real approach, researchers can explore a broader range of data and scenarios in the simulation environment, which is unencumbered by the physical constraints of real-world systems.

Nonetheless, the sim2real learning technique faces a significant challenge: the existence of a reality gap, which refers to discrepancies in environmental conditions and system responses between simulation and the actual physical system. The magnitude of this gap can critically affect the performance of the model, causing it to function improperly or experience performance degradation when applied to the real system [14]. To address this issue, extensive research efforts have been made with the most commonly adopted techniques being 'domain randomization' [15] and 'domain adaptation' [16]. Domain randomization introduces uncertainty into the simulation environment, enhancing the model's generalization ability and improving its transferability to real-world systems. However, this approach does not fundamentally address the physical and geometric discrepancies between simulation and reality, often resulting in unnecessary distribution shifts and potential performance degradation. Conversely, domain adaptation aims to mitigate distribution shifts between the simulation and real environments by refining the model's feature extraction process or identifying a shared feature space. While this can enhance model performance, it has limitations when model fidelity is low, the distribution gap is wide, or data are insufficient, leading to overfitting risks.

Even though these software-based approaches provide valuable strategies to reduce the reality gap, they inherently struggle to resolve the core physical and mechanical mismatches between simulation and real environments. To overcome these limitations, this study employs a hardware-based approach aimed at directly addressing the root causes of the reality gap. The proposed method involves deriving mathematical model equations of the actual system using the Euler–Lagrange equation and designing the mechanical structure of the physical system to align as closely as possible with the assumptions underlying the model. This alignment enhances model fidelity, thereby reducing the dynamic response differences between the simulation and the actual system. Unlike the indirect improvements offered by software-based solutions, the hardware-based approach directly mitigates discrepancies by refining the physical characteristics and mechanical precision of the system. By achieving high model fidelity, the proposed system enables the acquisition and learning of extensive data in a simulation environment free from physical constraints

while avoiding the performance degradation typically caused by the reality gap. This approach opens new possibilities for reinforcement learning-based controllers to address challenges previously unattainable using traditional control techniques.

To validate this approach, the paper demonstrates the effectiveness of a reinforcement learning-based controller capable of executing control actions that are unachievable with conventional two-degree-of-freedom control techniques for the swing-up control of a double-inverted pendulum. Section 2 details the limitations of traditional control techniques and describes how these can be addressed through the sim2real learning methodology. Additionally, the study expands on the swing-up problem by leveraging the mechanical characteristics of multi-stage inverted pendulums to define a new 'transition control' problem. Section 3 derives the mathematical model equations for a high-fidelity double-inverted pendulum system and outlines its mechanical design. Section 4 presents the design and experimental application of the reinforcement learning-based controller in both simulation and real-world environments.

Finally, the findings are summarized in Section 5, where the practical contributions of the proposed approach are discussed in detail. These core contributions are outlined below:

- Introduction of Transition Control for Multi-Stage Inverted Pendulums: This study introduces the concept of transition control, which extends beyond traditional swing-up and stabilization control by enabling controlled movement between multiple equilibrium points based on pendulum angles. To the best of our knowledge, this is the first application of reinforcement learning (RL) to transition control in a multi-stage inverted pendulum system.
- Hardware-Centered Approach to Reality Gap Reduction: Unlike conventional sim2real methods that rely solely on software-based domain adaptation, this study employs a hardware-driven approach to directly align the physical system with its mathematical model. By ensuring structural consistency with the Euler–Lagrange-based model, the proposed method minimizes discrepancies between simulation and real-world dynamics, effectively reducing the reality gap.
- Sim2Real Learning with High Model Fidelity: The proposed framework leverages high-fidelity modeling and mechanical precision to enable efficient sim2real learning without requiring extensive real-world retraining. This approach ensures effective policy transfer, allowing RL-based controllers to perform reliably and robustly on physical hardware.

2. Problem Statement

2.1. Recovery Characteristics

Graichen's control technique for the double-inverted pendulum, as described in [3], involves precomputing the swing-up trajectory through offline optimization and applying it to the system via feedforward control. Subsequently, it uses feedback control to correct deviations between the actual trajectory and the precomputed trajectory. This two-degree-of-freedom (2-DOF) control strategy successfully addressed the longstanding challenge of performing swing-up control under rail length constraints for the double-inverted pendulum. In 2013, the effectiveness of this method was further validated when another researcher applied it to a structurally more complex triple-inverted pendulum, achieving successful swing-up control in this system as well [17].

However, this control approach has a critical limitation: it becomes ineffective when strong external disturbances are applied. While the feedback mechanism can handle minor disturbances, a disturbance beyond a certain threshold causes the system's response to deviate significantly from the precomputed feedforward trajectory. When this happens, feedforward control becomes ineffective, and feedback control can no longer correct the

error, leaving the system in an uncontrollable state. Because the feedforward trajectory is precomputed offline, it cannot be adjusted during operation, making it impossible for the system to reinitiate swing-up control once it reaches this state.

To address these limitations, this study leverages the reinforcement learning technique. A reinforcement learning agent interacts with the environment, iteratively refining its action policy based on the state information observed and the corresponding rewards. Through repeated interactions, the agent accumulates extensive state information and continues learning until it can identify the optimal action for any given state. Once trained, the controller is capable of executing the desired control regardless of the system's initial or current state. This process can be likened to solving a maze with a fixed destination. If the starting point is randomly varied during the learning phase, the agent learns to navigate the maze effectively from any position. Upon completing training, the destination can be reached immediately irrespective of the starting location [18]. Similarly, in the case of the inverted pendulum, the reinforcement learning-based controller can guide the system to the upright state, corresponding to the swing-up control goal, regardless of the initial or disturbed state. Even when subjected to strong disturbances, the agent treats these disturbances as changes in the state information and calculates the appropriate control inputs to restore the desired state effectively.

The sim2real learning technique is critical for enabling the learning process described above. As illustrated in the earlier analogy, effective learning requires the reinforcement learning agent to experience a diverse range of states. However, in a real-world physical system, such exploration is constrained by physical limitations. For instance, due to gravity, researchers cannot arbitrarily initialize the angle and angular velocity of each pendulum except for the stable equilibrium state, where all pendulums are pointing downward. Consequently, the range of states accessible to the reinforcement learning agent is limited, preventing it from learning optimal actions for unobserved states. This limitation ultimately hampers the agent's ability to handle situations where disturbances occur effectively.

In contrast, a simulated environment eliminates such physical constraints. Within this environment, researchers can freely initialize all state variables, including the angles and angular velocities of the pendulums, as well as the position and acceleration of the cart. Leveraging these capabilities, the reinforcement learning agent can explore scenarios that may be unattainable in the real system, allowing it to accumulate a richer dataset and learn more robust control strategies. As a result, when confronted with strong disturbances, the agent is likely to recognize the perturbed state as one it has encountered during simulation training, enabling effective control without the system becoming uncontrollable.

In this paper, we define a reinforcement learning-based controller with these capabilities as exhibiting 'recovery characteristics'. Unlike traditional controllers, which may fail to maintain stability when exposed to strong disturbances, the proposed controller leverages the sim2real learning technique to restore stability even after the system enters an unstable state. This ability to adapt dynamically and recover from disturbances highlights a fundamental advantage of the sim2real approach, demonstrating its potential to address challenges that conventional control methods cannot overcome.

2.2. Transition Control

The previously mentioned references [3,17] focus solely on the swing-up control of multi-stage inverted pendulums. In the case of a single inverted pendulum, as shown in Figure 1a, there exists only one type of swing-up problem: transitioning from a stable equilibrium point, where the pendulum hangs downward, to an unstable equilibrium point in the upright position. For a double-inverted pendulum, which introduces an additional pendulum compared to the single inverted pendulum, there is one stable equilibrium

point with both pendulums hanging downward and three unstable equilibrium points, as shown in Figure 1b. This structural characteristic allows the swing-up control problem to be extended into a newly defined 'transition control' problem, which involves transitioning between the various equilibrium points. In the case of a double-inverted pendulum, there are 12 possible types of transitions, including the traditional swing-up. Research into controlling these transitions is as challenging as research on swing-up control itself.

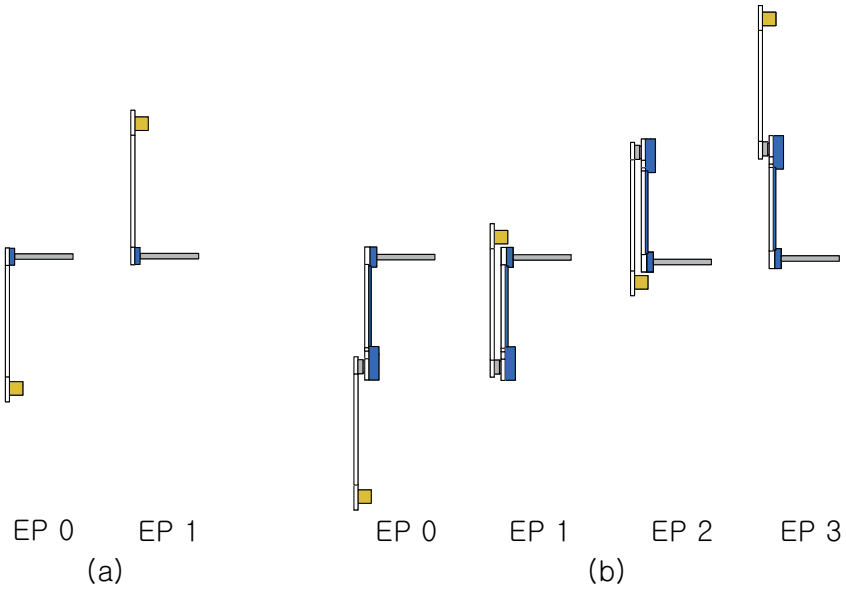

Figure 1. (**a**) Two equilibrium points of a single inverted pendulum. (**b**) Four equilibrium points of a double-inverted pendulum.

The four equilibrium points of the double-inverted pendulum—Down–Down, Down–Up, Up–Down, and Up–Up—are illustrated in Figure 1b. This paper adopts a binary representation to label these points, assigning 0 to the Down state and 1 to the Up state. For example, the Down–Up state, where the first pendulum hangs downward and the second pendulum is upright, is represented as the binary number 01 and referred to as EP1. Similarly, the Up–Up state corresponds to the binary number 11 and is referred to as EP3. This naming convention simplifies the assignment of equilibrium points by using binary rules, which can be easily extended as the number of pendulums increases. It also provides a clear and intuitive way to understand the equilibrium state of each pendulum in the system. Accordingly, the equilibrium points of the double-inverted pendulum are denoted as EP0 (Down–Down), EP1 (Down–Up), EP2 (Up–Down), and EP3 (Up–Up) throughout this paper.

The transition control problem fundamentally resembles the swing-up control problem but represents a new challenge where the success of control depends on the method applied, taking into account the rail length and actuator performance of the inverted pendulum system. Despite its importance, research on multi-stage inverted pendulums has primarily focused on the swing-up problem. To date, only one academic paper has addressed the transition control problem between equilibrium points, which was published by the research lab affiliated with the author of this paper [19]. In [19], the two-degree-of-freedom control technique was extended to solve the transition control problem. However, as with the limitations discussed earlier, when a disturbance exceeding a certain threshold is applied, the system becomes uncontrollable and is unable to transition to a specific equilibrium point from that state.

In this paper, the aforementioned problem is addressed using the sim2real-based recovery characteristic. By designing the reward function to assign the highest value when a specific equilibrium point is reached, the inverted pendulum is guided to achieve the

target equilibrium point regardless of its state. With this approach, learning for just the four equilibrium points is sufficient to enable transition control across all possible scenarios. In [19], the trajectories for all 12 possible transitions between the four equilibrium points were precomputed with each trajectory applied specifically to its corresponding transition. However, as highlighted earlier, a learning-based control strategy only requires training for the four equilibrium points themselves. This advantage becomes even more apparent when extending to a triple-inverted pendulum system, where learning needs to address only eight equilibrium points, compared to the 56 trajectories that must be precomputed in the classical control approach.

3. Double-Inverted Pendulum System

3.1. Mathematical Model of the Double-Inverted Pendulum

To fully leverage the unique advantages of the sim2real learning technique highlighted in Section 2, it is crucial that the actual system demonstrates high model fidelity to minimize the reality gap discussed in the introduction. This requires that the mathematical model equations used to construct the simulation environment (Sim) closely align with the dynamic characteristics of the actual pendulum system (Real). If the fidelity between the two is inadequate, successful learning in the simulation may fail to yield satisfactory performance when applied to the real system. To address this, we first outline the mechanical conceptual diagram and mathematical model equations of the double-inverted pendulum.

Figure 2 illustrates the mechanical conceptual diagram of the double-inverted pendulum used in the experiment. The variables in the diagram are defined as follows: M represents the mass of the cart. The masses of the first and second pendulums are denoted by m_1 and m_2, respectively. The distances from the rotational axes of the pendulums to their centers of mass are l_1 and l_2, while L_1 represents the distance from the rotational axis of the first pendulum to that of the second pendulum. The variable u indicates the cart's acceleration, and y denotes its displacement from the initial position. The rotational displacement of the first pendulum, θ_1, is defined as the angle it forms with the ground's vertical axis. The relative rotational displacement between the second pendulum and the first is represented by θ_2. Additionally, c_1 and c_2 are the friction coefficients at the rotational axes of the first and second pendulums, respectively. Lastly, i, j, and k indicate the coordinate axes of a rectangular coordinate system centered at the rail's origin.

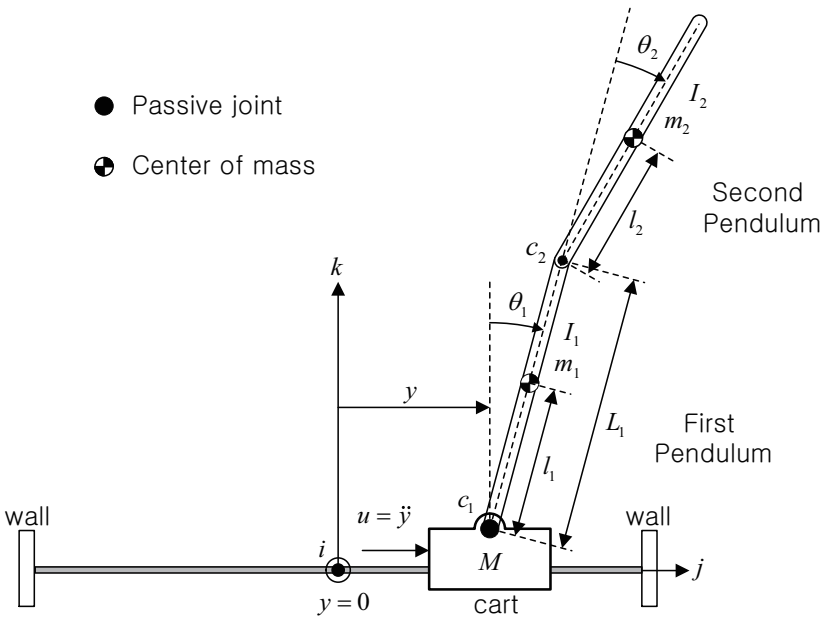

Figure 2. Mechanical conceptual diagram of a double-inverted pendulum.

This paper assumes the use of the SI unit system, and the mathematical model of the double-inverted pendulum is derived using the Euler–Lagrange equation, as shown in Equation (1). The derivation follows standard principles in analytical mechanics commonly used for multi-body dynamic systems [20].

$$\begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \ddot{y} + \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = 0. \tag{1}$$

The elements of the above equation are as shown in Equation (2).

$$\begin{aligned} n_1 &= h_1 \cos(\theta_1) + h_2 \cos(\theta_1 + \theta_2), \\ n_2 &= h_2 \cos(\theta_1 + \theta_2), \\ m_{11} &= h_3 + h_6 + 2h_4 \cos(\theta_2), \\ m_{12} &= h_6 + h_4 \cos(\theta_2), \\ m_{21} &= h_6 + h_4 \cos(\theta_2), \\ m_{22} &= h_6, \\ r_1 &= -h_4 \sin(\theta_2)(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) \\ &\quad -h_5 \sin(\theta_1) - h_7 \sin(\theta_1 + \theta_2) + c_1\dot{\theta}_1, \\ r_2 &= h_4 \sin(\theta_2)\dot{\theta}_1^2 - h_7 \sin(\theta_1 + \theta_2) + c_2\dot{\theta}_2. \end{aligned} \tag{2}$$

h_1 through h_7 are defined in Equation (3), where g denotes the gravitational acceleration, which is valued at 9.81 [m/s^2].

$$\begin{aligned} h_1 &= m_1 l_1 + m_2 L_1, \\ h_2 &= m_2 l_2, \\ h_3 &= I_1 + m_1 l_1^2 + m_2 L_1^2, \\ h_4 &= m_2 L_1 l_2, \\ h_5 &= g(m_1 l_1 + m_2 L_1), \\ h_6 &= I_2 + m_2 l_2^2, \\ h_7 &= g m_2 l_2. \end{aligned} \tag{3}$$

Equation (1) can be rearranged and rewritten in the form presented in Equation (4).

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = -\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}^{-1} \left(\begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \ddot{y} + \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \right). \tag{4}$$

By solving Equation (4), the expressions for $\ddot{\theta}_1$ and $\ddot{\theta}_2$ can be derived as shown in Equation (5).

$$\begin{aligned} \ddot{\theta}_1 &= \frac{(-m_{22}n_1 + m_{12}n_2)\ddot{y} + (-m_{22}r_1 + m_{12}r_2)}{\Phi}, \\ \ddot{\theta}_2 &= \frac{(m_{21}n_1 - m_{11}n_2)\ddot{y} + (m_{21}r_1 - m_{11}r_2)}{\Phi}, \\ \Phi &= m_{11}m_{22} - m_{12}m_{21}. \end{aligned} \tag{5}$$

At this point, in Equation (5), the state vector is defined as $x_1 = y$, $x_2 = \theta_1$, $x_3 = \theta_2$, $x_4 = \dot{y}$, $x_5 = \dot{\theta}_1$, $x_6 = \dot{\theta}_2$ and $\ddot{y}$ is represented as the acceleration u. Finally, the model equation of the double-inverted pendulum can be expressed as a nonlinear state equation as shown in Equation (6).

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ u \\ \frac{(-m_{22}n_1+m_{12}n_2)u+(-m_{22}r_1+m_{12}r_2)}{\Phi} \\ \frac{(m_{21}n_1-m_{11}n_2)u+(m_{21}r_1-m_{11}r_2)}{\Phi} \end{bmatrix}}_{f(x,u)}. \tag{6}$$

The derived model equation assumes that the cart moves strictly along the horizontal (j-axis) direction with no lateral displacement or rotational motion. Additionally, the first and second pendulums are constrained to rotate exclusively about the i-axis at their respective hinges, meaning that no unintended deviations or out-of-plane motions are considered. The model incorporates only viscous friction, which is proportional to velocity, while neglecting nonlinear static and Coulomb friction to simplify dynamic modeling and maintain analytical tractability.

To ensure high-fidelity sim2real transfer, the mechanical design must closely adhere to these assumptions. Any deviation from these constraints—such as unintended mechanical play, excessive friction, or out-of-plane motion—can introduce discrepancies between the simulated and real-world system, potentially degrading the controller's performance. Therefore, precise mechanical implementation is essential to accurately reflect the modeled dynamics and achieve reliable real-world deployment.

3.2. Hardware Design for High Model Fidelity

To design the actual system with high fidelity to the model equation, it must be constructed to exhibit only the movements that align with the assumptions used in the model equation. If the actual system exhibits movements or behaviors not accounted for in the model, discrepancies will arise between the dynamic responses of the simulation environment and the real system. This subsection proposes a double-inverted pendulum system designed to minimize the reality gap by enhancing fidelity between the model and the actual system responses. This is achieved by aligning the mechanical structure as closely as possible with the assumptions used in the model equation. The proposed mechanical structure of the double-inverted pendulum system is illustrated in Figure 3.

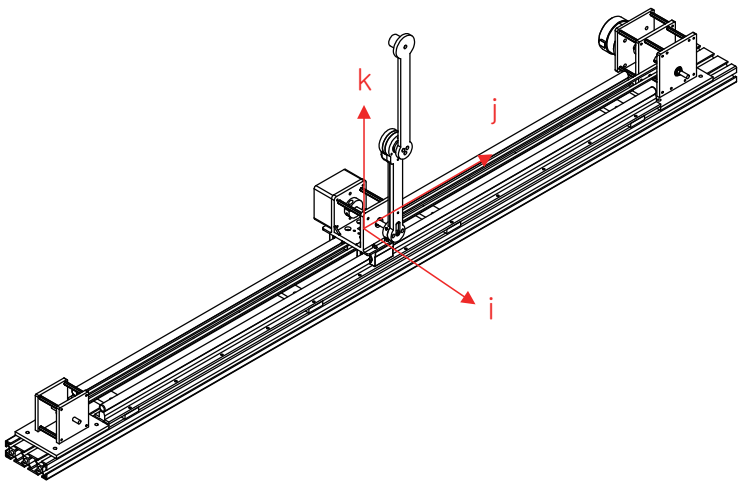

Figure 3. Mechanical structure of the double-inverted pendulum.

3.2.1. Actuator Part Design

Figure 4 depicts a design where the pulley is directly coupled to the actuator. Using an actuator with a gearbox can exacerbate the limit cycle problem caused by backlash. To ad-

dress this, the proposed design eliminates backlash by employing a BLDC motor without a gearbox to drive the pulley directly, thereby resolving the limit cycle issue. The shaft passing through the pulley is supported by two bearings, which ensures that belt tension is transferred exclusively to the shaft. This configuration prevents additional load on the BLDC motor, enhancing the cart's velocity control performance. Furthermore, the actuator is encased in high-rigidity aluminum alloy (aluminum 6061) plates, preventing structural damage or deformation and eliminating the introduction of elements not accounted for in the model equation.

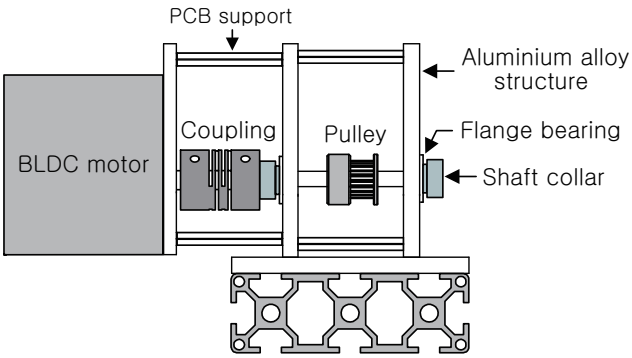

Figure 4. Structure of the proposed actuator.

3.2.2. Design of Cart and Rail System

Figures 5 and 6 illustrate two designs of the double-inverted pendulum rail system for the cart's translational motion. Figure 5 depicts a structure utilizing a V-slot 2040 aluminum extrusion. In this design, the force generated by the pendulum's movement is transmitted to the cart, inducing a twisting angle α in the extrusion, as illustrated in Figure 5. This twisting motion results in the pendulum rotating about the j-axis, as shown in Figure 2. However, such rotation violates the assumptions made in the model equation, thereby reducing the system's model fidelity.

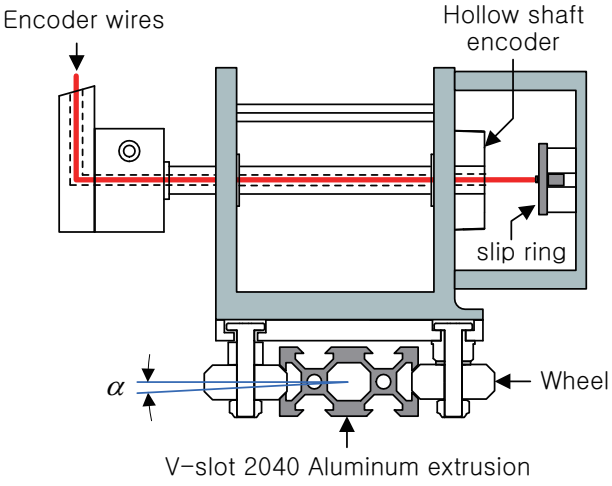

Figure 5. Cart and rail system using 2040 aluminum extrusion.

To address this issue, this paper adopts a design with a dual linear guide rail as shown in Figure 6. In this revised structure, the twisting angle of the rail, such as α, is effectively eliminated. Additionally, the encoder wire used to measure the pendulum's rotation is routed through a slip ring via a hollow shaft, ensuring that the wire's tension does not interfere with the pendulum's rotational motion.

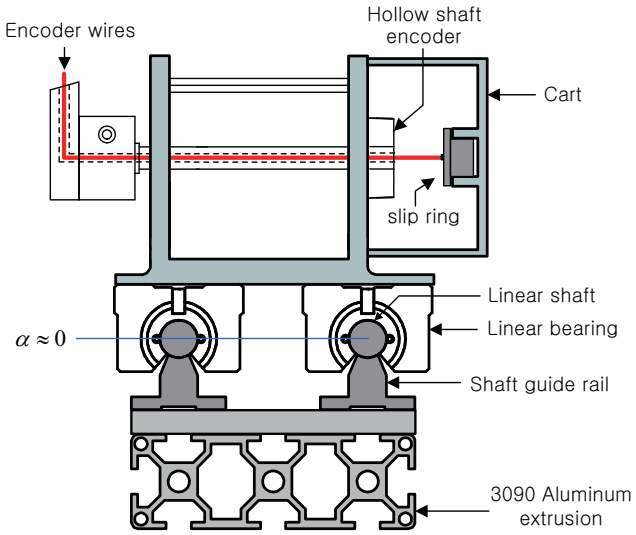

Figure 6. Enhanced rail system with 3090 aluminum extrusion and dual guide rails.

3.2.3. Design of Pendulum and Revolute Joint

Figure 7 provides an exploded view illustrating the conceptual coupling method between the first and second pendulums of the proposed inverted pendulum system. As depicted in the figure, the revolute joint connecting the two pendulums employs a dual bearing structure instead of a single bearing, ensuring rotation primarily about the *i*-axis. To measure the relative rotation angle θ_2 of the second pendulum with respect to the first, a compact magnetic encoder is utilized, enabling the miniaturization of the coupling area between the pendulums. Additionally, the first pendulum is secured to a shaft collar, which is clamped to the rotational shaft using surface pressure. This design provides a robust and reliable coupling between the pendulum and the shaft.

3.2.4. Friction Reduction Design

The double-inverted pendulum system employs wheels for the cart's translational motion, idler pulleys for the actuator, and bearings for the revolute joints of the pendulums. According to the model equation, only viscous friction components proportional to velocity and angular velocity are considered, while static friction and Coulomb friction are excluded. Therefore, the actual system must be manufactured to exhibit similar frictional characteristics.

Factory-shipped bearings are typically coated with high-viscosity grease to ensure durability over long-term use. However, using unmodified bearings in the construction of the double-inverted pendulum introduces static friction and Coulomb friction along with a high viscous friction coefficient. This results in increased actuator load requirements and impairs the system's velocity control performance. Additionally, excessive static friction in the revolute joint bearings leads to inconsistencies in the initial state values when setting the pendulum to its starting configuration ($\theta_1 = -\pi, \theta_2 = 0$), causing a limit cycle phenomenon due to initial state errors.

To address these issues, we adopt a method to significantly reduce friction by cleaning the grease from the bearings using a solvent and reapplying a specialized low-viscosity bearing oil.

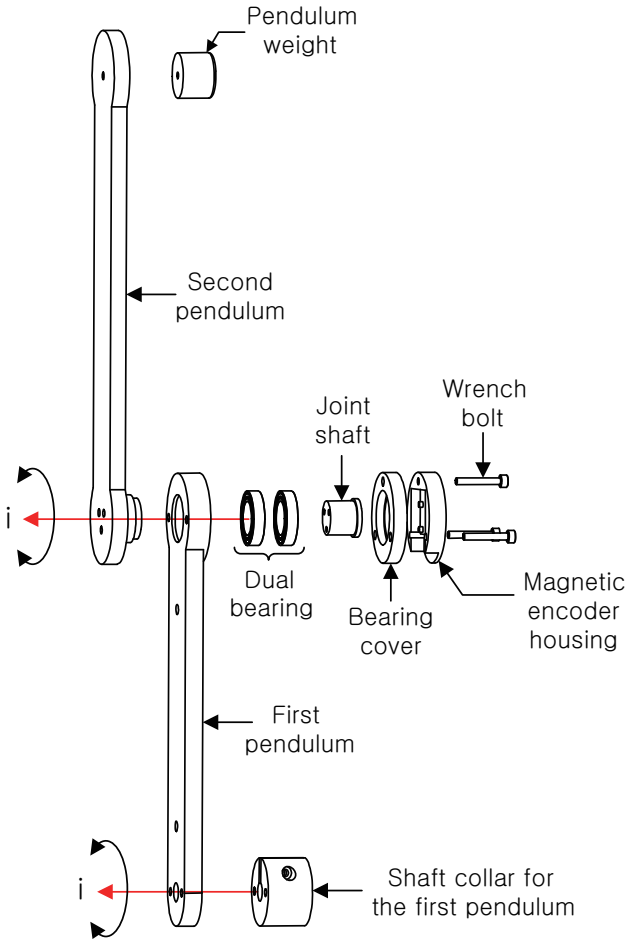

Figure 7. Exploded view of the double-inverted pendulum system.

4. Experimental Methods and Results

In this section, we conduct experiments to design a reinforcement learning-based controller with recovery characteristics utilizing the sim2real technique. The controller is developed using the previously described model equation and the double-inverted pendulum system, which was designed to achieve high fidelity to the model. It is then implemented on the actual system to validate its effectiveness.

4.1. Experimental Setup

The reinforcement learning environment, where the agent interacts directly for training, is implemented as a Python-based simulation built on the mathematical model described in Section 3. The physical parameters of the double-inverted pendulum used to construct the simulation environment are listed in Table 1. The ordinary differential equations (ODEs) are solved using the ode4–Runge–Kutta method.

In the simulation's learning environment, each episode lasts 10 s, and the ODE solver performs calculations at a timestep of 1 ms. The agent observes state information every 10 ms, allowing it to interact with the environment up to 1000 times per episode. Through these interactions, the agent incrementally improves its action policy. In addition to the termination condition where the timestep exceeds 1000, an early termination is triggered if the cart displacement y exceeds 0.4 [m]. This is to prevent scenarios in which the cart exceeds the operational range of the rail in the actual system, ensuring that the trained controller can function reliably when deployed on the real system.

Table 1. Physical parameters of the double-inverted pendulum system used in the experimental study.

Parameter	Link	
	$i = 1$	$i = 2$
m_i	0.2351 [kg]	0.1452 [kg]
I_i	0.0012 [kgm^2]	0.0010 [kgm^2]
l_i	0.0667 [m]	0.1288 [m]
L_i	0.1645 [m]	-
c_i	4.5116×10^{-4}	2.9198×10^{-4}

For real-time implementation and validation of the reinforcement learning-based controller, a Rapid Control Prototyping (RCP) system, independently developed by the authors' research laboratory [21], was employed. This custom-designed RCP system serves as the primary interface for executing the trained control policy on the physical system. It was specifically designed to provide precise control inputs while collecting real-time state feedback from sensors, enabling accurate policy execution under dynamic conditions.

The data acquisition and control system is structured to maintain high synchronization accuracy between sensing, computation, and actuation. The sampling rate is set to 1 kHz, and synchronization is managed through a MATLAB/Simulink-based RCP library, ensuring deterministic execution of the control loop. This architecture minimizes timing inconsistencies and maintains real-time performance throughout the experimental process.

The reinforcement learning-based controller design framework utilizing this RCP system has been previously described in [22], where its architecture and implementation details were outlined. The experimental setup in this study builds upon the same framework, ensuring consistency in system design and evaluation.

4.2. Implementation of RL-Based Controller

4.2.1. Algorithm

In this study, the reinforcement learning agent used for controller design was implemented using the Truncated Quantile Critics (TQC) algorithm [23]. TQC combines the strengths of Quantile Regression Deep Q-Network (QR-DQN) [24] and Soft Actor–Critic (SAC) [25], leveraging quantile-based value estimation to improve policy stability and learning efficiency. By truncating the upper quantiles of the predicted reward distribution, TQC mitigates overestimation bias, resulting in more reliable value estimates and robust policy updates. Given the highly nonlinear and dynamic nature of the double-inverted pendulum system, selecting an algorithm capable of maintaining stability and precision in continuous control tasks was critical. The multi-head critic structure of TQC further enhances its generalization capabilities, enabling it to adapt effectively to complex transition control problems. Based on these considerations, TQC was chosen to ensure stable learning and precise control execution, maximizing the success rate of training while maintaining robustness across different transition scenarios.

The hyperparameters used to implement this algorithm are summarized in Table 2. To reduce training time, the number of critic networks was decreased. At the same time, the size of the policy network representing the controller was increased to accommodate learning transition control for all four equilibrium points within a single neural network. All other hyperparameters were adopted from the original specifications provided by the authors of [23].

Table 2. Hyperparameters of the Truncated Quantile Critics algorithm employed in the reinforcement learning-based controller.

Hyperparameter	Value
Optimizer	Adam [26]
Learning rate	0.0003
Discount factor (γ)	0.99
Replay buffer size	1×10^6
Number of critics (N)	3 *
Number of hidden layers in critic networks	3
Size of hidden layers in critic networks	512
Number of hidden layers in policy networks	2
Size of hidden layers in 1st policy networks	400 *
Size of hidden layers in 2nd policy networks	300 *
Minibatch size	256
Nonlinearity	ReLU
Target smoothing coefficient (β)	0.005
Number of atoms (M)	25

* The values different from [23].

4.2.2. State and Action

The reinforcement learning agent implemented using the TQC algorithm continuously interacts with the Python-based double-inverted pendulum simulation environment, learning to perform transition control. Upon observing the state s at timestep t, the agent uses the following policy function to determine the action a.

$$\pi(a|s) \quad = \quad P(A_t = a|S_t = s) \tag{7}$$

In the simulation environment, the observable state information of the double-inverted pendulum consists of six state variables $\langle y, \theta_1, \theta_2, \dot{y}, \dot{\theta_1}, \dot{\theta_2} \rangle$, as defined in Equation (6). To enhance normalization and ensure continuity during the learning process, the angular variables θ_1 and θ_2 are represented as $\sin(\theta_i)$ and $\cos(\theta_i)$. For transition control, an additional target equilibrium point $\tau \in 0, 1, 2, 3$ is introduced, where each value of τ corresponds uniquely to one of the four equilibrium points. Based on the value of τ, the agent determines the target equilibrium point for transition control.

As a result, the state information observed by the reinforcement learning agent at each timestep comprises nine variables: $s = \langle y, \sin(\theta_1), \cos(\theta_1), \sin(\theta_2), \cos(\theta_2), \dot{y}, \dot{\theta_1}, \dot{\theta_2}, \tau \rangle$. Based on this state information, the agent determines an action a, which serves as the control input to the system. Specifically, a corresponds to the motor acceleration value u, which is constrained within the range $-15 \leq u \leq 15$ to align with the operating limitations of the actual system actuator.

4.2.3. Reward Function

The reinforcement learning agent improves its action policy based on the reward value at each moment of interaction with the environment. The reward function, used to calculate the reward value, varies depending on the target equilibrium point among the four available in the double-inverted pendulum system, which the transition control aims to reach. Table 3 shows the values of the target angle θ_i^* of each pendulum, which vary according to the state variable τ representing the target equilibrium point of the transition. The reward function for calculating the reward value is selected in the form of Equation (8).

$$\begin{aligned} R_u &= \exp(-0.015|u|), \\ R_y &= \exp(-0.5|y|), \\ R_{\theta_1} &= 0.5 + 0.5 * \cos(\theta_1 - {\theta_1}^*), \\ R_{\theta_2} &= 0.5 + 0.5 * \cos(\theta_2 - {\theta_2}^*), \\ R_{\dot{\theta}_1} &= \exp(-0.02|\dot{\theta}_1|), \\ R_{\dot{\theta}_2} &= \exp(-0.02|\dot{\theta}_2|). \end{aligned} \tag{8}$$

Table 3. Target angles for the first and second pendulums corresponding to each target equilibrium point.

τ	Target Angle	
	θ_1^*	θ_2^*
0	$-\pi$	0
1	$-\pi$	$-\pi$
2	0	$-\pi$
3	0	0

The final reward function is expressed as the product of all its components as follows.

$$Reward = R_u R_y R_{\theta_1} R_{\theta_2} R_{\dot{\theta}_1} R_{\dot{\theta}_2}. \tag{9}$$

The individual components of the reward function are shown in Figures 8 and 9. Each component is normalized to the range [0, 1], ensuring that the total reward calculated as their product also falls within this range. Since each episode consists of up to 1000 timesteps, the maximum possible reward in a single episode is 1000.

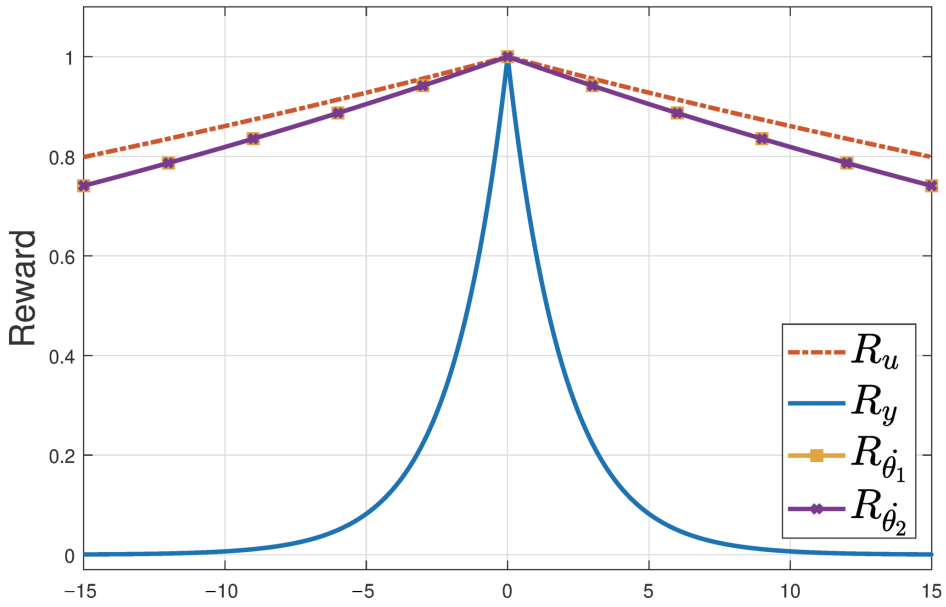

Figure 8. Reward elements independent of target equilibrium point graph.

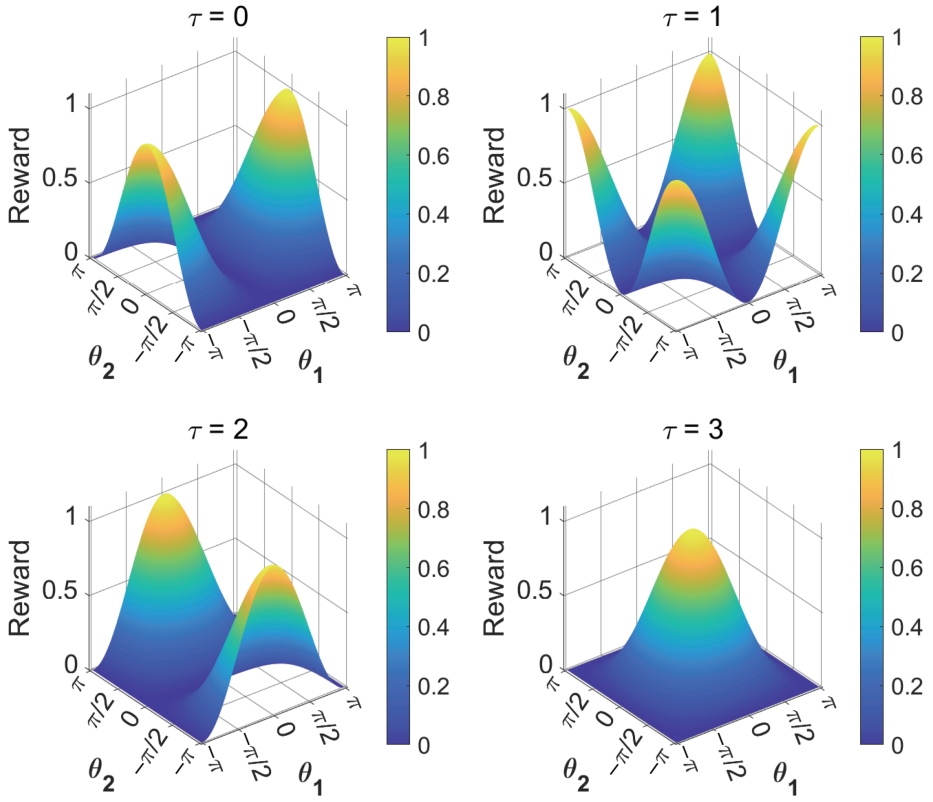

Figure 9. Reward elements dependent on target equilibrium point graph.

The reward function elements can be classified into two categories based on their dependence on the target equilibrium point τ. The first category includes R_u, R_y, $R_{\dot{\theta}1}$, and $R\dot{\theta}_2$, which are independent of the target angle $\theta_i{}^*$ determined by τ. These elements are designed such that the reward increases as their respective values approach zero. Through this mechanism, the reinforcement learning agent learns an action policy that minimizes the control input, maintains the cart position near the origin, and minimizes the motion of each pendulum.

The second category comprises R_{θ_1} and R_{θ_2}, which depend on the target angle $\theta_i{}^*$ specified by τ. As illustrated in Figure 9, these elements yield higher rewards as their values approach the target angle. This encourages the agent to learn an action policy that converges each pendulum to the desired angle, thereby satisfying the equilibrium state defined by the target τ.

4.2.4. Learning Strategy

The reinforcement learning process was conducted on a Windows-based computing system, utilizing an Intel Core i9-13900K CPU, a Z790 Pro RS WiFi motherboard, and 32.0 GB of DDR5 RAM. Training was performed solely on the CPU, requiring 3 million timesteps, which translated to approximately 1.77 days of physical time. The results of this training process are illustrated in Figure 10.

Throughout training, the value of τ was varied in each episode to encourage robustness across different system dynamics. The data illustrated in Figure 10 show that after approximately 500 k timesteps, the average reward across 10 episodes centers around 850. Despite this progress, the rewards do not stabilize at a specific value and continue to show fluctuations. Post-500 k timesteps, rewards ranged from a minimum of 786.8 to a maximum of 898.4, indicating variability in episode outcomes. This variability stems from the experiment's design, where the controller is tested under random initial conditions to ensure it possesses recovery characteristics.

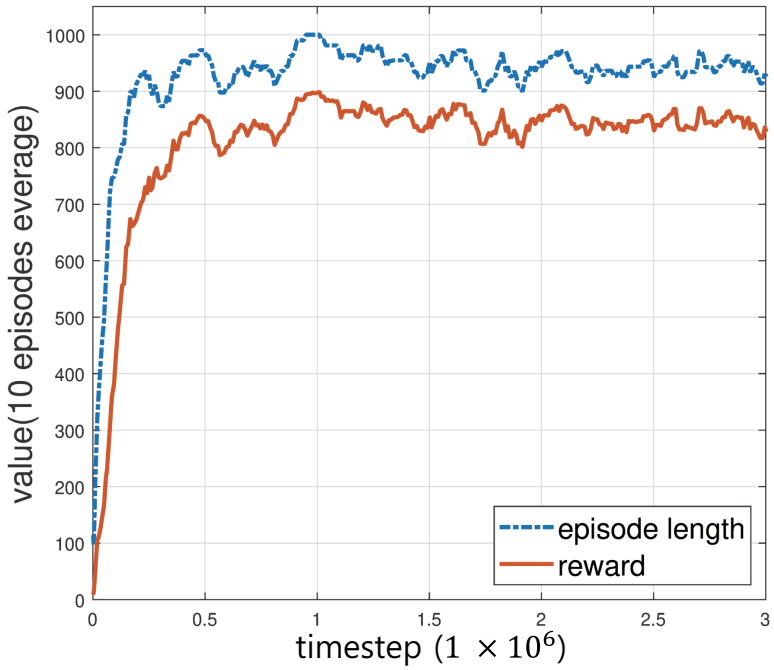

Figure 10. Learning result graph.

To create a learning environment that provides the agent with a wide range of state information, the simulation initializes the state variables $\langle y, \theta_1, \theta_2, \dot{y}, \dot{\theta_1}, \dot{\theta_2} \rangle$ randomly at the start of each episode. The distribution range for this random initialization of each state variable is defined by Equation (10), which is aligned with the operational limits of the actual physical system.

$$\begin{array}{rcl} y & \sim & U(-0.2, 0.2) \\ \theta_1 & \sim & U(-\pi, \pi) \\ \theta_2 & \sim & U(-\pi, \pi) \\ \dot{y} & \sim & U(-1, 1) \\ \dot{\theta_1} & \sim & U(-10, 10) \\ \dot{\theta_2} & \sim & U(-20, 20) \end{array} \tag{10}$$

However, when six random state variables are combined to form a single state, the resulting state information can sometimes violate realistic physical laws. Using such invalid state information as initial conditions for the model equation may lead to outcomes that contradict physical principles and lack validity. From the perspective of the reinforcement learning agent, being presented with unfamiliar and unrealistic state information causes it to generate highly random actions instead of the strategically learned ones. Such aimless cart movements quickly activate the simulation's early termination conditions, leading to the premature termination of the episode. Episodes affected by this phenomenon introduce irregularities in the average reward per episode, as their outcomes deviate significantly from typical learning progress.

Nevertheless, this issue is entirely irrelevant when applying the trained controller to the actual system. In real-world systems, violations of physical laws are impossible. Consequently, the agent is never exposed to unrealistic state information and calculates precise control inputs based solely on valid, real-world data.

4.3. Experimental Results and Discussion

4.3.1. Transition Control

To evaluate the performance of the implemented reinforcement learning-based controller and the model fidelity of the physical system, we compare the results of applying the controller to the 12 transition scenarios in the double-inverted pendulum system both in simulation and actual operation. The transition sequence between each equilibrium point follows the flowchart in Figure 11. The experiment is conducted over a total duration of 60 s with the equilibrium point transition period set to 5 s.

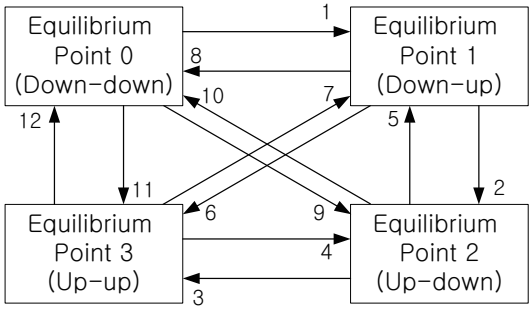

Figure 11. Flowchart of 12 transition control scenarios.

Figure 12 presents the continuous results of the transition control based on the flowchart. The dotted lines indicate the control outcomes in the simulation, while the solid lines represent the corresponding results obtained when applying the same controller to the actual system.

From Figure 12, it can be observed that the dynamic responses of the simulation and the physical system exhibit similar patterns to a meaningful extent. This demonstrates the effectiveness of the designed physical system in achieving high model fidelity, as it closely aligns with the model equation derived in Section 3. In particular, for θ_1 and θ_2, which represent the pendulum angles and serve as the primary control targets in the inverted pendulum system, the dynamic responses of the simulation and the physical system exhibit very high accuracy and nearly identical patterns.

However, in the case of $\dot{y}$ and $\dot{\theta}_1$, the physical system showed oscillations at specific equilibrium points, namely EP1 and EP3. Based on observations during the experiments, this phenomenon is attributed to limitations in sensor accuracy. While the physical system was designed to reflect the assumptions of the model equation as closely as possible, the quantization error caused by the resolution of the encoders measuring the pendulum angles was not accounted for. The encoder attached to the first pendulum has a resolution of 8192cpr (counts per revolution), while the encoder on the second pendulum has a resolution of 4096cpr. Quantization errors arise during the calculation of velocity values, which are negligible during the high-speed transition phase but significantly impact the observed state information during the stabilization phase at low speed after reaching an equilibrium point. Furthermore, EP1 and EP3 are equilibrium points where the second pendulum, the most sensitive component of the system, remains upright. At these points, quantization errors cause slight increases in the control input, leading to larger movements of the second pendulum. The process of correcting these amplified movements results in the ripple phenomena observed during stabilization.

To eliminate these ripple phenomena, two approaches can be considered. First, higher-resolution encoders can be used to reduce quantization errors. Second, software-based techniques such as the M/T method [27], which uses MCUs to mitigate quantization errors, or the Kalman filter [28], which estimates velocity values through model-based filtering, can be employed.

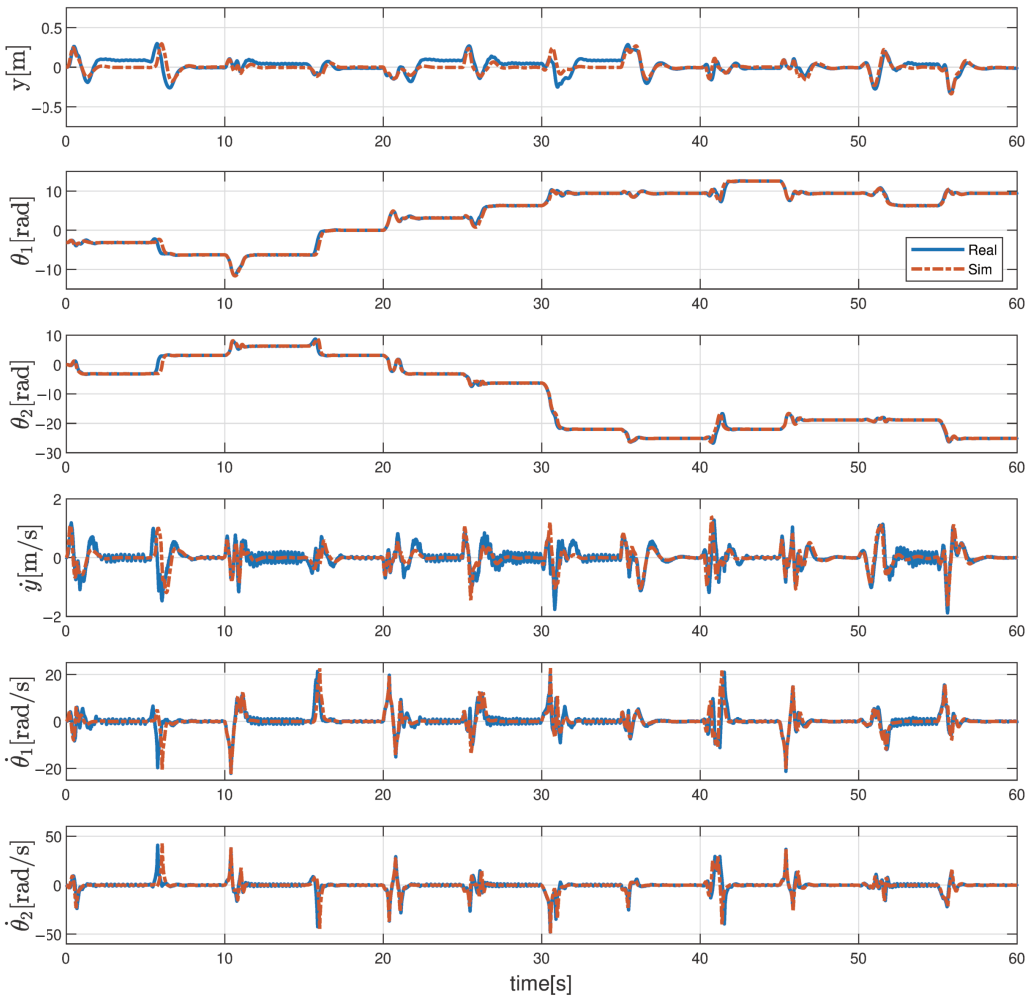

Figure 12. Comparative results of 12 transition control scenarios between the physical system (solid line) and simulation (dotted line).

4.3.2. Recovery Characteristic

Additionally, an experiment was conducted to validate the recovery characteristics of the proposed RL-based controller by applying a strong external disturbance while maintaining a specific equilibrium point. To provide a comparative evaluation, the classical two-degree-of-freedom (2-DOF) control method proposed by [3] was also tested under identical conditions.

Both controllers performed a standard swing-up transition from EP0 to EP3. At approximately 6 s, a strong external disturbance was applied. The results are presented in Figure 13.

For clear visualization, the remainder operation was applied to θ_1 and θ_2, constraining their range to $-\pi < \theta_i < \pi$. The red-dotted vertical line in Figure 13 denotes the moment the external disturbance was introduced. The left panel illustrates the response of the RL-based controller, where the system initially deviates from EP3 due to the perturbation but successfully executes transition control to restore stability.

Conversely, the right panel presents the response of the 2-DOF control method. Upon disturbance, the cart rapidly reaches the rail's physical constraints, rendering further control infeasible. The precomputed feedforward trajectory, which forms the basis of the 2-DOF approach, fails to accommodate the perturbation, and the feedback mechanism lacks sufficient corrective capability, leading to a complete loss of control.

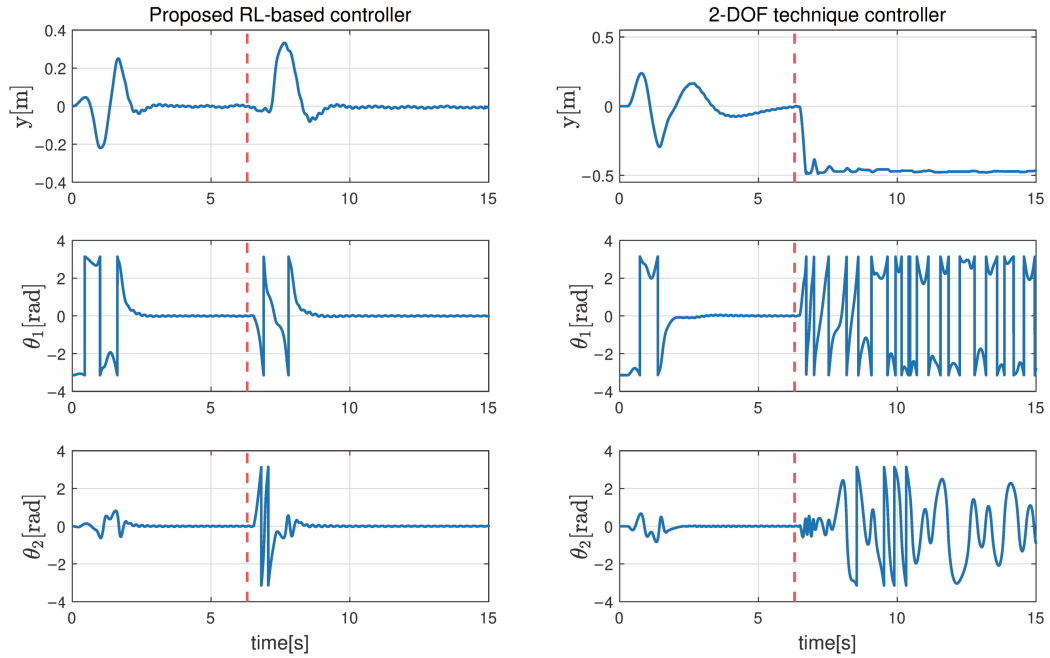

Figure 13. Disturbance application experiment results. Left: RL-based controller. Right: 2-DOF control method.

These results empirically validate the recovery characteristics of the RL-based controller, demonstrating its robustness against significant external disturbances. Unlike the 2-DOF method, which is inherently limited by its offline trajectory optimization and lack of adaptability, the RL-based controller successfully stabilizes the system by leveraging its learned policy, ensuring resilience and effective recovery in real-world applications.

To provide a more intuitive demonstration of the recovery characteristics and transition control performance, a video of the experimental process was recorded and uploaded to the authors' research lab YouTube channel. The Figure 14 provides a link to the video at https://youtu.be/y3Vi17auUpc (accessed on 10 December 2024) (Video title: Reinforcement Learning Transition Control of a Double Inverted Pendulum).

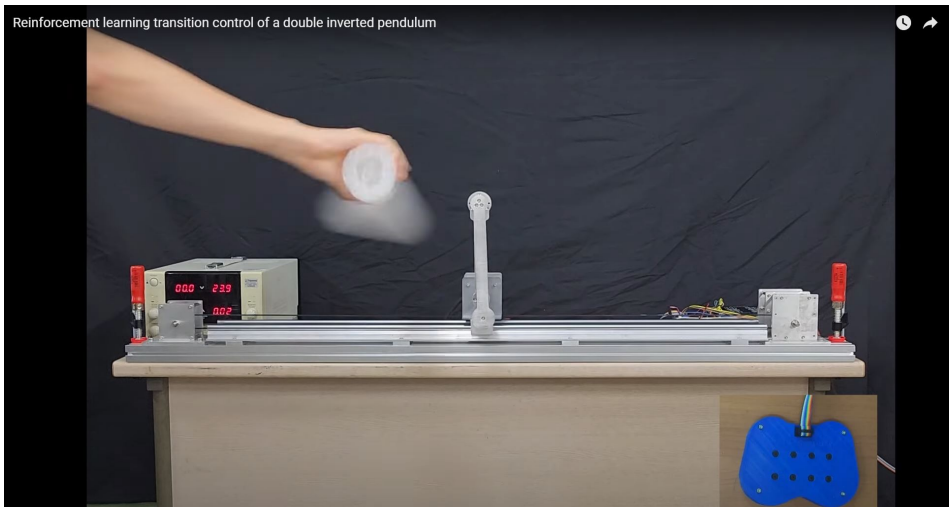

Figure 14. YouTube video recording the experimental results.

In the video, it can be observed that regardless of the disturbance applied to the system, the reinforcement learning-based controller successfully executes transition control, consistently returning the system to the equilibrium point it held prior to the disturbance. Furthermore, repeated disturbance applications at different equilibrium states confirm

the controller's robustness, as the system continuously stabilizes without failure. These empirical results support the findings discussed in the manuscript, demonstrating that the trained controller maintains high recovery capability under various tested conditions provided that the applied disturbances do not exceed the mechanical limits of the system.

5. Conclusions and Future Works

This study demonstrated the successful execution of all 12 transition controls in a double-inverted pendulum system using a reinforcement learning-based controller. The controller exhibited 'Recovery Characteristics', enabling it to stabilize the system and recover equilibrium under strong external disturbances. These achievements mark a significant milestone as the first application of artificial intelligence-based learning techniques for transition control in multi-stage inverted pendulum systems.

To achieve this, the system was designed to minimize the reality gap between simulation and real environments through a hardware-based approach. By aligning the mechanical design of the physical system with the assumptions of the model equation, the study achieved high model fidelity. Consistent dynamic responses were observed when applying the same controller in both simulation and physical environments, validating the robustness of the proposed approach and its ability to facilitate sim2real transfer without additional software-based corrections.

However, as highlighted in the experimental results, the ripple phenomena observed after stabilization remain unresolved. Addressing these issues will require future research that integrates techniques such as the M/T method or Kalman filters, which directly address quantization errors, with methods like domain randomization to improve generalization under uncertainty.

The findings of this study also provide a foundation for extending reinforcement learning-based control to the triple-inverted pendulum system, which is one of the most challenging testbeds in multi-stage inverted pendulum research. While the swing-up problem for the triple-inverted pendulum in a real-world physical system was first solved in 2013 and artificial intelligence-based swing-up control was introduced in 2023, no studies have yet addressed transition control in such systems using AI-based controllers. This study lays the groundwork for developing a reinforcement learning-based controller capable of performing 56 transitions across the eight equilibrium points of a triple-inverted pendulum while maintaining recovery characteristics.

Author Contributions: Conceptualization, T.L. and Y.S.L.; methodology, T.L.; software, T.L.; hardware, D.J.; validation, T.L.; formal analysis, T.L.; investigation, T.L.; writing—original draft preparation, T.L.; writing—review and editing, Y.S.L. and T.L.; visualization, T.L. and D.J.; supervision, Y.S.L.; project administration, T.L.; funding acquisition, Y.S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Inha University Research Grant.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Otani, Y.; Kurokami, T.; Inoue, A.; Hirashima, Y. A swingup control of an inverted pendulum with cart position control. *IFAC Proc. Vol.* **2001**, *34*, 395–400. [CrossRef]
2. Jaiwat, P.; Ohtsuka, T. Real-time swing-up of double inverted pendulum by nonlinear model predictive control. In Proceedings of the 5th International Symposium on Advanced Control of Industrial Processes, Hiroshima, Japan, 28–30 May 2014; pp. 290–295.
3. Graichen, K.; Treuer, M.; Zeitz, M. Swing-up of the double pendulum on a cart by feedforward and feedback control with experimental validation. *Automatica* **2007**, *43*, 63–71. [CrossRef]

4. Kiumarsi, B.; Vamvoudakis, K.G.; Modares, H.; Lewis, F.L. Optimal and autonomous control using reinforcement learning: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 2042–2062. [CrossRef] [PubMed]
5. Zhang, H.; Zhao, C.; Ding, J. Robust safe reinforcement learning control of unknown continuous-time nonlinear systems with state constraints and disturbances. *J. Process Control* **2023**, *128*, 103028. [CrossRef]
6. François-Lavet, V.; Henderson, P.; Islam, R.; Bellemare, M.G.; Pineau, J. An introduction to deep reinforcement learning. *Found. Trends Mach. Learn.* **2018**, *11*, 219–354. [CrossRef]
7. Lai, J.; Xiong, J.; Shu, Z. Model-free optimal control of discrete-time systems with additive and multiplicative noises. *Automatica* **2023**, *147*, 110685. [CrossRef]
8. Dev, A.; Chowdhury, K.R.; Schoen, M.P. Q-Learning Based Control for Swing-Up and Balancing of Inverted Pendulum. In Proceedings of the 2024 Intermountain Engineering, Technology and Computing (IETC), Logan, UT, USA, 13–14 May 2024; pp. 209–214.
9. Israilov, S.; Fu, L.; Sánchez-Rodríguez, J.; Fusco, F.; Allibert, G.; Raufaste, C.; Argentina, M. Reinforcement learning approach to control an inverted pendulum: A general framework for educational purposes. *PLoS ONE* **2023**, *18*, e0280071. [CrossRef] [PubMed]
10. Baek, J.; Jun, H.; Park, J.; Lee, H.; Han, S. Sparse variational deterministic policy gradient for continuous real-time control. *IEEE Trans. Ind. Electron.* **2020**, *68*, 9800–9810. [CrossRef]
11. Gil, Y.; Park, J.-H.; Baek, J.; Han, S. Quantization-aware pruning criterion for industrial applications. *IEEE Trans. Ind. Electron.* **2021**, *69*, 3203–3213. [CrossRef]
12. Dulac-Arnold, G.; Levine, N.; Mankowitz, D.J.; Li, J.; Paduraru, C.; Gowal, S.; Hester, T. Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis. *Mach. Learn.* **2021**, *110*, 2419–2468. [CrossRef]
13. Zhao, W.; Queralta, J.P.; Westerlund, T. Sim-to-real transfer in deep reinforcement learning for robotics: A survey. In Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, Australia, 1–4 December 2020; pp. 737–744.
14. Salvato, E.; Fenu, G.; Medvet, E.; Pellegrino, F.A. Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning. *IEEE Access* **2021**, *9*, 153171–153187. [CrossRef]
15. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 23–30.
16. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; March, M.; Lempitsky, V. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **2016**, *17*, 1–35.
17. Glück, T.; Eder, A.; Kugi, A. Swing-up control of a triple pendulum on a cart with experimental validation. *Automatica* **2013**, *49*, 801–808. [CrossRef]
18. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
19. Jeong, J.; Choi, C.; Ju, D.; Lee, Y.S. A study on the implementation of transition control of double inverted pendulum using LW-RCP. *J. Inst. Control Robot. Syst.* **2023**, *29*, 694–703. [CrossRef]
20. Spong, M.W.; Hutchinson, S.; Vidyasagar, M. *Robot Modeling and Control*, 2nd ed.; John Wiley and Sons: Hoboken, NJ, USA, 2020; ISBN 9781119523994.
21. Lee, Y.S.; Jo, B.; Han, S. A light-weight rapid control prototyping system based on open source hardware. *IEEE Access* **2017**, *5*, 11118–11130. [CrossRef]
22. Lee, T.; Ju, D.; Lee, Y.S. Development environment of reinforcement learning-based controllers for real-world physical systems using LW-RCP. *J. Inst. Control Robot. Syst.* **2023**, *29*, 543–549. (In Korean) [CrossRef]
23. Kuznetsov, A.; Shvechikov, P.; Grishin, A.; Vetrov, D. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 5556–5566.
24. Dabney, W.; Rowland, M.; Bellemare, M.; Munos, R. Distributional reinforcement learning with quantile regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
25. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1861–1870.
26. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
27. Ohmae, T.; Matsuda, T.; Kamiyama, K.; Tachikawa, M. A microprocessor-controlled high-accuracy wide-range speed regulator for motor drives. *IEEE Trans. Ind. Electron.* **1982**, *3*, 207–211. [CrossRef]
28. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; University of North Carolina: Chapel Hill, NC, USA, 1995.