Experimentos de control en un robot didáctico móvil de dos ruedas con autobalance para estudiantes de licenciatura y posgrado

Arturo Antonio Muñoz-Flores[1], Ricardo Alvarez-Salas[1], Antonio Cárdenas[1], Mario González-García[2], Luis Alberto González-Murillo[1], Héctor Méndez-Azúa[1]
[1]Facultad de Ingeniería, Universidad Autónoma de San Luis Potosí, San Luis Potosí, 78290 México
[2]SECIHTI - Facultad de Ingeniería, Universidad Autónoma de San Luis Potosí, San Luis Potosí, 78290 México
ralvarez@uaslp.mx

Resumen

El presente artículo describe el modelado y el control de un robot de dos ruedas con autobalance. Se utiliza el enfoque de Euler-Lagrange para obtener el modelo del robot. Una vez definidas las ecuaciones dinámicas, se linealiza el sistema en el punto de equilibrio inestable. Posteriormente, se utiliza el método de Ackermann para diseñar un controlador de retroalimentación de estados. Para comprobar el correcto funcionamiento del controlador se realizan simulaciones numéricas. Finalmente, se programa el algoritmo de control en el microcontrolador ATmega328 incorporado al robot. Este proyecto tiene como objetivo servir de plataforma didáctica para la enseñanza de la teoría de control basada en modelo a estudiantes de licenciatura y de posgrado en ramas afines a la robótica.

Palabras clave: Robot móvil; autobalance; modelado; ecuaciones de Euler-Lagrange, control.

Introducción

El uso de los robots móviles ha ido incrementando con el paso de los años debido a su versatilidad para realizar diversos tipos de tareas, por ejemplo, transporte de materiales, navegación en diversos tipos de espacios y uso lúdicos y sociales (Chan *et al*., 2013). Un robot móvil puede tener ruedas o piernas como mecanismos de locomoción. Dentro de los robots con ruedas se incluyen los robots de dos ruedas con autobalance.

Un robot que tiene tres o más ruedas puede mantener su equilibrio estático, en contraste, un robot de dos ruedas con autobalance requiere de un algoritmo de control para mantener su equilibrio en todo momento. Desde el punto de vista del modelado y control, esta característica conduce a un sistema inestable, subactuado y no lineal. La subactuación implica que el número de entradas de control disponibles es menor que el número de grados de libertad del sistema, por lo que las estrategias de control deben considerar de manera conjunta el equilibrio y el movimiento del robot.

Los robots de dos ruedas con autobalance presentan una configuración diferencial que aparte de mantener el equilibrio les permite controlar en ángulo de la rotación. Asimismo, el mantener constantemente su equilibrio permite responder a perturbaciones y conservar una configuración cercana a la posición vertical.

Estas características permiten considerar el uso de este tipo de robots para aplicaciones en espacios con restricciones de movilidad, como pasillos y zonas de circulación reducida. Además, la utilización de dos ruedas permite emplear ruedas de mayor diámetro, dependiendo del diseño de la plataforma, lo que puede modificar las capacidades de desplazamiento sobre superficies con irregularidades.

En la revisión presentada en (Zhang y Mohamad Nor, 2025), el objetivo de control predominante en los robots con autobalance de dos ruedas es la estabilización del ángulo de inclinación, problema fundamental asociado a la dinámica tipo péndulo invertido y a la naturaleza subactuada del sistema. La mayoría de las estrategias analizadas, incluyendo controladores lineales, no lineales, adaptativos e inteligentes se enfocan principalmente en garantizar estabilidad.

En (Hou *et al*., 2023) proponen un controlador jerárquico por modos deslizantes combinado con un observador no lineal de perturbaciones (NOD) para el seguimiento de trayectorias en robots de dos ruedas con autobalance. No obstante, el enfoque se valida únicamente mediante simulaciones.

Los autores del trabajo (Garces *et al*., 2020) presentan el diseño e implementación de un robot con autobalance híbrido capaz de realizar seguimiento de objetos y evasión de obstáculos mediante visión artificial y sensores ultrasónicos.

El trabajo (Tsai *et al*., 2008) presenta una arquitectura de navegación planeada para un robot con autobalance de dos ruedas basada en un modelo matemático linealizado. A partir de este modelo se diseñan controladores adaptativos.

Los autores del artículo (Hsieh y Li, 2024), proponen un procedimiento automatizado para la entrada a edificios mediante rampas accesibles en robots móviles con autobalance. El método integra detección de objetos basada en YOLOv4. Sin embargo, el enfoque es fundamentalmente reactivo y no incorpora un modelo dinámico explícito del robot ni análisis formal de estabilidad.

En el trabajo realizado en (Lin *et al*., 2022) se propone un robot móvil autónomo estructurado como un Sistema Ciber-Físico (CPS). La estrategia de navegación combina un direccionamiento visual *end-to-end* basado en ResNet18 con una localización topológica mediante clasificación de imágenes basada en ConvNet.

El trabajo (Lin y Li, 2025) los autores propusieron un marco de navegación RGB-D de *end to end* para un robot con autobalance de dos ruedas, integrando una arquitectura ConvNet de cuatro entradas para unificar múltiples tareas de navegación en pasillos dentro de un único modelo.

En (Li *et al*., 2021) se propone un robot móvil con autobalance de dos ruedas que incorpora un esquema de navegación visual inteligente basado en aprendizaje profundo de tipo *end to end*. El enfoque emplea una arquitectura de red neuronal convolucional.

A diferencia de trabajos que abordan únicamente la estabilización de robots con autobalance, en (Le *et al*., 2024) se propone una estrategia integral que combina estabilización dinámica y seguimiento de trayectoria planificada. Se usa controladores basados en *Sliding Mode Control* (SMC) y *Hierarchical Sliding Mode Control* (HSMC). Aunque los resultados obtenidos se limitan a simulaciones.

Los autores del trabajo realizado en (Liu *et al*., 2017), propusieron una mejora del algoritmo RBPF-SLAM para entornos interiores estructurados, incorporando segmentación geométrica y emparejamiento de líneas extraídas de datos LIDAR con el fin de optimizar la función de importancia de un filtro de partículas. El método fue validado experimentalmente.

En el trabajo (Kim *et al*., 2025), se propone un método de preprocesamiento denominado *Pitch Variation Filter* para mitigar el efecto de variaciones abruptas de inclinación en robots con autobalance de dos ruedas que emplean exclusivamente LiDAR 2D para localización y mapeo. Pero en el artículo no se aborda el modelo dinámico y el controlador de equilibrio utilizado.

En (Khabazi y Shariyat, 2024) se presenta el modelado y simulación de la navegación autónoma de un robot con autobalance de dos ruedas. Se deriva un modelo matemático del sistema para el diseño de un controlador PID orientado a estabilizar el ángulo de inclinación, la orientación y la velocidad lineal.

Los autores del trabajo (Chen *et al*., 2025) derivan un modelo dinámico utilizando el enfoque de Euler–Lagrange y se linealizó alrededor del punto de equilibrio. Para el control de equilibrio, se implementó una estrategia PID adaptativa doble difusa con anti-saturación integral (ADFAS-PID), con ajuste de parámetros en línea.

En el trabajo (Liu *et al*., 2024) se aplica un control no lineal predictivo basado en modelo a una silla de ruedas con autobalance. El objetivo de usar un controlador no lineal es controlar un rango mayor de grados de inclinación de la silla. Las pruebas se realizan en simulación y muestran que la silla es capaz de levantarse desde una posición casi horizontal. Sin embargo, el par necesario para lograr levantar la silla es de casi 200 Nm, lo cual es muy grande para aplicarse en un prototipo real.

Los autores de (Zhao *et al*., 2023) diseñan un robot con autobalance para inspeccionar y monitorear los equipos de salas de bombeo en minas de carbón. De esta manera, este trabajo muestra una función de este tipo de robots enfocada a aplicaciones industriales. Para controlar el equilibrio se tiene un controlador PID. Se agrega SLAM mediante el algoritmo de Gmapping en un LiDAR 2D. Sin embargo, el algoritmo de SLAM no contempla la dinámica de equilibrio. Las pruebas se presentan en Gazebo de ROS.

El artículo (Papadimitriou *et al*., 2025), presenta el desarrollo de un controlador de conmutación basado en un algoritmo genético para desarrollar parámetros de control más eficaces y aumentar la autonomía en un robot con autobalance. Los resultados se presentan en el simulador CoppeliaSim, mostrando que el robot es capaz de mantener el equilibrio al recorrer superficies inclinadas e irregulares.

En (Bekhiti *et al*., 2025), se diseña un controlador para un robot con autobalance, sin embargo, se mezcla técnicas basadas en modelo con las de aprendizaje adaptativo. Este artículo tiene como objetivo analizar la capacidad de seguimiento del controlador, al simular que el robot se estabilicé en un ángulo distinto al necesario para mantener el equilibrio. Las pruebas son únicamente en simulación.

Los autores de (Zhang *et al*., 2025), extienden el controlador HSMC incorporando la dinámica de deslizamiento terminal en una estructura jerárquica de doble capa. A este controlador se le llama *Modified Dual Hierarchical Terminal Sliding Mode Control*. Los resultados se limitan a simulaciones en Matlab/Simulink.

En (Gandarilla *et al*., 2022), se controla un robot con autobalance con la combinación de redes neuronales y linealización entrada-salida. La ventaja de usar redes neuronales es que complementan el controlador haciéndolo adaptable. Las pruebas se realizaron en un robot real.

Los autores de (Wang *et al*., 2021), realizaron la construcción de un robot con autobalance para asistir en la acción de caminar para personas. El controlador es del tipo robusto de multi nivel. Se realizan pruebas experimentales donde el robot mantiene el equilibrio, permitiendo ayudar a una persona con discapacidad a caminar.

En (Yang *et al*., 2024) se presenta el control de un robot con autobalance mediante la técnica de *Robust Integral of Sign Error,* RISE. Se usa la *Modified Prescribed Performance Function*, MPPF, para controlar los errores de seguimiento. Para atenuar las perturbaciones externas se hace uso de la *Radial Basis Function Neural Network*, RBFNN. Las pruebas realizadas fueron únicamente en simulación.

En (Pham *et al*., 2026) se hace uso de una combinación de HSMC, control óptimo y un *High Order Disturbance Observer,* HODO, para controlar un robot con autobalance. El objetivo es reducir el consumo de energía de la batería, robustez y reducir el *Chattering.* Asimismo, se usa la técnica de *Adaptive Dynamic Programming*, ADP, para calcular una ley óptima de aproximación a la superficie usada en el HSMC.

Los autores del trabajo (Nguyen *et al*., 2025) presentan un controlador para un robot con autobalance que permite mantener el equilibrio y seguir una trayectoria. La generación de trayectorias se realiza mediante un algoritmo A* mejorado. También se usa *Knowledge Distillation* (KD) para la segmentación de imágenes de una cámara monocular montada en el robot. Con las imágenes captadas se procede a usar la técnica de *Birds Eyes View* (BEV) para generar un mapa. Se realizan pruebas experimentales donde el robot sigue una trayectoria marcada.

El trabajo presentado en (Mai *et al*., 2023) se usa un controlador difuso y óptimo para el control de un robot con autobalance. A su vez, los autores usan el análisis de estabilidad de Lyapunov para diseñar las reglas difusas del controlador. Es necesario recalcar que el proceso de optimización se realiza *off-line*. Se realizan pruebas experimentales donde el robot mantiene el equilibrio mientras recorre una distancia determinada.

En (Moritz *et al*., 2024) se diseña y modelo un robot con autobalance. Usan la cinemática inversa para obtener las velocidades de referencia para que el robot pueda seguir una trayectoria. El modelo dinámico del robot incorpora los ángulos de inclinación y rotación para posteriormente ser linealizado. Se incorpora un generador de trayectorias *Artifitial Potential Field* (APF). Se usa tres controladores PID para controlar el equilibrio y seguimiento de trayectoria. Se muestran los resultados en simulación donde el robot es capaz de seguir una trayectoria circular.

En (Ahmadi Jeyed y Ghaffari, 2022) se realiza un control optimo no lineal basado en parametrización para un robot con autobalance. Se utiliza un modelo que considera tanto el ángulo de inclinación como el de rotación. El controlador por utilizar es el *State Dependt Riccati Equation* (SDRE) que conserva la naturaleza no lineal del sistema. De esta manera son consigue que las ganancias del controlador cambien conforme cambia el estado del robot. El robot mantiene el equilibrio mientras sigue las referencias de los estados.

Deducción del modelado dinámico

Para el modelado dinámico del robot de autobalance se hace uso del enfoque de Euler-Lagrange, siguiendo la referencia de (Mahler y Haase, 2013). Para validar que el modelo sea correcto se analizan los resultados obtenidos gráficamente buscando que dicho comportamiento sea congruente con la física del sistema. Para comenzar con el modelado se usa el diagrama mostrado en la Figura 1.

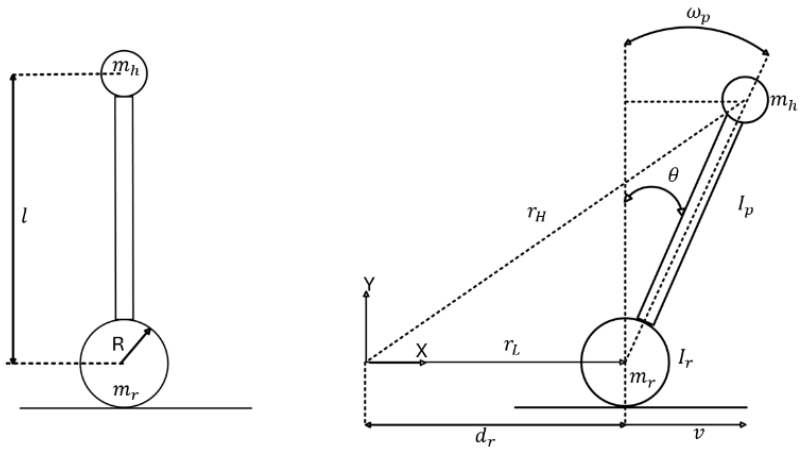


Figura 1. Diagrama del robot de dos ruedas con autobalance.
Fuente: Autoría propia.

donde

• R es el radio de las ruedas.

• m_r es la masa concentrada de las ruedas.

• m_h es la masa concentrada del cuerpo del robot.

• l es la distancia del eje de las ruedas al centro de masa del robot m_h.

• θ es el ángulo de inclinación del robot respecto al eje vertical, siendo el sentido de las manecillas del reloj el sentido positivo.

• d_r es la distancia del centro de las ruedas respecto al marco de referencia.

• v es la velocidad de las ruedas del robot.

• r_h es la distancia del centro de masa del robot al marco de referencia.

• r_l es la distancia del centro de las ruedas al marco de referencia.

• I_r es el momento de inercia del conjunto de ruedas del robot respecto al eje de rotación de estas.

• I_p es el momento de inercia del robot respecto al eje de rotación de las ruedas.

• ω_r es la velocidad angular de las ruedas.

• ω_p es la velocidad angular del cuerpo del robot.

Para utilizar el enfoque de Euler-Lagrange es necesario calcular las energías cinética y potencial del robot. La energía cinética de las ruedas se muestra en la ecuación (1). Debido a que el centro de masa de las ruedas se encuentra a la misma altura del marco de referencia la energía potencia de las ruedas será cero.

$$K_r = \frac{1}{2} m_r v^2 + \frac{1}{2} I_r \omega_r^2 \tag{1}$$

Para definir las energías cinéticas y potenciales del péndulo invertido es necesario definir el vector r_H y su respectiva derivada. Esto se muestra en las ecuaciones (2) y (3).

$$r_H = [(d_r + l\, sen\theta) \quad (lcos\theta) \quad 0] \tag{2}$$

$$\dot{r}_H = \left[(\dot{d}_r + l\dot{\theta} cos\theta) \quad \left(-l\dot{\theta} sen\theta\right) \quad 0\right] = \left[(\dot{d}_r + l\omega_p cos\theta) \quad \left(-l\omega_p sen\theta\right) \quad 0\right] \tag{3}$$

Con estas ecuaciones se puede definir la energía cinética del péndulo invertido, mostrada en la ecuación (4).

$$K_p = \frac{1}{2} m_h \dot{r}_h^T \dot{r}_H + \frac{1}{2} I_p \omega_p^2 = \frac{1}{2} m_h \left(v^2 + 2l\omega_p v\, cos\theta + l^2 \omega_p^2\right) + \frac{1}{2} I_p \omega_p^2 \tag{4}$$

Además, se calcula la energía potencial del péndulo invertido, representada en (5).

$$V_p = m_h [0 \quad g \quad 0] \begin{bmatrix} (\dot{d}_r + l\dot{\theta} cos\theta) \\ \left(-l\dot{\theta} sen\theta\right) \\ 0 \end{bmatrix} = m_h g l cos\theta \tag{5}$$

Además de las energías previamente definidas, es necesario representar la energía disipada por la fricción. Para cumplir con este objetivo se establece una función de disipación de Rayleigh (6).

$$\mathcal{R}_p = \frac{1}{2}\left(c_v v^2 + c_\omega \left(\frac{v}{R} - \omega_p\right)^2\right) \tag{6}$$

Con todos los elementos anteriores se puede calcular el Lagrangiano (7) y las ecuaciones diferenciales del sistema siguiendo el enfoque de Euler-Lagrange denotado por las ecuaciones (8) y (9).

$$L = (K_p + K_r) - (V_p + V_r) \tag{7}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial v}\right) - \frac{\delta L}{\delta d_r} + \frac{\delta \mathcal{R}_p}{\delta v} = \tau_1 \tag{8}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \omega_p}\right) - \frac{\delta L}{\delta \theta} + \frac{\delta \mathcal{R}_p}{\delta \omega_p} = \tau_2 \tag{9}$$

Finalmente, se obtienen las dos ecuaciones diferenciales que rigen el comportamiento del sistema (10) y (11). La entrada $\tau_1 = \frac{\tau}{R}$ corresponde a la fuerza que hace mover al robot adelante y atrás, y τ es el par generado por los motores del robot. La entrada τ_2 es cero ya que corresponde a un par que controla directamente e ángulo de inclinación. Como no hay un motor acoplado al péndulo invertido, esta entrada es cero.

$$\left(m_h + m_r + \frac{I_r}{R^2}\right)\dot{v} + l m_h \left(\dot{\omega}_p cos\theta - \omega_p^2 sen\theta\right) + c_v v + \frac{c_\omega v}{R^2} - \frac{c_\omega \omega_p}{R} = \tau_1 = \frac{\tau}{R} \tag{10}$$

$$\left(I_p + l^2 m_h\right)\dot{\omega}_p + l m_h (\dot{v} cos\theta - g sen\theta) + c_\omega \omega_p - \frac{c_\omega v}{R} = \tau_2 = 0 \tag{11}$$

Al resolver el sistema de ecuaciones diferenciales simultaneas (10) y (11) se puede obtener la representación en espacio de estados. Posteriormente se puede linealizar en el punto de equilibrio inestable del sistema. Los estados se muestran a continuación.

$$x = \begin{bmatrix} x_1 = d_r \\ x_2 = \dot{x}_1 = v \\ x_3 = \theta \\ x_4 = \dot{x}_3 = \omega_p \end{bmatrix} \tag{12}$$

El modelo resultante tiene la forma que se muestra en las ecuaciones (13) a (16).

$$f_1 = \dot{x}_1 = x_2 \tag{13}$$

$$f_2 = \dot{x}_2 = f_2(x_1, x_2, x_3, x_4) \tag{14}$$

$$f_3 = \dot{x}_3 = x_4 \tag{15}$$

$$f_4 = \dot{x}_4 = f_4(x_1, x_2, x_3, x_4) \tag{16}$$

Para linealizar el sistema se usa las matrices (17) y (18). Posteriormente se evalúan las matrices en el punto de equilibrio seleccionado, así como con los parámetros mecánicos del robot.

$$A = \left[\frac{\partial f(x,u)}{\partial x}\right]_{\overline{x},\overline{u}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \end{bmatrix}_{x=[0,0,0,0],\tau=0} \tag{17}$$

$$B == \left[\frac{\partial f(x,y)}{\partial u}\right]_{\overline{x},\overline{u}} = \begin{bmatrix} 0 \\ \frac{\partial f_2}{\partial \tau} \\ 0 \\ \frac{\partial f_4}{\partial \tau} \end{bmatrix}_{x=[0,0,0,0],\tau=0} \tag{18}$$

Debido a que la unidad donde se programará el algoritmo de control es un microcontrolador ATmega328, es necesario discretizar el sistema linealizado. Para esto se usa un período de muestreo de 15 ms. El sistema discretizado se presenta en las ecuaciones (19) y (20).

$$A_{disc} = \begin{bmatrix} 1 & 0.01403 & -0.0004097 & 1.716 \times 10^{-5} \\ 0 & 0.8737 & -0.05284 & 0.002088 \\ 0 & 0.009983 & 1.012 & 0.01483 \\ 0 & 1.297 & 1.554 & 0.9817 \end{bmatrix} \tag{19}$$

$$B_{disc} = \begin{bmatrix} 0.0059 \\ 0.767 \\ -0.04546 \\ -5.863 \end{bmatrix} \tag{20}$$

Diseño del controlador

Para diseñar el controlador por retroalimentación de estados, se utiliza el método de Ackermann. Para ello, se comienza por obtener la matriz de controlabilidad mostrada en la ecuación (21). Dicha matriz es de rango pleno, por lo tanto, el sistema es controlable.

$$C_d = \begin{bmatrix} 0.0059 & 0.0166 & 0.0258 & 0.0339 \\ 0.7670 & 0.6603 & 0.5734 & 0.5026 \\ -0.0455 & -0.1253 & -0.1918 & -0.2489 \\ -5.8632 & -4.8319 & -4.0818 & -3.5615 \end{bmatrix} \tag{21}$$

Posteriormente, se obtiene el polinomio característico del sistema A_{disc}, ecuación (22).

$$z^4 - 3.8672z^3 + 5.5771z^2 - 3.5544z + 0.8445 = z^4 + \alpha_1 z^3 + \alpha_2 z^2 + \alpha_3 z + \alpha_4 = 0 \tag{22}$$

Lo siguiente es obtener una serie de matrices auxiliares para este proceso, las cuales se muestran en las ecuaciones (23) a (25).

$$\bar{C}_d^{-1} = \begin{bmatrix} 1 & \alpha_1 & \alpha_2 & \alpha_3 \\ 0 & 1 & \alpha_1 & \alpha_2 \\ 0 & 0 & 1 & \alpha_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -3.8672 & 5.5771 & -3.5544 \\ 0 & 1 & -3.8672 & 5.5771 \\ 0 & 0 & 1 & -3.8672 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{23}$$

$$P_d^{-1} = (C_d)(\bar{C}_d^{-1}) = \begin{bmatrix} 0.0059 & -0.0062 & -0.0054 & 0.0055 \\ 0.7670 & -2.3059 & 2.2975 & -0.7587 \\ -0.0455 & 0.0505 & 0.0391 & -0.0442 \\ -5.8632 & 17.8426 & -18.0955 & 6.1162 \end{bmatrix} \tag{24}$$

$$P_d = (1 \times 10^3) \begin{bmatrix} -5.0495 & -0.4022 & -0.8374 & -0.0514 \\ -5.0495 & -0.3114 & -0.7850 & -0.0397 \\ -5.0495 & -0.2243 & -0.7456 & -0.0286 \\ -5.0495 & -0.1408 & -0.7188 & -0.0179 \end{bmatrix} \tag{25}$$

De acuerdo con los valores propios deseados para el sistema, se construye el polinomio característico deseado, ecuación (26). Este polinomio garantiza que el sistema presenta un comportamiento similar al de un sistema de segundo grado.

$$z^4 - 3.3411z^3 + 4.1661z^2 - 2.2970z + 0.4724 = z^4 + \bar{\alpha}_1 z^3 + \bar{\alpha}_2 z^2 + \bar{\alpha}_3 z + \bar{\alpha}_4 = 0 \tag{26}$$

Finalmente, el vector de retroalimentación se obtiene con las ecuaciones (27) y (28).

$$\bar{K} = \begin{bmatrix} \bar{\alpha}_1 - \alpha_1 \\ \bar{\alpha}_2 - \alpha_2 \\ \bar{\alpha}_3 - \alpha_3 \\ \bar{\alpha}_4 - \alpha_4 \end{bmatrix}^T = \begin{bmatrix} 0.5262 \\ -1.4110 \\ 1.2573 \\ -0.3722 \end{bmatrix}^T \tag{27}$$

$$K = (\bar{K})(P_d) = [-1.7275 \quad -1.8906 \quad -3.0031 \quad -0.3155] \tag{28}$$

Resultados

Para poder comprobar que el modelo y el diseño del controlador sean correctos, se procede a realizar una simulación en lazo abierto, Figura 2 y en lazo cerrado Figura 3.

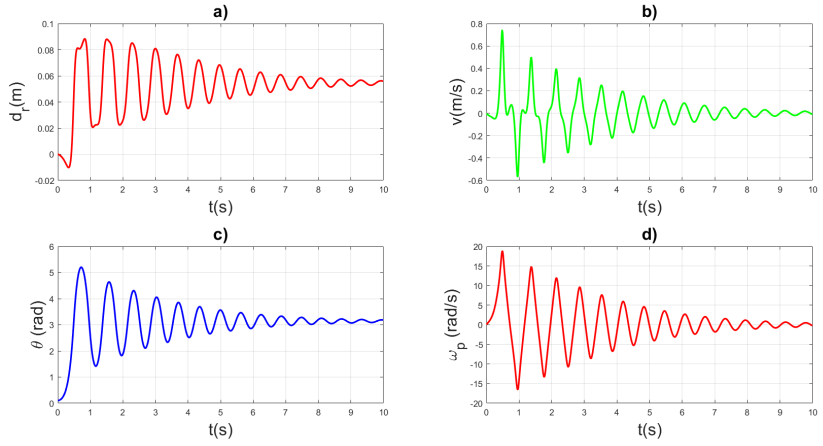


Figura 2. Respuestas del sistema en lazo abierto en tiempo discreto: a) Distancia recorrida. b) Velocidad lineal. c) Ángulo de inclinación. d) Velocidad angular.
Fuente: Autoría propia.

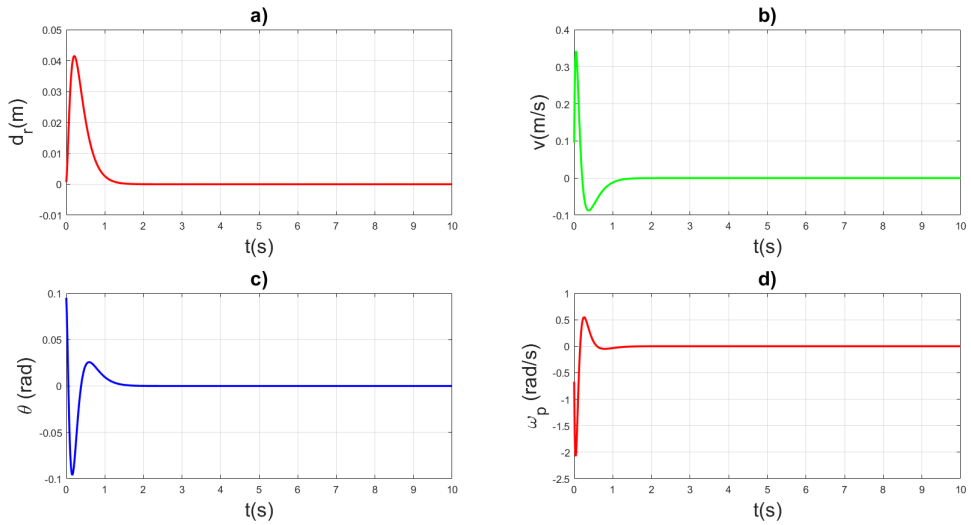


Figura 3. Respuestas del sistema en lazo cerrado en tiempo discreto: a) Distancia recorrida. b) Velocidad lineal. c) Ángulo de inclinación. d) Velocidad angular.
Fuente: Autoría propia.

Al implementar el algoritmo de control en el microcontrolador del robot, este logró mantener el equilibrio satisfactoriamente. En la Figura 4 se muestra al robot en equilibrio.

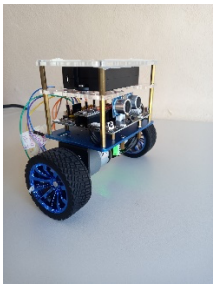


Figura 4. Robot manteniendo el equilibrio.
Fuente: Autoría propia.

En la Figura 5 se muestra una gráfica del ángulo de inclinación obtenida directamente del sensor IMU del robot. Se puede observar que el ángulo de inclinación se mantiene en un rango entre -1^o y 2^o.

Para determinar el desempeño del controlador en lazo cerrado se hace uso de la Integral del Error Cuadrático (IEC), la Integral del Valor Absoluto de la Señal de Control (IVABSC) y la Integral del Valor Absoluto de la Variación de la Señal de Control (IVAVSC). Estos parámetros se muestran en las ecuaciones (29) a (31). Para esta tarea se usa de referencia (Maldonado Ramos *et al*., 2020) y (Dorf y Bishop, 2017). Se debe recalcar que todo este procedimiento se realiza con datos experimentales obtenidos mediante la unidad inercial del sensor y el monitor serial de la interfaz de la plataforma desarrollada. Y la comparación se realiza entre el controlador diseñado y un conjunto de PI-PD que venía implementado de fabrica en el robot.

$$IEC = \sum_{k=0}^{N} e[k]^2 \tag{29}$$

$$IVABSC = \sum_{k=0}^{N} |u[k]| \tag{30}$$

$$IVAVSC = \sum_{k=0}^{N} \left[\frac{u[k]-u[k-1]}{\Delta t}\right] \tag{31}$$

Los parámetros de desempeño para ambos controladores se muestran en la Tabla 1. Se puede apreciar que en general los parámetros del IVAVSC en el caso del controlador por retroalimentación de estados son prácticamente la mitad de los valores del controlador PI-PD. Esto indica que la variación de la señal de control tiene menos variaciones que la señal del controlador PI-PD. Aunque el PI-PD tiene un error menor, de acuerdo con los valores IEC.

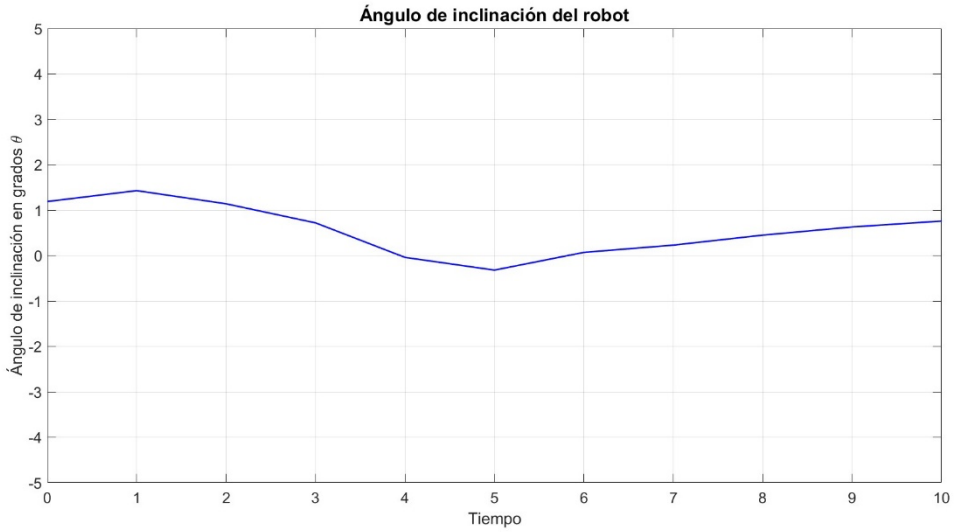


Figura 5. Ángulo de inclinación del robot medido por la unidad IMU.
Fuente: Autoría propia.

***Tabla 1**. Parámetros de desempeño de los controladores analizados.*

Parámetro de desempeño	Controlador PI-PD	Controlador por retroalimentación de estados
IEC	11.5540	55.2093
IVABSC (PWM del motor derecho)	1340.9900	1285.5700
IVABSC (PWM del motor izquierdo)	1169.2700	1285.5700
IVAVSC (PWM del motor derecho)	8864.6000	3182.2000
IVAVSC (PWM del motor izquierdo)	7636.0000	3182.2000

Conclusiones

De acuerdo con los resultados obtenidos, se diseñó un controlador basado en el modelo para que un robot de dos ruedas con autobalance mantenga el equilibrio. El desarrollo del modelo del robot permite comprender su dinámica inestable, observada en las simulaciones. Además, simular el comportamiento del robot con el controlador diseñado permite verificar que el diseño ha sido realizado correctamente. Finalmente, al utilizar parámetros de desempeño, se puede analizar cuantitativamente el rendimiento de los controladores.

El presente trabajo sienta las bases para disponer de una plataforma de bajo costo para la enseñanza de algoritmos de control basados en modelo para estudiantes de licenciatura y posgrado en las diversas ramas de la ingeniería relacionadas con la robótica.

Referencias

Ahmadi Jeyed, H., y Ghaffari, A. (2022). A nonlinear optimal control based on the SDRE technique for the two-wheeled self-balancing robot. *Australian Journal of Mechanical Engineering,* 20(3), 722–730. https://doi.org/10.1080/14484846.2020.1745733

Bekhiti, B., Al-Sabur, R., Roudane, M., Younis, J. A., y Sharkawy, A.-N. (2025). Intelligent neuro-fuzzy adaptive MIMO control for a self-balancing two wheeled autonomous robot via recursive resolution of the matrix Diophantine equation. *Discover Robotics,* 1(1), 11. https://doi.org/10.1007/s44430-025-00011-3

Chan, R. P. M., Stol, K. A., y Halkyard, C. R. (2013). Review of modelling and control of two-wheeled robots. *Annual Reviews in Control,* 37(1), 89–103. https://doi.org/10.1016/j.arcontrol.2013.03.004

Chen, J., He, N., Xu, Z., Dou, M., y He, L. (2025). Design and implementation of a novel two-wheeled composite self-balancing robot for stationary operations in unknown terrain. *IEEE Access,* 13, 86032–86045. https://doi.org/10.1109/ACCESS.2025.3569325

Dorf, R. C., y Bishop, R. H. (2017). *Modern control systems* (13th ed.). Pearson.

Gandarilla, I., Montoya-Cháirez, J., Santibáñez, V., Aguilar-Avelar, C., y Moreno-Valenzuela, J. (2022). Trajectory tracking control of a self-balancing robot via adaptive neural networks. *Engineering Science and Technology, an International Journal,* 35, 101259. https://doi.org/10.1016/j.jestch.2022.101259

Garces, S. R., Beb, M. Y., Boubakari, A., Ammar, H. H., y Shalaby, M. A. W. (2020). Hybrid self-balancing and object tracking robot using artificial intelligence and machine vision. In 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES) (pp. 264–269). IEEE. https://doi.org/10.1109/NILES50944.2020.9257916

Hou, M., Zhang, X., Chen, D., y Xu, Z. (2023). Hierarchical sliding mode control combined with nonlinear disturbance observer for wheeled inverted pendulum robot trajectory tracking. *Applied Sciences,* 13(7), 4350. https://doi.org/10.3390/app13074350

Hsieh, H.-C., y Li, C.-H. G. (2024). Autonomous building entry integrating ConvNet object detection and cascaded navigation control. In 2024 IEEE International Conference on Advanced Intelligent Mechatronics (AIM) (pp. 603–608). IEEE. https://doi.org/10.1109/AIM55361.2024.10637179

Khabazi, H., y Shariyat, M. (2024). Control and simulation of autonomous navigation of a self-balancing robot. (pp. 226–232). https://doi.org/10.1109/ICRoM64545.2024.10903531

Kim, D., Lee, H., y Choi, K. H. (2025). Pitch variation filter for LiDAR-only SLAM and localization in self-balancing mobile robot. *IEEE Robotics and Automation Letters*, 10(12), 12836–12843. https://doi.org/10.1109/LRA.2025.3626231

Le, H. N., Pham, M.-K., Pham, D.-H., y Nguyen, T.-V.-A. (2024). Trajectory tracking and stabilization of two-wheeled balancing mobile robot with hierarchical and sliding mode control. *International Journal of Dynamics and Control*, 13(1), 10. https://doi.org/10.1007/s40435-024-01518-0

Li, C.-H. G., Zhou, L.-P., y Chao, Y.-H. (2021). Self-balancing two-wheeled robot featuring intelligent end-to-end deep visual-steering. *IEEE/ASME Transactions on Mechatronics,* 26(5), 2263–2273. https://doi.org/10.1109/TMECH.2020.3036579

Lin, H.-Z., Chen, H.-H., Choophutthakan, K., y Li, C.-H. G. (2022). Autonomous mobile robot as a cyber-physical system featuring networked deep learning and control. In *2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (AIM) (pp. 268–274). IEEE. https://doi.org/10.1109/AIM52237.2022.9863365

Lin, Z.-Y., y Li, C.-H. G. (2025). Enhanced hallway navigation featuring end-to-end RGB-D quadruple-ConvNet steering. In 2025 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) (pp. 1–6). IEEE. https://doi.org/10.1109/AIM64088.2025.11175760

Liu, C., Qin, K., Xin, G., Li, C., y Wang, S. (2024). Nonlinear model predictive control for a self-balancing wheelchair. *IEEE Access,* 12, 28938–28949. https://doi.org/10.1109/ACCESS.2024.3368853

Liu, F., Chen, Y., y Li, Y. (2017). Research on indoor robot SLAM of RBPF improved with geometrical characteristic localization. In 2017 29th Chinese Control and Decision Conference (CCDC) (pp. 3325–3330). IEEE. https://doi.org/10.1109/CCDC.2017.7979080

Mahler, B., y Haase, J. (2013). Mathematical model and control strategy of a two-wheeled self-balancing robot. In IECON 2013—39th Annual Conference of the IEEE Industrial Electronics Society (pp. 4198–4203). IEEE. https://doi.org/10.1109/IECON.2013.6699809

Mai, T. A., Dang, T. S., Ta, H. C., y Ho, S. P. (2023). Comprehensive optimal fuzzy control for a two-wheeled balancing mobile robot. *Journal of Ambient Intelligence and Humanized Computing, 14*(7), 9451–9467. https://doi.org/10.1007/s12652-023-04613-w

Maldonado Ramos, J. J., Garrido, R., y Castro Zavala, G. (2020). Real-time control prototype and its use in automatic control courses. In Memorias del Congreso Nacional de Control Automático (pp. 1–6).

Moritz, J., Musa, M., y Wejinya, U. (2024). Design and modeling of a two-wheeled differential drive robot. *International Journal of Control, Automation and Systems*, 22(7), 2273–2282. https://doi.org/10.1007/s12555-023-0288-8

Nguyen, V.-T., Duong, D.-N., y Dang, T.-V. (2025). Optimal two-wheeled self-balancing mobile robot strategy of navigation using adaptive fuzzy controller-based KD-SegNet. *Intelligent Service Robotics*, 18(3), 661–685. https://doi.org/10.1007/s11370-025-00606-0

Papadimitriou, K. D., Murasovs, N., Giannaccini, M. E., y Aphale, S. (2025). Genetic algorithm-based control of a two-wheeled self-balancing robot. *Journal of Intelligent y Robotic Systems*, 111(1), 34. https://doi.org/10.1007/s10846-025-02236-1

Pham, V.-C., Tran, N., Ngo, H.-H., Mac, P., y Tran, H. (2026). Robust optimal control for two-wheeled inverted pendulum via hierarchical sliding mode control and high-order disturbance observer with off-policy adaptive dynamic programming. *Journal of Control, Automation and Electrical Systems,* 37(4), 746–763. https://doi.org/10.1007/s40313-026-01280-y

Tsai, C.-C., Chan, C.-K., y Fan, Y.-H. (2008). Planned navigation of a self-balancing autonomous service robot. In 2008 IEEE Workshop on Advanced Robotics and Its Social Impacts (pp. 1–6). IEEE. https://doi.org/10.1109/ARSO.2008.4653617

Wang, F.-C., Chen, Y.-H., Wang, Z.-J., Liu, C.-H., Lin, P.-C., y Yen, J.-Y. (2021). Decoupled multi-loop robust control for a walk-assistance robot employing a two-wheeled inverted pendulum. *Machines,* 9(10), Article 205. https://doi.org/10.3390/machines9100205

Yang, S., Pang, H., Liu, L., Zheng, L., Wang, L., y Liu, M. (2024). A RISE-based asymptotic prescribed performance trajectory tracking control of two-wheeled self-balancing mobile robot. *Nonlinear Dynamics,* 112(17), 15327–15348. https://doi.org/10.1007/s11071-024-09569-w

Zhang, H., y Mohamad Nor, N. (2025). Control strategies for two-wheeled self-balancing robotic systems: A comprehensive review. *Robotics,* 14(8), 101. https://doi.org/10.3390/robotics14080101

Zhang, H., Mohamad Nor, N., y Umar, S. N. H. (2025). Modified dual hierarchical terminal sliding mode control design for two-wheeled self-balancing robot. *Electronics,* 14(13), 2692. https://doi.org/10.3390/electronics14132692

Zhao, J., Li, J., y Zhou, J. (2023). Research on two-round self-balancing robot SLAM based on the Gmapping algorithm. *Sensors,* 23(5), 2489. https://doi.org/10.3390/s23052489