Revista Iberoamericana de Automática e Informática Industrial

Revista Iberoamericana de Automática e Informática Industrial 22 (2025) 253-264

www.revista-riai.org

Controlador por aprendizaje reforzado para un péndulo con rueda inercial experimental

Antonio Concha-Sánchez[a], Brandom J. Jiménez-Hernández[a], Suresh Thenozhi[d], Ramón Jiménez-Betancourt[b], Suresh K. Gadi[c,*]

[a] *Facultad de Ingeniería Mecánica y Eléctrica, Universidad de Colima, Coquimatlán, Colima 28400, México.*
[b] *Facultad de Ingeniería Electromecánica, Universidad de Colima, Manzanillo, Colima 28860, México.*
[c] *Facultad de Ingeniería Mecánica y Eléctrica, Universidad Autónoma de Coahuila, Torreón, Coahuila 27276, México.*
[d] *Facultad de Ingeniería, Universidad Autónoma de Querétaro, Santiago de Querétaro, Querétaro 76010, México.*

To cite this article: Concha-Sánchez, A., Jiménez-Hernández, B., Thenozhi, S., Jiménez-Betancourt, R., Gadi, S. K., 2025, Reinforcement Learning Controller for an Experimental Inverted Pendulum with an Inertial Wheel, Revista Iberoamericana de Automática e Informática Industrial, 22, 253-264. https://doi.org/10.4995/riai.2025.23001

Resumen

En este artículo se aborda el control de estabilización de un péndulo invertido con rueda inercial mediante el diseño de dos enfoques de control. Por un lado, se implementa un controlador convencional que utiliza el Regulador Cuadrático Lineal (LQR, por sus siglas en inglés, *Linear Quadratic Regulator*), y por otro lado, se propone como alternativa no convencional que emplea un controlador basado en aprendizaje por refuerzo (RL, por sus siglas en inglés, *Reinforcement Learning*). Se presenta el diseño mecánico, el modelo matemático y la identificación paramétrica de una plataforma experimental de bajo costo desarrollada para validar los controladores. Además, se diseña un observador de estado utilizando el método de Sylvester para estimar las velocidades del péndulo y de la rueda, necesarias para ambos controladores. El controlador RL utiliza un agente actor-crítico entrenado mediante el algoritmo DDPG (por sus siglas en inglés, *Deep Deterministic Policy Gradient*), basado en el modelo matemático del sistema. Finalmente, se comparan los desempeños de ambos controladores a través de resultados experimentales, concluyendo que el controlador RL logra un menor error en estado estacionario, mientras que el LQR exhibe mejor respuesta transitoria.

Palabras clave: Péndulo con rueda inercial, regulador cuadrático lineal, aprendizaje reforzado, algoritmo DDPG, modelado e identificación del sistema.

Reinforcement Learning Controller for an Experimental Inverted Pendulum with an Inertial Wheel

Abstract

In this article, the stabilization control of an inverted pendulum with an inertia wheel is addressed through the design of two control approaches. On one hand, a conventional controller based on the LQR is implemented, and on the other, a non-conventional alternative is proposed using a RL-based controller. The mechanical design, mathematical modeling, and parameter identification of a low-cost experimental platform developed to validate the controllers are presented. Additionally, a state observer is designed using the Sylvester method to estimate the velocities of the pendulum and the wheel, which are necessary for both controllers. The RL controller employs an actor-critic agent trained with the DDPG algorithm based on the system's mathematical model. Finally, the performance of both controllers is compared through experimental results, demonstrating that the RL controller achieves lower steady-state error, while the LQR exhibits better transient response.

Keywords: Inertia wheel pendulum, linear quadratic regulator, reinforcement learning, DDPG algorithm, system identification and modelling.

*Autor para correspondencia: Research@SKGadi.com

1. Introducción

El control del péndulo invertido es un problema clásico en el ámbito del control automático y la robótica, conocido por su complejidad y por ser un excelente banco de pruebas para evaluar diversas técnicas de control. Un péndulo con rueda inercial es una variante interesante de este sistema, en el que se utiliza una rueda giratoria para generar el torque necesario para mantener el equilibrio del péndulo. Este sistema tiene aplicaciones prácticas en la estabilización de satélites, motocicletas, sistemas de almacenamiento de energía, y robots móviles (Block et al., 2007).

Hidayati and Wasiwitono (2021); Zaborniak et al. (2024); Chinelato et al. (2020) abordan el control del péndulo con rueda inercial mediante técnicas de control lineal tales como el controlador Proporcional Integral Derivativo (PID) y el regulador cuadrático lineal (LQR). Técnicas de control no lineales también se han empleado para estabilizar a este sistema, entre las que destacan control predictivo (Kanjanawanishkul, 2015), basado en pasividad (Hfaiedh et al., 2020), por retroceso o backstepping (Teja et al., 2020), por modos deslizantes (Guo and Liu, 2019), por linealización por retroalimentación (Montoya and Gil-González, 2020), y por moldeo de energía más inyección de amortiguamiento (Sandoval et al., 2022). En este sistema también se han aplicado técnicas de control inteligente, entre las que destacan el control difuso descrito por Chen et al. (2019), así como una red neuronal adaptable dual pospuesta por Zabihifar et al. (2021).

En los últimos años, el aprendizaje por refuerzo (RL) ha ganado relevancia para el control de sistemas dinámicos, inspirándose en los mecanismos de aprendizaje biológico mediante interacción con el entorno (Zai and Brown, 2020). Su aplicación abarca diversos sistemas pendulares, desde el péndulo simple (Demircioğlu et al., 2024) e invertido (Israilov et al., 2023) hasta configuraciones como el doble (Lee et al., 2025), rotatorio (Ho et al., 2025), con rueda inercial (Zhu et al., 2023; Huanlong et al., 2021; Vadlamudi et al., 2024), y balancín con ruedas (Ataç et al., 2021). Estudios comparativos como el de Demircioğlu et al. (2024) demuestran la efectividad de algoritmos avanzados como DDPG, *Soft Actor-Critic* (SAC) y *Twin Delayed DDPG* (TD3), donde DDPG y TD3 emplean políticas deterministas con redes objetivo para estabilidad, mientras que SAC utiliza políticas estocásticas con maximización de entropía. Para el péndulo invertido, Özalp et al. (2020) comparan mediante simulaciones los algoritmos RL *Deep Q-Networks* (DQN), *Double DQN* (DDQN), *Dueling Deep-Q Network* (D3QN), *Reinforce, Asynchronous Advanced Actor Critic Asynchronous* (A3C), y *Synchronous Advantage Actor-Critic* (A2C).

Por otro lado, Israilov et al. (2023) implementan experimentalmente un controlador DQN robusto a ruido y fricción, sin requerir modelado matemático preciso del péndulo invertido. Lee et al. (2025) desarrollan un controlador basado en el algoritmo *Truncated Quantile Critics* (TQC) para un péndulo doble, capaz de transitar entre 12 puntos de equilibrio y recuperarse de perturbaciones, validado experimentalmente. Similarmente, Ho et al. (2025) aplican TD3 para controlar un péndulo rotatorio sin modelo matemático. Para el desafío adicional del péndulo invertido triple, Baek et al. (2024) proponen el método *Virtual Experience Replay* (VER), que mejora la eficiencia del entrenamiento al generar trayectorias virtuales a partir de simetrías geométricas, demostrando robustez frente a incertidumbres del mundo real.

Con respecto al control del péndulo con rueda inercial se encuentra el trabajo realizado por Huanlong et al. (2021), quienes presentan un algoritmo de aprendizaje reforzado basado en el método actor-critico que emplea una función de recompensa dependiente de la posición y velocidad del péndulo; los autores verificaron el comportamiento del controlador por medio de simulaciones. Por otro lado, Vadlamudi et al. (2024) propusieron un controlador con aprendizaje por refuerzo basado en el algoritmo DDPG descrito por descrito por Lillicrap et al. (2015), para el control del equilibrio de motocicletas usando una rueda inercial; los autores modelaron la motocicleta como un péndulo y simularon su control por medio de un agente cuya recompensa depende del ángulo y velocidad angular de la motocicleta, así como del par de control aplicado a la rueda inercial.

También, existen enfoques híbridos que integran aprendizaje por refuerzo con control convencional, adaptativo o robusto, y representan una solución prometedora para sistemas complejos e inestables. Estos métodos aprovechan la solidez teórica de los controladores clásicos y la capacidad adaptativa del RL para manejar incertidumbres y perturbaciones. Un ejemplo destacado es el trabajo de Zhu et al. (2023), que combina RL con un controlador por modos deslizantes en una arquitectura serie-paralela para estabilizar un robot bicicleta en pavimentos curvos, donde un módulo RL ajusta el punto de equilibrio y otro optimiza las ganancias del controlador en línea. Por otro lado, Oliveira et al. (2022) emplean RL para la sintonización inteligente de ganancias en sistemas no lineales de segundo orden. En contraste, Johannink et al. (2019); Ataç et al. (2021) proponen arquitecturas, donde un controlador PID clásico maneja la dinámica conocida y un componente de RL compensa efectos no modelados tales como fricción y perturbaciones. Asimismo, Hernandez et al. (2024) integran RL con control PD y linealización por retroalimentación para el péndulo invertido rotatorio, demostrando la versatilidad de estos enfoques híbridos en sistemas pendulares.

Este artículo presenta el diseño de un controlador LQR tradicional, el cual se compara experimentalmente con un controlador propuesto RL para el sistema de péndulo con rueda inercial. Se describen en detalle el diseño mecánico y el modelo matemático de una plataforma experimental de bajo costo desarrollada para la realización de experimentos. Asimismo, se propone una técnica para estimar los parámetros desconocidos del sistema utilizando el método de mínimos cuadrados. Una vez obtenidos, estos parámetros se emplean tanto para simular el sistema como para el diseño del controlador LQR y de un observador de estado usando el método de Sylvester. Este observador estima las velocidades del péndulo y de la rueda, necesarias para el correcto funcionamiento de ambos controladores.

El controlador RL propuesto utiliza un agente con el enfoque actor-crítico, el cual es entrenado mediante el algoritmo DDPG que combina la eficiencia del aprendizaje determinístico con la capacidad de las redes neuronales profundas para aproximar funciones complejas, permitiendo así el aprendizaje de políticas de control óptimas en entornos continuos. Para el en-

trenamiento del agente RL, se emplea el modelo matemático del sistema, lo cual permite al agente optimizar su política de acción mediante simulaciones antes de ser implementado en la plataforma experimental. A diferencia de los trabajos de Huanlong et al. (2021); Vadlamudi et al. (2024), el controlador propuesto se prueba experimentalmente y emplea todo el estado del sistema, así como la señal de control en la función de recompensa del agente. Esto resalta la aplicabilidad práctica y la robustez del enfoque propuesto, trasladando el conocimiento teórico a un entorno experimental real.

El artículo está organizado como sigue. La sección 2 describe la plataforma de péndulo con rueda inercial de bajo costo desarrollada para comparar experimentalmente las técnicas de control propuestas. Por otro lado, la sección 3 introduce el modelo matemático que describe el comportamiento del sistema, mientras que la sección 4 detalla la técnica diseñada para la estimación de sus parámetros. Posteriormente, la sección 5 introduce los controladores propuestos, y la sección 6 aborda el diseño de un observador de estados para estimar las velocidades del péndulo y la rueda inercial. Las secciones 7 y 8 comparan el desempeño de los controladores por medio de simulaciones y experimentos, respectivamente. Finalmente, la sección 9 establece las conclusiones de este trabajo.

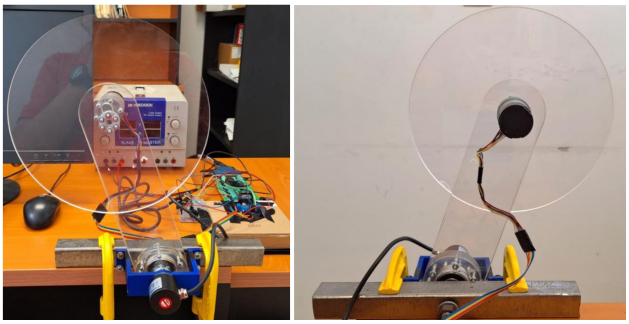


Figura 1: Péndulo con rueda inercial desarrollado.

2. Plataforma de péndulo con rueda inercial

La Figura 1 presenta la plataforma desarrollada, construida con láminas de acrílico de 6 mm de grosor. El péndulo tiene una longitud de 30.14 cm y un ancho de 7.34 cm; además, está compuesto por dos láminas de acrílico unidas, formando un espesor total de 12 mm. El extremo inferior del péndulo está acoplado a un eje que se conecta a un encoder modelo E38S6G5 de 2400 pulsos por revolución (PPR), utilizado para medir la posición angular del péndulo. El encoder está montado sobre un soporte de plástico azul que limita el rango de movimiento del péndulo a aproximadamente ±15 grados con respecto a la vertical. En el otro extremo del péndulo se encuentra un motorreductor de corriente directa (CD) Pololu 37D, con una relación de engranajes de 19:1, equipado con un encoder de 1200 PPR. Este motor acciona una rueda de inercia de 6 mm de espesor y 12 cm de radio, cuya función es mantener el péndulo en posición vertical. El costo del sistema, que incluye el péndulo, la rueda inercial y el motor, es de aproximadamente 100 USD, desglosado de la siguiente manera: 5 USD para el encoder, 80 USD para el motorreductor y 15 USD para la lámina de acrílico. Cabe destacar que las señales de los encoders se adquieren a través de una tarjeta de adquisición de datos *National Instruments* (NI) PCIe-6321. Esta tarjeta también genera la señal de control para el sistema, y se utiliza un driver de puente H modelo L298N como circuito de interfaz entre la tarjeta NI y el motor. La programación del control del sistema se realiza en *Matlab-Simulink*, mientras que la comunicación entre la tarjeta PCIe-6321 y una computadora personal se lleva a cabo mediante bloques de la librería *Simulink Desktop Real-Time.*

3. Modelado matemático del sistema

3.1. Modelado del péndulo con la rueda inercial

La Figura 2 ilustra el diagrama de cuerpo libre del sistema, donde θ y ϕ son los ángulos de rotación del péndulo y de la rueda inercial, respectivamente. Por otro lado, ℓ_1 representa la distancia del eje de rotación del péndulo a su centro de masa cm_1, mientras que ℓ_2 es la distancia que existe entre el eje de rotación del péndulo y el centro de masa cm_2 de la rueda inercial. Además, las masas del péndulo y de la rueda están representadas como m_1 y m_2, mientras que los momentos de inercia del péndulo y de la rueda con respecto a sus centros de masa se denotan como I_{cm1} e I_{cm2}, respectivamente.

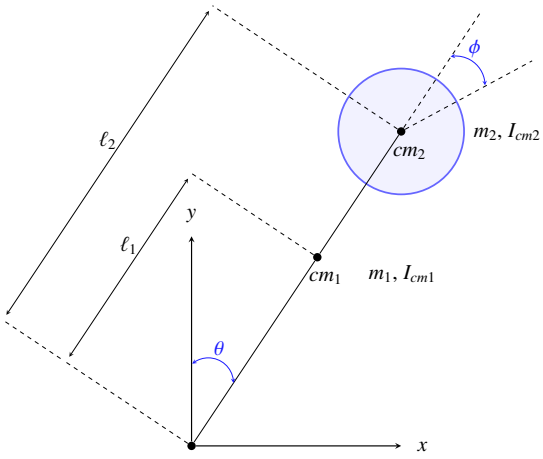


Figura 2: Diagrama de cuerpo libre del sistema.

Para la obtención del modelo matemático del sistema se emplea el formalismo de Euler-Lagrange, que requiere de la energía cinética K y potencial del sistema U representadas como

$$K = K_1 + K_2, \quad U = U_1 + U_2$$

donde K_1, U_1 y K_2, U_2 son la energía cinética y potencial del péndulo y de la rueda inercial, respectivamente, las cuales están dadas por:

$$K_1 = \frac{1}{2}\dot{\theta}^2\left(m_1\ell_1^2 + I_{cm1}\right), \quad K_2 = \frac{1}{2}\left[m_2\ell_2^2\dot{\theta}^2 + (\dot{\theta}+\dot{\phi})^2 I_{cm2}\right] \tag{1}$$

$$U_1 = mg\ell_1\cos\theta, \quad U_2 = mg\ell_2\cos\theta \tag{2}$$

donde g es la aceleración de la gravedad.

Sea L el Lagrangiano definido como

$$L = K - U \tag{3}$$

Entonces, el modelo matemático del sistema se obtiene por medio de las siguientes ecuaciones de Euler-Lagrange

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta}}\right] - \frac{\partial L}{\partial \theta} = 0, \qquad \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\phi}}\right] - \frac{\partial L}{\partial \phi} = \tau \tag{4}$$

donde τ es el par aplicado por el motor acoplado a la rueda inercial.

Resolviendo las ecuaciones en (4) produce

$$\beta_1\ddot{\theta} + \ddot{\phi}I_{cm2} - \beta_2 \sin\theta = 0 \tag{5a}$$
$$(\ddot{\theta} + \ddot{\phi})I_{cm2} = \tau \tag{5b}$$

donde

$$\beta_1 = m_1\ell_1^2 + m_2\ell_2^2 + I_{cm1} + I_{cm2}, \quad \beta_2 = (m_1\ell_1 + m_2)g \tag{6}$$

Cabe mencionar que el eje del péndulo se acopla a su base mediante una articulación compuesta por un tornillo de 1/2 pulgada de diámetro y una tuerca. La tuerca no se aprieta, su función principal es evitar que el péndulo se desprenda del eje. Esta configuración hace que la fricción en el eje sea despreciable y no afecte la dinámica del sistema. Por ello, en el modelo se decidió omitir este término.

3.2. Modelado del motorreductor de cd

El modelo matemático del motor reductor que hace girar la rueda inercial está dado por:

$$J_m\ddot{q} + \left(f_m + \frac{k_a k_b}{R_a}\right)\dot{q} + \frac{\tau}{\rho^2} = \frac{k_a}{\rho R_a}u \tag{7}$$

donde q es la posición en el eje de la salida de la caja de engranes, el cual está conectado a la rueda inercial. Por lo anterior, $q = \phi$. Además, u es el voltaje de alimentación del motor, τ es el par aplicado a la carga, el cual es igual a la expresión en (5b). Por otro lado, los parámetros R_a y L_a son la resistencia e inductancia de la armadura, mientras que ρ, f_m, J_m, k_a y k_b son la relación de reducción de la caja de engranes, el coeficiente de fricción viscosa del eje del motor, el momento de inercia del rotor, así como la constante de par y de fuerza contraelectromotriz, respectivamente. Es importante mencionar que el modelo matemático en (7) se obtiene suponiendo que la inductancia de armadura del motor es despreciable, es decir, $L_a \approx 0$, lo cual es razonable para este sistema.

La ecuación (7) se puede reescribir en forma compacta como:

$$\ddot{\phi} + a\dot{\phi} + c\tau = bu \tag{8}$$

donde se ha sustituido ϕ por q, y los parámetros a, b y c están dados por:

$$a = \frac{f_m}{J_m} + \frac{k_a k_b}{R_a J_m}, \qquad b = \frac{k_a}{\rho R_a J_m}, \qquad c = \frac{1}{\rho^2 J_m} \tag{9}$$

3.3. Modelo matemático del sistema completo

Sustituyendo τ de (8) en (5b), se obtiene el siguiente modelo del sistema conformado por el subsistema mecánico péndulo-rueda inercial y por el subsistema del motorreductor

$$\beta_1\ddot{\theta} + I_{cm2}\ddot{\phi} - \beta_2 \sin\theta = 0 \tag{10a}$$
$$cI_{cm2}\ddot{\theta} + \beta_3\ddot{\phi} + a\dot{\phi} = bu \tag{10b}$$

donde $\beta_3 = cI_{cm2} + 1$.

El péndulo se opera alrededor de $\theta = 0$ y se cumple la siguiente aproximación $\sin\theta \approx \theta$. Tomando en cuenta está aproximación, las expresiones en (10) se pueden reescribir como:

$$\ddot{\theta} = \beta_4\theta + \beta_5\dot{\phi} + \beta_6 u \tag{11a}$$
$$\ddot{\phi} = \beta_7\theta + \beta_8\dot{\phi} + \beta_9 u \tag{11b}$$

donde

$$\beta_4 = \frac{\beta_2\beta_3}{\beta_{10}}, \qquad \beta_5 = \frac{aI_{cm2}}{\beta_{10}}, \qquad \beta_6 = \frac{-bI_{cm2}}{\beta_{10}},$$
$$\beta_7 = \frac{-c\beta_2 I_{cm2}}{\beta_{10}}, \qquad \beta_8 = \frac{-a\beta_1}{\beta_{10}}, \qquad \beta_9 = \frac{b\beta_1}{\beta_{10}}, \tag{12}$$
$$\beta_{10} = \beta_1\beta_3 - cI_{cm2}^2$$

Sea las variables de estado:

$$x_1 = \theta, \quad x_2 = \dot{\theta}, \quad x_3 = \phi, \quad x_4 = \dot{\phi} \tag{13}$$

Entonces, las dos ecuaciones diferenciales de segundo orden en (11) son equivalentes al modelo en espacio de estados siguiente:

$$\overbrace{\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{bmatrix}}^{\dot{\mathbf{x}}} = \overbrace{\begin{bmatrix}0 & 1 & 0 & 0\\ \beta_4 & 0 & 0 & \beta_5\\ 0 & 0 & 0 & 1\\ \beta_7 & 0 & 0 & \beta_8\end{bmatrix}}^{\mathbf{A}}\overbrace{\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix}}^{\mathbf{x}} + \overbrace{\begin{bmatrix}0\\ \beta_6\\ 0\\ \beta_9\end{bmatrix}}^{\mathbf{B}}u$$
$$\overbrace{\begin{bmatrix}y_1\\ y_2\end{bmatrix}}^{\mathbf{y}} = \overbrace{\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}}^{\mathbf{C}}\overbrace{\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix}}^{\mathbf{x}} + \overbrace{\begin{bmatrix}0\\ 0\end{bmatrix}}^{\mathbf{D}}u \tag{14}$$

4. Estimación de los parámetros del modelo del sistema

La Tabla 1 presenta los valores de los parámetros del péndulo y de la rueda inercial. Los momentos de inercia I_{cm1} y I_{cm2}, así como las distancias a los centros de masa ℓ_1 y ℓ_2, se obtuvieron analíticamente a partir de sus geometrías. Además, el parámetro m_1 se compone de la masa del péndulo, incluyendo su tornillería (0.287 kg), la masa de un cople conformado por 3 láminas de acrílico (0.074 kg), el cual tiene un diámetro de 7.34 cm y conecta el péndulo con su eje de rotación, y la masa del motor de CD junto con su soporte, que se acopla a la rueda inercial (0.206 kg). Por otro lado, el parámetro m_2 corresponde a la masa de la rueda de acrílico, incluyendo su tornillería (0.334 kg), más la masa de un cople de aluminio que une el eje del motor con el de la rueda (0.007 kg).

Tabla 1: Parámetros del sistema.

Parámetro	Valor	Unidad
ℓ_1	0.133	m
ℓ_2	0.221	m
m_1	0.567	kg
m_2	0.341	kg
I_{cm1}	0.0045	kg·m^2
I_{cm2}	0.0024	kg·m^2

4.1. Parámetros del modelo del motor de cd

De acuerdo con (8), el modelo del motor tiene tres parámetros a, b y c, los cuales están definidos en (9). La hoja de datos del motor, la cual está disponible en Pololu Corporation (2020), presenta la gráfica del desempeño del motor, con la cual es posible determinar las siguientes constantes $k_a = 0.158$ Nm/A, $k_b = 0.212$ Vs/rad, $R_a = 2.2\ \Omega$ y $f_m = 3.35x10^{-4}$ Nms. Sin embargo, la inercia del rotor J_m no se proporciona por el fabricante y es desconocida. Por lo anterior, se propone la siguiente metodología para estimar J_m, así como los parámetros a, b y c.

4.2. Estimación de a y b usando el mínimos cuadrados

Supóngase que no se le aplica carga al motor, es decir, τ es cero y la ecuación (8) se reduce a

$$\ddot{\phi}(t) + a\dot{\phi}(t) = bu(t) \tag{15}$$

Las señales $\dot{\phi}$ y $\ddot{\phi}$ no están disponibles, y por ello se propone estimar los parámetros a y b usando solo mediciones de ϕ y de u. Esto es posible al multiplicar (15) por el filtro pasabajas $F(s) = \lambda_2/(s^2 + \lambda_1 s + \lambda_2)$ resultando la siguiente parametrización lineal

$$z(t) = \boldsymbol{\psi}^T(t)\boldsymbol{\theta} \tag{16}$$

donde $z(t) = \mathcal{L}^{-1}\left[s^2F(s)\Phi(s)\right]$, además,

$$\boldsymbol{\psi}(t) = \begin{bmatrix} \mathcal{L}^{-1}\left[-sF(s)\Phi(s)\right] \\ \mathcal{L}^{-1}\left[F(s)U(s)\right] \end{bmatrix}, \quad \boldsymbol{\theta} = \begin{bmatrix} a \\ b \end{bmatrix} \tag{17}$$

Los símbolos $\mathcal{L}$ y $\mathcal{L}^{-1}$ denotan la transformada de Laplace y su inversa, respectivamente. Más aún, $\Phi(s) = \mathcal{L}[\phi(t)]$ y $U(s) = \mathcal{L}[u(t)]$.

Las señales $z(t)$ y $\boldsymbol{\psi}(t)$ en (16) se muestrean cada T_s segundos y se emplean para obtener un estimado $\hat{\boldsymbol{\theta}} = [\hat{a},\ \hat{b}]^\intercal$ de $\boldsymbol{\theta}$ usando el método de mínimos cuadrados recursivo dado por:

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \boldsymbol{P}(k)\boldsymbol{\psi}(k)\varepsilon(k) \tag{18}$$

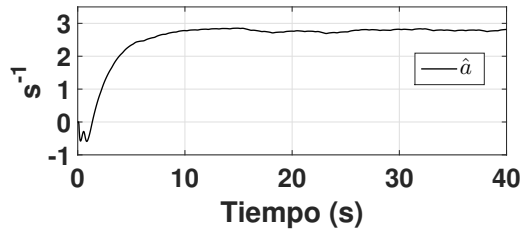
$$\boldsymbol{P}(k) = \frac{1}{\gamma}\left[\boldsymbol{P}(k-1) - \frac{\boldsymbol{P}(k-1)\boldsymbol{\psi}(k)\boldsymbol{\psi}^T(k)\boldsymbol{P}(k-1)}{\gamma + \boldsymbol{\psi}^T(k)\boldsymbol{P}(k-1)\boldsymbol{\psi}(k)}\right] \tag{19}$$

$$\varepsilon(k) = z(k) - \boldsymbol{\psi}^T(k)\hat{\boldsymbol{\theta}}(k-1) \tag{20}$$

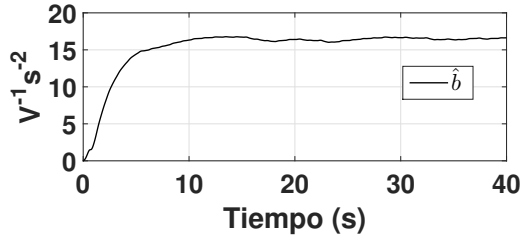
donde $\boldsymbol{P}$ y γ se conocen como matriz de covarianza y factor de olvido, respectivamente. Se selecciona $\lambda = 0.997$, así como el valor inicial de la matriz de covarianza $\boldsymbol{P}(0) = 1000\mathbf{I}_{2x2}$. Además, se aplica la siguiente entrada u de voltaje al motor

$$u(t) = 5\sin(\pi t) + 2\cos(3\pi t)\ \text{V} \tag{21}$$

La Figura 3 presenta los parámetros estimados $\hat{a}$ y $\hat{b}$ producidos por el mínimos cuadrados. Se observa que $\hat{a}$ y $\hat{b}$ convergen a 2.8 y 16.5, respectivamente. De ahora en adelante, se considerará que $a = \hat{a}$ y $b = \hat{b}$ para el cálculo de los parámetros β_i, $i = 1, 2, \ldots, 9$ en (12) y (14).



(a) Parámetro estimado $\hat{a}$.



(b) Parámetro estimado $\hat{b}$.

Figura 3: Parámetros estimados por el mínimos cuadrados.

4.3. Estimación de J_m y c

De la segunda expresión en (9), se deduce que J_m se puede calcular como

$$J_m = \frac{k_a}{rR_ab} = \frac{0.158\ \text{Nm/A}}{(19)(2.2\ \Omega)\left(\frac{16.5}{\text{V s}^2}\right)} = 2.3x10^{-4}\ \text{kg m}^2 \tag{22}$$

Esta inercia J_m se usa para obtener el parámetro c, cuyo valor es

$$c = \frac{1}{(19)^2(2.3x10^{-4}\ \text{kg m}^2)} = 12\ \text{kg}^{-1}\text{m}^{-2} \tag{23}$$

4.4. Matrices **A** y **B** del sistema

Una vez obtenidos todos los parámetros del péndulo, rueda y motor es posible calcular las matrices **A** y **B** del sistema mostradas en (14), las cuales están dadas por:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 44.129 & 0 & 0 & 0.195 \\ 0 & 0 & 0 & 1 \\ -1.235 & 0 & 0 & -2.727 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0 \\ -1.148 \\ 0 \\ 16.07 \end{bmatrix} \tag{24}$$

5. Regulador Cuadrático Lineal

El sistema se controla por medio del Regulador Cuadrático Lineal o LQR. Con este controlador se minimiza el siguiente índice de desempeño (Franklin et al., 2015), proporcionando un control óptimo en términos de compromiso entre la precisión y el esfuerzo de control

$$J = \int_0^\infty (\mathbf{x}^\text{T}\mathbf{Q}\mathbf{x} + ru^2)dt \tag{25}$$

donde r es una constante positiva, mientras que $\mathbf{Q} \in \mathbb{R}^{4\times4}$ es una matriz simétrica semidefinida positiva, es decir, $\mathbf{Q} \geq 0$. Tanto la matriz $\mathbf{Q}$ como r se diseñan como sugiere Franklin

et al. (2015), donde $\mathbf{Q}$ es una matriz diagonal dada por $\mathbf{Q} = \text{diag}[q_{11}, q_{22}, q_{33}, q_{44}]$, cuyos coeficientes y r satisfacen

$$q_{ii} = \frac{1}{\kappa_i^2}, \qquad r = \frac{1}{\kappa_r^2} \tag{26}$$

para $i = 1, \ldots, 4$, donde κ_i es el máximo valor aceptable de $|x_i|$, mientras que κ_r es el máximo valor aceptable de $|u|$.

El índice de desempeño J en (25) se puede escribir como:

$$J = \int_0^\infty g(t)dt, \tag{27}$$

$$g(t) = q_{11}x_1^2(t) + q_{22}x_2^2(t) + q_{33}x_3^2(t) + q_{44}x_4^2(t) + ru^2(t) \tag{28}$$

donde se seleccionaron los siguientes parámetros

$$\begin{gathered}\kappa_1 = 0.0091 \text{ rad} = 0.52\,^\circ, \quad \kappa_2 = 0.35 \text{ rad/s} = 20\,^\circ/\text{s} \\ \kappa_3 = 0.87 \text{ rad} = 50\,^\circ, \quad \kappa_4 = 55.5 \text{ rad/s} = 8.83 \text{ rev/s}, \\ \kappa_r = 3.5 \text{ V}\end{gathered} \tag{29}$$

Sustituyendo los valores en radianes de estos parámetros en (26) resulta

$$\begin{gathered}q_{11} = 12000, \quad q_{22} = 8.21, \quad q_{33} = 1.31, \quad q_{44} = 0.00032 \\ r = 0.082\end{gathered} \tag{30}$$

De esta forma se penaliza fuertemente las desviación $\theta(t)$ del péndulo con respecto a la vertical, así como el uso excesivo de la señal de control $u(t)$.

5.0.1. *Cálculo de la ganancia* $\mathbf{K_{LQR}}$

La señal de control u que minimiza el índice de desempeño J en (25) se calcula como

$$u = -\mathbf{K_{LQR}}\mathbf{x} \tag{31}$$

$$\mathbf{K_{LQR}} = r^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P} \tag{32}$$

donde $\mathbf{P}$ es una matriz definida positiva, tal que $\mathbf{P} = \mathbf{P}^{\mathrm{T}} > 0$, la cual es solución de la ecuación de Riccati siguiente

$$\mathbf{A}^{\mathrm{T}}\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}r^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P} = -\mathbf{Q} \tag{33}$$

Sustituyendo la solución $\mathbf{P}$ de (33) en (32) resulta

$$\mathbf{K_{LQR}} = [-599.75, -70.30, -4, -2.87] \tag{34}$$

Con esta ganancia $\mathbf{K_{LQR}}$, el sistema en lazo cerrado tiene los siguientes polos

$$\mu_{1,2} = -1.91 + 1.64j, \qquad \mu_{3,4} = -16.73 + 12.9j \tag{35}$$

Nótese que con el controlador LQR se tiene un sistema dominante de segundo orden en lazo cerrado con frecuencia natural ω_n y factor de amortiguamiento relativo ζ dados por 2.51 rad/s y 0.76, respectivamente. Con estos parámetros el sistema tiene un máximo sobreimpulso M_p de 2.5 %, así como un tiempo de establecimiento t_s de 2.1 s usando el criterio del 2 % (Ogata, 2010).

5.1. *Controlador por aprendizaje reforzado*

El aprendizaje por refuerzo es un área del aprendizaje automático donde un agente o controlador aprende a tomar acciones de control optimizando una función de recompensa a través de la interacción con el sistema (Sutton and Barto, 2018). A diferencia del controlador LQR descrito en la sección anterior, el agente no recibe instrucciones explícitas sobre qué acciones de control ejecutar, sino que debe descubrirlas a través de prueba y error, recibiendo retroalimentación en forma de recompensas o penalizaciones.

La Figura 4 representa de manera general los componentes del agente en un algoritmo de aprendizaje por refuerzo usando el método actor-critico. Se observa que se compone de dos subsistemas llamados actor y crítico, cuyas funciones se describen en la siguiente sección. Por otro lado, la Figura 5 muestra al agente y su interacción con el entorno, es decir con el sistema péndulo-rueda-motor para su control de balanceo usando el aprendizaje por refuerzo. Al ejecutar una acción de control $u(t)$, el agente cambia el estado del entorno $\mathbf{x}(t)$, que a su vez genera una recompensa $R(t)$ basada en esa acción. Con esta información, el agente puede decidir qué acciones tomar en el futuro. La recompensa $R(t)$ es una señal de retroalimentación proporcionada por el entorno en respuesta a la acción tomada por el agente en un determinado estado $\mathbf{x}(t)$. Es una medida de la bondad de una acción en términos de los objetivos del agente, y se utiliza para guiar el proceso de aprendizaje del agente. Acciones que resultan en recompensas más altas son consideradas mejores.

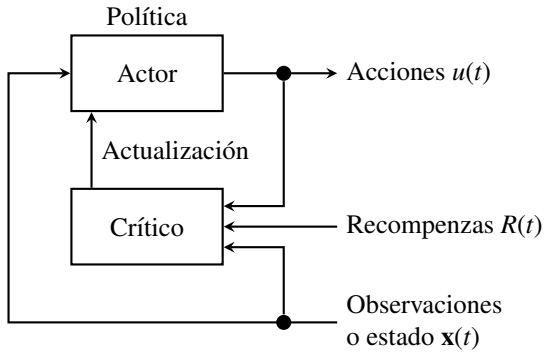


Figura 4: Estructura del agente o controlador.

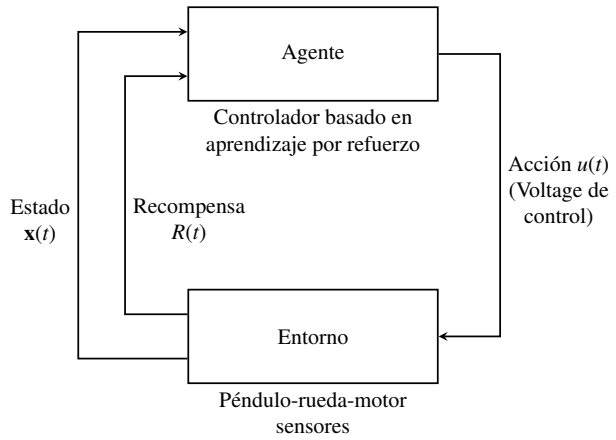


Figura 5: Agente y su interacción con el entorno.

5.1.1. Algoritmo de aprendizaje del agente

Para el aprendizaje del agente se emplea el algoritmo DDPG. Este algoritmo se utiliza principalmente en problemas de control continuo, donde las acciones no son discretas, sino que pueden tomar cualquier valor dentro de un rango. Este algoritmo combina la eficiencia del aprendizaje determinístico con la capacidad de las redes neuronales profundas para aproximar funciones complejas. Se basa en la idea de políticas determinísticas, lo que significa que, para un estado dado, el agente siempre tomará la misma acción. Este enfoque permite el uso de métodos de gradiente para optimizar la política del agente de manera más directa y eficiente (Lillicrap et al., 2015).

El actor aprende la política del agente, es decir, qué acción tomar en cada estado. El actor tiene como entrada el estado $\mathbf{x}(t)$ del entorno y produce como salida la acción $\mu(\mathbf{x}(t))$, como se observa en la Figura 6, donde μ representa la política que la red actor sigue para seleccionar acciones. Para promover la exploración durante el entrenamiento, se añade un término de ruido ϵ, resultando en la acción exploratoria $u(t) = \mu(\mathbf{x}(t)) + \epsilon$. Por otro lado, el crítico ayuda a evaluar la calidad de las acciones tomadas por el actor en términos de las recompensas futuras esperadas. Esto permite que la red actor aprenda a tomar acciones que el crítico estima como las más valiosas.

Con el algoritmo DDPG, tanto los subsistemas actor como crítico del agente están conformados por dos redes neuronales, una denominada principal y la otra objetivo. El DDPG utiliza las redes neuronales objetivo del actor y del crítico para estabilizar el entrenamiento y evitar oscilaciones en las actualizaciones. Estas redes objetivo son copias de las redes actor y crítico principales y se actualizan más lentamente. Los parámetros de las redes neuronales actor y crítico principales se denotan como $\boldsymbol{\theta}_\mu$ y $\boldsymbol{\theta}_Q$, respectivamente. Por otro lado, los parámetros de las correspondientes redes neuronales objetivos se representan como $\boldsymbol{\theta}_{\mu'}$ y $\boldsymbol{\theta}_{Q'}$. La actualización de los parámetros de las redes objetivo depende de la actualización de los parámetros de su correspondiente red principal, siguiendo una tasa de actualización suave para mejorar la convergencia del algoritmo. Dicha tasa de actualización para la red del actor objetivo está dada por $\boldsymbol{\theta}_{\mu'}(k+1) = \xi\boldsymbol{\theta}_\mu(k) + (1-\xi)\boldsymbol{\theta}_{\mu'}(k)$, donde ξ se denomina factor de suavizado de la red objetivo. Esta expresión indica que los nuevos parámetros de la red objetivo son una combinación ponderada de los parámetros actuales de la red principal y de los parámetros anteriores de la red objetivo. Una expresión similar se utiliza para actualizar los parámetros $\boldsymbol{\theta}_{Q'}$ de la red neuronal crítico objetivo. Los detalles matemáticos del algoritmo DDPG se describen por Lillicrap et al. (2015); Matlab (2024).

La red actor se actualiza aprendiendo a tomar mejores decisiones basándose en las evaluaciones de la red crítico. Primero, la red actor sugiere una acción $\mu(\mathbf{x})$ para un estado dado $\mathbf{x}(t)$. Luego la red crítico evalúa esta acción y proporciona una puntuación denotada como $Q(\mathbf{x}(t), \mu(\mathbf{x}))$, que indica su valor en términos de recompensas futuras. La red actor ajusta sus parámetros para maximizar esta puntuación $Q(\mathbf{x}(t), \mu(\mathbf{x}))$, aprendiendo a elegir políticas que la red crítico considera más valiosas. Este proceso se repite muchas veces, ayudando al agente a mejorar su comportamiento en el entorno

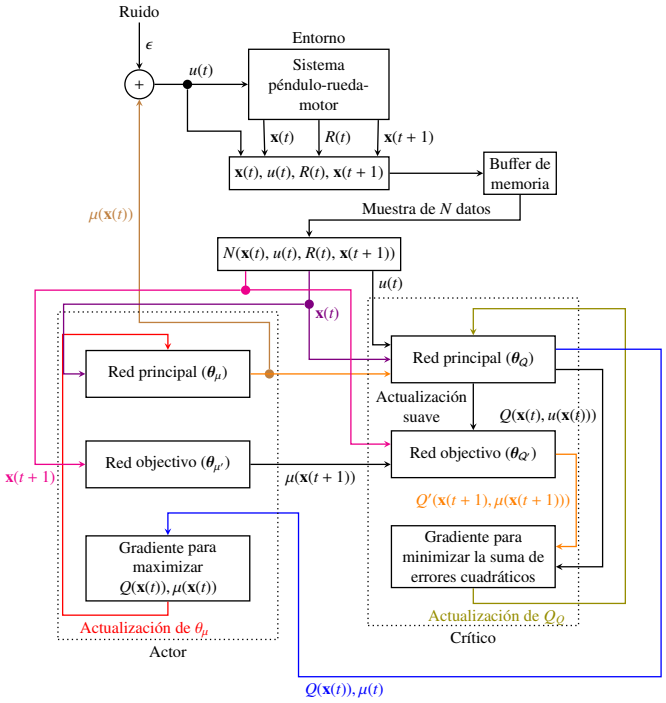


Figura 6: Algoritmo DDPG.

Por otro lado, la red crítico se actualiza para mejorar su capacidad de evaluar la calidad de las acciones tomadas por la red actor. Primero, toma el estado $\mathbf{x}(t)$ y la acción $u(t)$ y predice $Q(\mathbf{x}(t), u(t))$. Luego, tras ejecutar la acción $u(t)$, el agente observa la recompensa inmediata $R(t)$ y el nuevo estado $\mathbf{x}(t+1)$. Usando este nuevo estado y la política $\mu(\mathbf{x}(t+1))$ determinada por la red actor objetivo, la red crítico objetivo predice la próxima recompensa futura descontada $\gamma Q'(\mathbf{x}(t+1), \mu(\mathbf{x}(t+1)))$, donde γ se denomina factor de descuento que pondera la importancia de las recompensas futuras. Se compara la predicción original $Q(\mathbf{x}(t), u(t))$ con la suma de la recompensa inmediata $R(t)$ y la nueva predicción descontada $\gamma Q'(\mathbf{x}(t+1), \mu(\mathbf{x}(t+1)))$ para calcular el error e entre esta diferencia. Finalmente, la red crítico ajusta sus parámetros para minimizar la suma de errores cuadráticos del error e, mejorando su precisión en futuras predicciones.

Además, el DDPG emplea un buffer de memoria o experiencia de repetición, donde se almacenan transiciones individuales del agente en la forma de tuplas $(\mathbf{x}(t), u(t), R(t), \mathbf{x}(t+1))$. Durante el entrenamiento, se seleccionan N muestras aleatorias de este buffer para actualizar las redes actor y crítico. Este enfoque ayuda a romper la correlación temporal entre las experiencias consecutivas y mejora la estabilidad y eficiencia del aprendizaje.

5.1.2. Configuración del algoritmo DDPG

Modelo de simulación. El entrenamiento del agente se realiza fuera de línea para evitar posibles impactos o golpes al sistema físico. Este entrenamiento permite acelerar significativamente el proceso, ya que se lleva a cabo mediante simulaciones extensivas que se ejecutan a una velocidad mucho mayor que la de los experimentos en planta. La idea es entrenar al agente en el entorno simulado y posteriormente transferirlo al sistema físico. Esta transferencia se conoce como aprendizaje de transferencia,

y se espera que el agente mantenga su rendimiento y capacidad de control en el entorno real, gracias a la fidelidad del modelo simulado.

La condición inicial del péndulo durante el entrenamiento es aleatoria con distribución uniforme y puede tomar valores entre ±8°. El uso de condiciones iniciales aleatorias durante el entrenamiento tiene como objetivo mejorar la robustez y generalización del controlador en diferentes escenarios.

Función de recompensa. Se emplea la siguiente función de recompensa para el controlador por aprendizaje reforzado

$$R(t) = -g(t) \tag{36}$$

donde la función $g(t)$ se definió en (27), y es el argumento de la integral correspondiente al índice desempeño a minimizar por el controlador LQR. Nótese que la función $J = \int_0^\infty g(t)dt$ se minimiza con el controlador LQR, mientras que la función $R(t) = -g(t)$ se maximiza con el controlador por aprendizaje reforzado. Se eligió esta recompensa $R(t)$ ya que, de esta forma, se penaliza al estado y a la señal de control con los mismas parámetros q_{ii}, $i = 1, 2, 3, 4$ y r empleados en el controlador LQR, estableciendo así un marco consistente para evaluar y comparar sus desempeños.

Estructura de las redes neuronales actor y crítico. Las redes neuronales actor y crítico del algoritmo DDPG se implementaron como se muestran en la Figuras 7 y 8, respectivamente. Estas redes tienen capas Completamente Conectadas (CC), inicializadas aleatoriamente antes de comenzar el entrenamiento. La salida de varias capas utiliza la Función de Activación (FA) ReLU (*Rectified Linear Unit*). Esta FA introduce no linealidad en las redes neuronales y mitiga el problema del desvanecimiento del gradiente, facilitando la propagación de gradientes en capas profundas y acelerando la convergencia del entrenamiento. Por otro lado, la tercera capa de la red de actor tiene una neurona con función de activación *tanh* para garantizar que la salida tome valores entre -1 y 1. Posteriormente, esta salida se escala a -12 y 12, que es el voltaje de control suministrado al motor acoplado a la rueda inercial. Nótese que en el caso de la última capa de la red crítico, ésta es una neurona con función de activación lineal para no restringir el valor predicho de $Q(\mathbf{x}(t), u(t))$

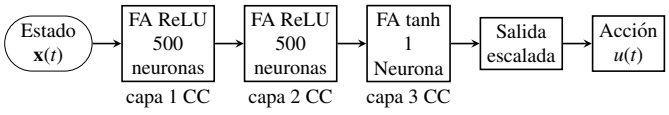


Figura 7: Red neuronal del actor.

Parámetros del entrenamiento del agente. La Tabla 2 muestra los parámetros empleados para entrenar el agente usando el algoritmo DDPG. Como se observa en la tabla, se emplea un factor ξ de suavizado pequeño para garantizar que el algoritmo DDPG se mantenga estable mientras aprende de manera efectiva; con este valor de ξ, las redes objetivo responderán lentamente a los cambios en las redes principales. Este suavizado gradual evita oscilaciones indeseadas al ajustar los pesos de las redes objetivo, contribuyendo a un entrenamiento más predecible.

Por otro lado, se usa un valor γ de descuento de 0.99, lo cual indica que el agente da casi tanto peso a las recompensas futuras como a las inmediatas, promoviendo una estrategia de aprendizaje que favorece las recompensas a largo plazo y contribuye a una mayor estabilidad en el entrenamiento. Esto resulta particularmente útil en sistemas dinámicos como el péndulo con rueda inercial, donde las decisiones actuales afectan significativamente los estados futuros. Asimismo, se selecciona un buffer grande para que el agente capture una amplia variedad de situaciones, lo cual es crucial para un aprendizaje efectivo y generalizado. Además, se utiliza un tamaño de mini-lote de 40, lo cual representa un equilibrio entre introducir suficiente variabilidad en las actualizaciones y mantener la eficiencia computacional. Este tamaño permite actualizaciones frecuentes de los parámetros de la red, mejorando la velocidad de convergencia y la capacidad del agente para aprender de manera robusta y eficiente.

El ruido ϵ utilizado para introducir exploración en las acciones de control $u(t)$ del agente es del tipo *Ornstein-Uhlenbeck* (OU), cuyos parámetros son los que tiene por defecto *Matlab*, que son media cero y desviación estándar de 0.3. Este ruido ϵ induce variabilidad en las acciones para escapar de óptimos locales (Lillicrap et al., 2015). También, ϵ mejora la robustez frente a discrepancias no modeladas, tales como la fricción viscosa. El ruido OU simula una exploración más suave y temporalmente correlacionada en comparación con el ruido Gaussiano, lo cual es apropiado para sistemas continuos y dinámicos como el considerado.

Cabe mencionar que el algoritmo de optimización utilizado para actualizar los parámetros del agente es el Adam (*Adaptive Moment Estimation*), el cual emplea los parámetros empleados por defecto en *Matlab*. Este optimizador se selecciona por su capacidad para manejar gradientes ruidosos y por su convergencia eficiente al utilizar tasas de aprendizaje adaptativas.

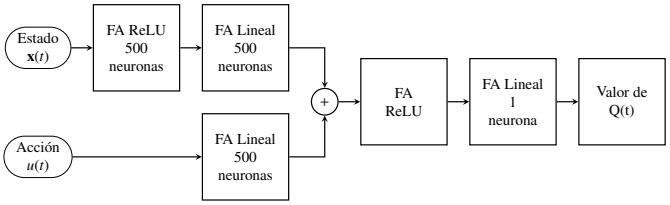


Figura 8: Red neuronal del crítico.

Resultados del entrenamiento del agente. Para el entrenamiento se empleó el bloque RL Agent de la librería *Reinforcement Learning* de *Matlab-Simulink*. La Figura 9 muestra la evolución del promedio de la recompensa R, la cual está marcada con color azul. Se observa que su valor aumenta al llegar a 100 episodios y posteriormente se mantiene oscilando hasta los 400 episodios. Enseguida se incrementa el valor de R hasta estabilizarse alrededor de un valor de -1500 a partir de los 500 episodios. Este aumento en el promedio de recompensas indica que el agente fue aprendiendo a tomar mejores decisiones hasta que convergió a una política óptima. Por otro lado, la gráfica amarilla muestra la evolución de la función Q. También, a partir de los 400 episodios la función Q fue aumentando y posteriormente se estabilizó, indicando que el agente aprendió de manera efectiva y consistente.

Tabla 2: Parámetros de entrenamiento del DDPG.

Parámetro	Valor
Período de muestreo T_s	0.01 s
Factor ξ de suavizado de la red objetivo	0.001
Factor γ de descuento de las recompensas futuras	0.99
Longitud del buffer de experiencias	$2x10^6$
Tamaño N del mini-lote usado para actualizar la red	40
Número máximo de episodios de entrenamiento	850
Tasa de aprendizaje de la red Actor	$5x10^{-4}$
Tasa de aprendizaje de la red Crítico	$1x10^{-3}$
Pasos por episodio	500

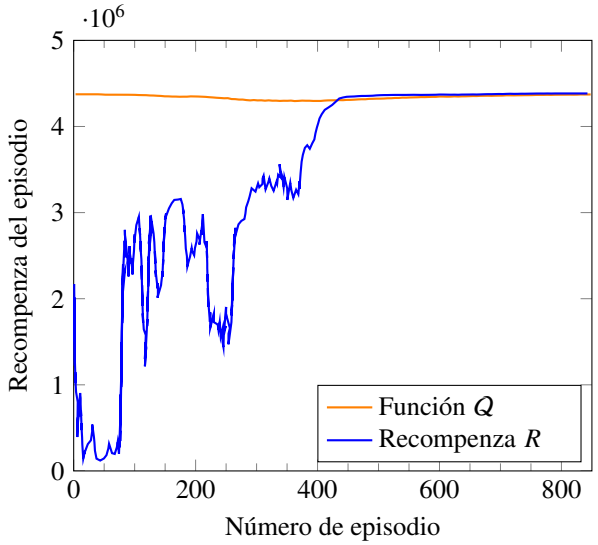


Figura 9: Resultado del entrenamiento del agente.

Cabe mencionar que el entrenamiento del agente se finalizó en 850 episodios, en un tiempo de 4 horas y 20 minutos, utilizando una computadora personal con procesador Intel(R) Core (TM) i7-6700 CPU de 3.4 GHz y 8 GB de memoria RAM. Asimismo, existen alternativas como el uso de hardware especializado con capacidades de procesamiento en paralelo, que podrían reducir significativamente este tiempo de entrenamiento.

6. Observador de estado

Para el diseño de los controladores LQR y RL se supuso que se tiene acceso a todo el estado $\mathbf{x} = [x_1, x_2, x_3, x_4]^T$ del sistema. Sin embargo, en la práctica esto no es posible ya que solo se miden las variables $x_1 = \theta$ y $x_3 = \phi$ mediante encoders. Entonces, se diseña un observador de estado para estimar las variables x_2 y x_4, lo cual es posible ya que el sistema (14) es completamente observable puesto que la siguiente matriz de observabilidad $\mathcal{O}$ es de rango completo $n = 4$ (Kuo, 1996)

$$\mathcal{O} = [\mathbf{C}, \mathbf{CA}, \mathbf{CA^2}, \mathbf{CA^3}]^T \tag{37}$$

El modelo del observador de estado tiene la siguiente expresión

$$\dot{\hat{\mathbf{x}}} = (\mathbf{A} - \mathbf{K_e C})\hat{\mathbf{x}} + \mathbf{B}u + K_e y \tag{38}$$

donde $\hat{\mathbf{x}} = [\hat{x}_1], \hat{x}_2, \hat{x}_3, \hat{x}_4]^T$ es un estimado del vector de estado $\mathbf{x}$, mientras que $\mathbf{K_e}$ es la ganancia del observador, la cual se tiene que diseñar para que la matriz $\mathbf{A} - \mathbf{K_e C}$ tenga los valores propios deseados con parte real negativa garantizando que el error de observación $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$ convergerá a cero.

6.0.1. *Calculo de la ganancia* $\mathbf{K_e}$

Para obtener la ganancia $\mathbf{K_e}$ del observador se usa el método de Sylvester (Espinosa, 2014), con el cual se requiere una matriz diagonal $\Gamma = \text{diag}[\lambda_1, \lambda_2, \lambda_3, \lambda_4]$, donde λ_i, $i = 1, 2, \ldots, 4$ son los polos deseados del observador.

La ecuación de Sylvester está dada por

$$\mathbf{A^T}\mathcal{X} - \mathcal{X}\mathbf{\Gamma} = \mathbf{C^T G} \tag{39}$$

donde $\mathbf{G} \in \mathbb{R}^{2\times 4}$ es una matriz arbitraria.

Si la solución $\mathcal{X}$ de (39) es invertible, entonces la ganancia $\mathbf{K_e}$ del observador se calcula como

$$\mathbf{K_e} = \left(\mathbf{G}\mathcal{X}^{-1}\right)^T \tag{40}$$

En caso de que la matriz $\mathcal{X}$ no sea invertible se debe seleccionar otra matriz $\mathbf{G}$.

Los polos del observador se seleccionaron como $\lambda_1 = -50$, $\lambda_2 = -51$, $\lambda_3 = -52$ y $\lambda_4 = -53$, mientras que $\mathbf{G}$ se eligió como

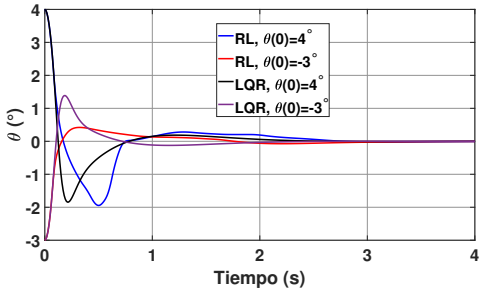
$$\mathbf{G} = \begin{bmatrix} -2 & 5 & 10 & -11 \\ 8 & 1 & -11 & 5 \end{bmatrix} \tag{41}$$

Con esta selección de $\mathbf{G}$, la solución $\mathcal{X}$ resultó invertible y se obtuvo $\mathbf{K_e}$.

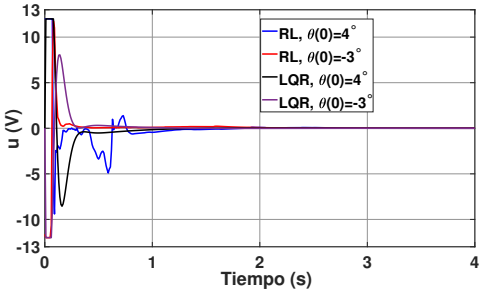
7. Simulaciones

Se realizaron simulaciones de los controladores RL y LQR propuestos, empleando el observador de estado diseñado previamente. Los archivos de las simulaciones se encuentran en (Concha and Gadi, 2025c). Para obtener una simulación más realista, la señal de control del LQR se saturó a ±12 V, lo que corresponde a los límites de voltaje que puede alcanzar el motor acoplado a la rueda inercial. La Figura 10 presenta la respuesta de θ y la señal de control u de ambos controladores, suponiendo que el péndulo se encuentra en reposo y partiendo de las condiciones iniciales $\theta(0) = 4°$ y $\theta(0) = -3°$. Se observa que, con ambos controladores, θ se aproxima a 0° a partir de los 2 s.

Nótese que el péndulo alcanza la posición vertical en un tiempo de subida menor con el controlador LQR que con el RL, lo cual resulta en un sobreimpulso mayor para el LQR cuando la condición inicial es $\theta(0) = -3°$. Además, las señales de control u de ambos controladores alcanzan ±12 V durante aproximadamente 0.1 s, permitiendo que el péndulo se acerque rápidamente a su posición vertical; una vez que éste se encuentra cerca de dicha posición, las señales de control u se reducen a valores próximos a 0 V.



(a) Posición $\theta(t)$ del péndulo.



(b) Señal de control $u(t)$.

Figura 10: Posición del péndulo y señal de control.

8. Experimentos

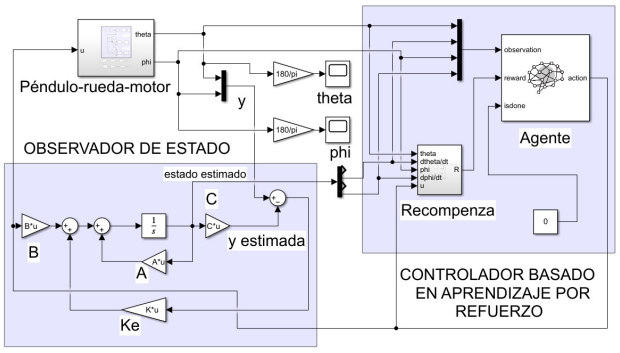
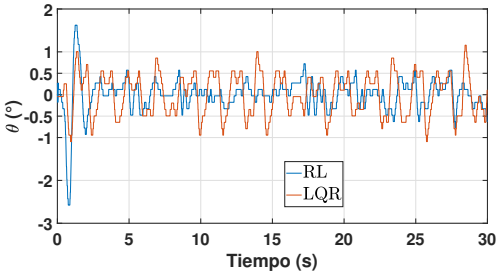


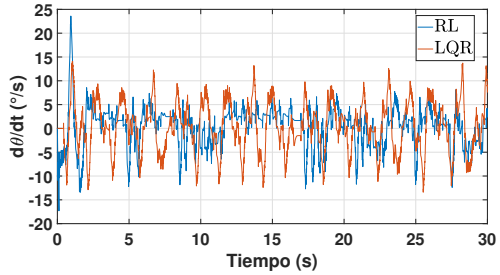
Figura 11: Diagrama de bloques en lazo cerrado para los experimentos.

La Figura 11 presenta el diagrama de bloques en *Simulink* del sistema de control en lazo cerrado usando el aprendizaje por refuerzo. El archivo de este diagrama está disponible para descarga en el repositorio de Concha and Gadi (2025c). En la Figura 11 se puede apreciar el estado $\mathbf{x}$ y la recompensa R empleada por el agente. Además, se muestra el observador de estado diseñado para estimar la velocidad del péndulo $\dot{\theta}$ y de la rueda $\dot{\phi}$.

En las Figuras 12, 13 y 14 se comparan los resultados experimentales de los controladores LQR y RL. A diferencia de las simulaciones, el controlador RL en la práctica genera una respuesta inicial rápida en el péndulo, provocando oscilaciones de alta amplitud que se atenúan después de los 2 segundos. Con RL, el ángulo máximo θ fue de 2.58°, frente a 1.1°del LQR. No obstante, en ambos casos el péndulo se estabiliza cerca de la vertical después de 2 s, lo que coincide con las simulaciones.

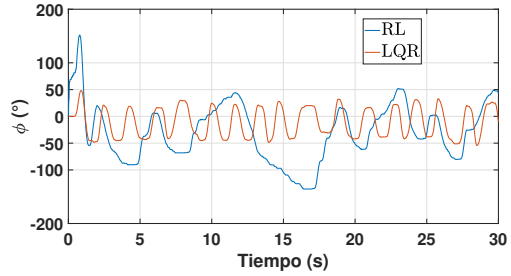


(a) Posición $\theta(t)$ del péndulo.

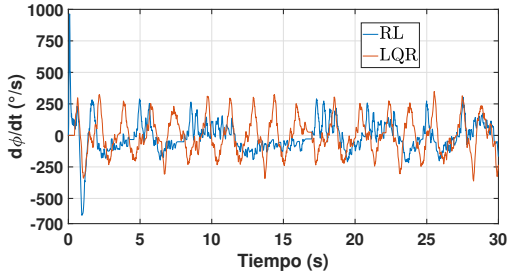


(b) Velocidad $\dot{\theta}(t)$ del péndulo.

Figura 12: Posición y velocidad del péndulo.



(a) Posición $\phi(t)$ de la rueda.



(b) Velocidad $\dot{\phi}(t)$ de la rueda.

Figura 13: Posición y velocidad de la rueda.

Además, en la respuesta de θ se observa el efecto de cuantización del encoder, cuya resolución es de 0.15° (360°/2400). Por otro lado, la Figura 13 muestra que el controlador RL induce una velocidad angular inicial elevada $\dot{\phi}$ en la rueda, ya que la señal de control $u(t)$ satura al voltaje máximo de 12 V para llevar rápidamente el péndulo a la vertical. Este comportamiento explica las oscilaciones transitorias de gran amplitud en el péndulo.

La Tabla 3 presenta los valores RMS y absolutos máximos de las señales θ, $\dot{\theta}$, ϕ, $\dot{\phi}$ y u correspondientes a ambos controla-

Tabla 3: Comparación de señales e índices de desempeño entre los controladores LQR y RL.

	θ_{rms} (°)	$\lvert\theta_{max}\rvert$ (°)	$\dot{\theta}_{rms}$ $\left(\frac{°}{s}\right)$	$\lvert\dot{\theta}_{max}\rvert$ $\left(\frac{°}{s}\right)$	ϕ_{rms} (°)	$\lvert\phi_{max}\rvert$ (°)	$\dot{\phi}_{rms}$ $\left(\frac{°}{s}\right)$	$\lvert\dot{\phi}_{max}\rvert$ $\left(\frac{°}{s}\right)$	u_{rms} (V)	$\lvert u_{max}\rvert$ (V)	J_d	E_u
LQR	0.47	1.16	5.1	13.72	28.87	54.3	142	362	2.54	7.02	43.41	196.65
RL	0.26	0.78	3.62	12.7	58.95	135.6	103	303	2.1	9.28	78.20	163.13

dores. Estos valores fueron calculados utilizando las muestras obtenidas durante el período de 5 a 30 s, así como un período de muestreo de $T_s = 10$ ms. Asimismo, esta tabla presenta los valores de los siguientes índices de desempeño discretos:

$$J_d = \sum_{k=1}^{N} \left(\mathbf{x}[k]^{\mathrm{T}} \mathbf{Q}\, \mathbf{x}[k] + r\, u[k]^2\right) T_s, \qquad E_u = \sum_{k=1}^{N} u[k]^2\, T_s, \tag{42}$$

donde $N = 3000$ es el número total de muestras. El índice J_d representa la versión discreta de la función de costo J definida en (25), evaluando el comportamiento del sistema en el intervalo $t \in [0, 30]$ s, incluyendo la respuesta transitoria. Por otro lado, E_u cuantifica la energía total de la señal de control u durante el experimento.

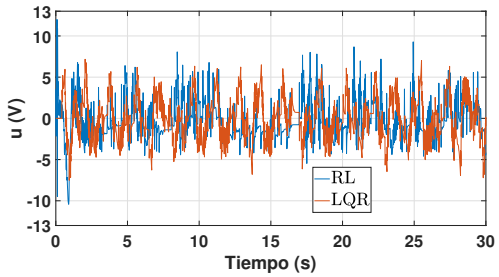
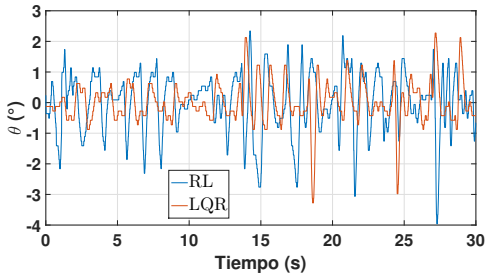
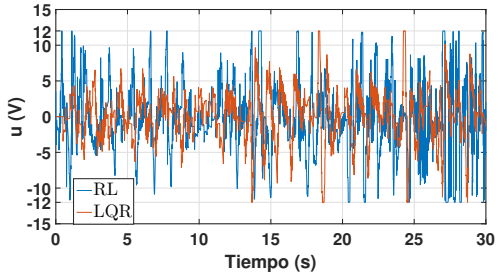


Figura 14: Señal de control $u(t)$.



(a) Posición $\theta(t)$ del péndulo ante perturbaciones.



(b) Señal de control $u(t)$ ante perturbaciones.

Figura 15: Posición del péndulo y señal de control ante perturbaciones.

En la Tabla 3 se observa que los valores θ_{rms} y $|\theta_{max}|$, medidos a partir de 5 s, presentan valores menores con el controlador RL en comparación con el LQR. Asimismo, la señal de control u_{rms} también es más reducida con el controlador RL, aunque $|u_{max}|$ alcanza un valor de 9.28 V, superior al registrado con el LQR. Por otro lado, el índice de desempeño J_d resulta menor para el controlador LQR que con el controlador RL, lo cual refleja una mejor compensación entre la estabilización del péndulo y el esfuerzo de control durante el transitorio. Finalmente, la energía E_u de la señal de control es ligeramente menor con el controlador RL.

8.1. Caso con perturbaciones

Se evaluó el desempeño de ambos controladores RL y LQR al perturbar el péndulo mediante empujones ligeros aplicados manualmente en múltiples ocasiones. Los videos de los experimentos con los controladores RL y LQR están disponibles en (Concha and Gadi, 2025a) y Concha and Gadi (2025b), respectivamente.

La Figura 15 muestra la evolución temporal del ángulo θ del péndulo y la correspondiente señal de control u para ambos controladores. Durante las perturbaciones, la señal u alcanzó repetidamente su valor absoluto máximo de 12 V. Se observa que las perturbaciones generan oscilaciones en el péndulo, y que el amortiguamiento es más rápido con el controlador LQR que con el RL. Este comportamiento sugiere que, ante perturbaciones externas, el controlador LQR regula el ángulo θ con mayor eficiencia que el controlador RL.

9. Conclusiones

En este artículo se propusieron dos estrategias de control para estabilizar un péndulo con rueda inercial en su posición vertical: un controlador basado en aprendizaje por refuerzo (RL) y un controlador LQR. Se describió en detalle la estructura y configuración de las redes neuronales que conforman el agente RL, compuestas por las redes de actor y crítico, y se presentó la evolución del entrenamiento del agente utilizando el algoritmo DDPG junto con el modelo matemático del sistema obtenido mediante el formalismo de Euler-Lagrange. Además, se desarrolló y explicó el diseño de un observador de estado utilizando el método de Sylvester para estimar las velocidades del péndulo y de la rueda inercial, información que se emplea tanto en el controlador LQR como en el agente como parte del estado.

El controlador RL se comparó experimentalmente con el controlador LQR. Para lograr una comparación equitativa, se diseñó la función de recompensa del controlador RL de manera que contenga los mismos parámetros que la función objetivo minimizada por el controlador LQR. Los resultados experimentales mostraron que, si bien el controlador LQR presenta una amplitud de transitorio inicial mucho menor que la del controlador RL, en estado estacionario la desviación del péndulo respecto a la vertical es menor con el controlador RL. Asimismo,

tras el transitorio inicial, el valor RMS de la señal de control es ligeramente inferior para el controlador RL.

Sin embargo, ante perturbaciones, el controlador LQR exhibe un desempeño superior, ya que las oscilaciones transitorias se amortiguan más rápidamente que con el controlador RL. Como trabajo futuro, se desarrollará un análisis formal de estabilidad en lazo cerrado para el controlador RL propuesto, y se propondrá un esquema híbrido RL-LQR que combine las ventajas de ambos enfoques, compensando dinámicas no modeladas y perturbaciones, así como para mejorar la respuesta transitoria inicial del sistema.

Agradecimientos

Los autores agradecen a la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI) y al Programa para el Desarrollo Profesional Docente (PRODEP-SEP) de México por el apoyo para la realización de este trabajo. Igualmente, agradecen al editor y revisores por sus constructivos comentarios que enriquecieron este trabajo.

Referencias

Ataç, E., Yıldız, K., Ülkü, E. E., 2021. Use of PID control during education in reinforcement learning on two wheel balance robot. Gazi University Journal of Science Part C: Design and Technology 9 (4), 597–607.

Baek, J., Lee, C., Lee, Y. S., Jeon, S., Han, S., 2024. Reinforcement learning to achieve real-time control of triple inverted pendulum. Engineering Applications of Artificial Intelligence 128, 107518.

Block, D. J., Åström, K. J., Spong, M. W., 2007. The reaction wheel pendulum. Morgan & Claypool Publishers.

Chen, B.-R., Hsu, C.-F., Lee, T.-T., 2019. Stabilization of inertia wheel inverted pendulum using fuzzy-based hybrid control. In: 2019 International Conference on Machine Learning and Cybernetics (ICMLC). IEEE, pp. 1–6.

Chinelato, C. I. G., Neves, G. P. D., Angélico, B. A., 2020. Safe control of a reaction wheel pendulum using control barrier function. IEEE Access 8, 160315–160324.

Concha, A., Gadi, S. K., 2025a. Control de un péndulo con rueda inercial mediante aprendizaje por refuerzo.
URL: https://youtu.be/j1V13qQUsuE

Concha, A., Gadi, S. K., 2025b. Control de un péndulo con rueda inercial usando un regulador cuadrático lineal.
URL: https://youtu.be/Y8OqHMgy3Sc

Concha, A., Gadi, S. K., 2025c. Simulaciones y experimentos: control RL y LQR aplicado a un péndulo con rueda inercial.
URL: https://github.com/skgadi/Projects/tree/master/2024-pendulum-with-flywheel/Simulaciones_experimentos_articulo

Demircioğlu, U., Bakır, H., Bakır, R., June 2024. An investigation of pendulum control using reinforcement learning: Comparison of different agents. In: 3rd International Conference on Engineering, Natural and Social Sciences (ICENSOS 2024). pp. 94–101.

Espinosa, J. J., 2014. Control lineal de sistemas multivariables. Researchgate.

Franklin, G. F., Powell, J. D., Emami-Naeini, A., Powell, J. D., 2015. Feedback control of dynamic systems, 7th Edition. Pearson, Upper Saddle River, NJ.

Guo, W., Liu, D., 2019. Sliding mode observe and control for the underactuated inertia wheel pendulum system. IEEE Access 7, 86394–86402.

Hernandez, R., Garcia-Hernandez, R., Jurado, F., 2024. Modeling, simulation, and control of a rotary inverted pendulum: A reinforcement learning-based control approach. Modelling 5 (4), 1824–1852.

Hfaiedh, A., Chemori, A., Abdelkrim, A., 2020. Stabilization of the inertia wheel inverted pendulum by advanced IDA-PBC based controllers: Comparative study and real-time experiments. In: 2020 17th International Multi-Conference on Systems, Signals & Devices (SSD). IEEE, pp. 753–760.

Hidayati, A. N., Wasiwitono, U., 2021. Modeling and control of inertia wheel pendulum system with LQR and PID control. In: 2021 International Seminar on Intelligent Technology and Its Applications (ISITIA). IEEE, pp. 135–140.

Ho, T.-N., et al., 2025. Model-free swing-up and balance control of a rotary inverted pendulum using the TD3 algorithm: Simulation and experiments. Engineering, Technology & Applied Science Research 15 (1), 19316–19323.

Huanlong, L., Zhengjie, W., Bin, J., Hongyu, P., 2021. An inertia wheel pendulum control method based on actor-critic learning algorithm. In: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, pp. 1281–1285.

Israilov, S., Fu, L., Sánchez-Rodríguez, J., Fusco, F., Allibert, G., Raufaste, C., Argentina, M., 2023. Reinforcement learning approach to control an inverted pendulum: A general framework for educational purposes. Plos one 18 (2), e0280071.

Johannink, T., Bahl, S., Nair, A., Luo, J., Kumar, A., Loskyll, M., Ojea, J. A., Solowjow, E., Levine, S., 2019. Residual reinforcement learning for robot control. In: 2019 international conference on robotics and automation (ICRA). IEEE, pp. 6023–6029.

Kanjanawanishkul, K., 03 2015. LQR and MPC controller design and comparison for a stationary self-balancing bicycle robot with a reaction wheel. Kybernetika 54, 173–191.
DOI: 10.14736/kyb-2015-1-0173

Kuo, B. C., 1996. Automatic control systems. Prentice Hall PTR.

Lee, T., Ju, D., Lee, Y. S., 2025. Transition control of a double-inverted pendulum system using Sim2Real reinforcement learning. Machines 13 (3), 186.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2015. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

Matlab, 2024. Deep deterministic policy gradient (DDPG) agents.
URL: https://la.mathworks.com/help/reinforcement-learning/ug/ddpg-agents.html

Montoya, O. D., Gil-González, W., 2020. Nonlinear analysis and control of a reaction wheel pendulum: Lyapunov-based approach. Engineering Science and Technology, an International Journal 23 (1), 21–29.

Ogata, K., 2010. Modern control engineering, 5th Edition. Prentice Hall.

Oliveira, A. I. S., Leite, A. C., Caarls, W., 2022. Intelligent robust control for second-order non-linear systems with smart gain tuning based on reinforcement learning. In: 2022 International Joint Conference on Neural Networks (IJCNN). pp. 1–8.
DOI: 10.1109/IJCNN55064.2022.9892099

Özalp, R., Varol, N. K., Taşci, B., Uçar, A., 2020. A review of deep reinforcement learning algorithms and comparative results on inverted pendulum system. Machine Learning Paradigms: Advances in Deep Learning-based Technological Applications, 237–256.

Pololu Corporation, 2020. 37D Metal Gearmotors.
URL: https://www.pololu.com/file/0J1706/pololu-37d-metal-gearmotors.pdf

Sandoval, J., Kelly, R., Santibáñez, V., 2022. Sobre el control por moldeo de energía más inyección de amortiguamiento de sistemas mecánicos. Revista Iberoamericana de Automática e Informática industrial 19 (4), 407–418.

Sutton, R. S., Barto, A. G., 2018. Reinforcement learning: An introduction. MIT press.

Teja, G. P., Dhabale, A., Waghmare, T., 2020. Nonlinear control of the reaction wheel pendulum using passivity-based control and backstepping control. In: 2020 IEEE First International Conference on Smart Technologies for Power, Energy and Control (STPEC). IEEE, pp. 1–6.

Vadlamudi, S., Lakshmi, K. V., Yaramala, A., Uppalapati, C., Kolluri, P. K., 2024. Self balancing motorcycle using reinforcement learning. In: 2024 International Conference on Emerging Systems and Intelligent Computing (ESIC). IEEE, pp. 691–696.

Zabihifar, S. H., Navvabi, H., Yushchenko, A. S., 2021. Dual adaptive neural network controller for underactuated systems. Robotica 39 (7), 1281–1298.

Zaborniak, D., Patan, K., Witczak, M., 2024. Design, implementation, and control of a wheel-based inverted pendulum. Electronics 13 (3), 514.

Zai, A., Brown, B., 2020. Deep reinforcement learning in action. Manning Publications.

Zhu, X., Deng, Y., Zheng, X., Zheng, Q., Chen, Z., Liang, B., Liu, Y., 2023. Online series-parallel reinforcement-learning-based balancing control for reaction wheel bicycle robots on a curved pavement. IEEE Access 11, 66756–66766.