

Tuning of a Fractional-Order PID Controllers for Differential Two-Wheeled Mobile Robot Using Particle Swarm Optimization *

González-Olvera, Marcos Angel^{*,**} Martínez-Vázquez, Elías *

* *Universidad Autónoma de la Ciudad de México, Colegio de Ciencia y
Tecnología (e-mail: marcos.angel.gonzalez@uacm.edu.mx,
elias.martinez.vasquez@estudiante.uacm.edu.mx)*
** *Corresponding author.*

Abstract: This paper presents a practical methodology for tuning a fractional-order PID (FOPID) controller applied to a two-wheeled differential mobile robot. Using the robot's kinematic model and the particle swarm optimization (PSO) method, the goal is to improve the system's control precision and stability. Numerical results show that the FOPID controller, compared to its integer-order PID counterpart, offers greater flexibility in parameter tuning and better trajectory tracking performance, achieving a reduction in error. However, challenges related to the FOPID implementation, such as higher computational demand, are also discussed. Future work includes the experimental validation of the results on a physical platform, allowing for the evaluation of the controller's performance under real-world conditions.

Keywords: Fractional-order PID, two-wheeled mobile robot, tuning, particle swarm optimization, robot control.

1. INTRODUCCIÓN

Los robots móviles de tracción diferencial son una configuración común en aplicaciones prácticas por su simplicidad y facilidad de uso en tareas como limpieza, transporte industrial, detección submarina, vigilancia y sillas de ruedas automáticas (Xie et al., 2018). El control de estos robots generalmente se basa en modelos cinemáticos, más sencillos que los dinámicos y adecuados para las bajas velocidades típicas.

El diseño del controlador es crucial para lograr el seguimiento de trayectorias definidas o desplazarse a una ubicación específica en un tiempo determinado (Parhi, 2005). Tecnologías como el análisis de imágenes o video pueden añadir una carga considerable en su desempeño (Kamel and Zhang, 2014).

Los controladores PID se usan ampliamente debido a su efectividad para lograr estabilidad y seguimiento Meng et al. (2018). A pesar de ciertas limitaciones, los controladores PID siguen siendo una solución robusta en diversos sistemas Barsan (2019). Sin embargo, la sintonización manual de los parámetros PID puede ser compleja debido a la dinámica no lineal del robot. Por ello, se han

explorado técnicas como el Algoritmo de Optimización por Enjambre de Partículas (PSO), redes neuronales y algoritmos evolutivos (Pak et al., 2023; Abajo et al., 2022), que han mostrado eficacia en la optimización de controladores PID (Mohamed and Hamza, 2019).

Para ampliar la flexibilidad en la sintonización, los controladores PID de orden fraccionario, conocidos como $PI^\lambda D^\mu$, se han empleado en diversas aplicaciones por ofrecer más parámetros ajustables para obtener un desempeño mejorado (Martinez-Alonso and González-Olvera, 2016). Sin embargo, su implementación en tiempo real presenta retos debido a los mayores requisitos computacionales y las dificultades en la sintonización (Abed et al., 2022; Shah and Agashe, 2016; Monje et al., 2008), limitando su adopción (Somefun et al., 2021).

Este artículo presenta una metodología para la sintonización de un controlador $PI^\lambda D^\mu$ para un robot móvil de dos ruedas, utilizando el algoritmo PSO para optimizar los parámetros del controlador fraccionario, mejorando así la precisión y estabilidad en el seguimiento de trayectorias. Se comparan los resultados con los obtenidos por controladores PID tradicionales y se discuten los desafíos de implementación.

Este trabajo está dividido en las siguientes partes: En la Sección 2 se discuten los antecedentes necesarios para comprender el uso de controladores PID de orden fraccionario y su sintonización, así como una intro-

* Los autores desean agradecer su apoyo para la realización de este trabajo a la Universidad Autónoma de la Ciudad de México (Colegio de Ciencia y Tecnología) mediante el proyecto UACM-CCYT-2023-IMP-05.

ducción al cálculo fraccionario. La Sección 3 presenta el planteamiento del problema, donde se describe el modelo cinemático del robot móvil de dos ruedas y el modelo de error asociado al seguimiento de trayectorias. En la Sección 4 se detallan los resultados numéricos obtenidos a partir de la sintonización del controlador $PI^\lambda D^\mu$ utilizando el algoritmo de optimización por enjambre de partículas (PSO), comparando su rendimiento con un controlador PID tradicional. Finalmente, en la Sección 5 se presentan las conclusiones del trabajo, discutiendo las ventajas y limitaciones del enfoque propuesto, así como posibles líneas de trabajo futuro.

2. ANTECEDENTES

2.1 Conceptos básicos de cálculo fraccionario

El cálculo de orden fraccionario es una generalización de la integración y la diferenciación a órdenes no enteros, y cuyo operador fundamental de orden fraccionario se denota como ${}_a D_t^\alpha$, donde a y t son los límites y α ($\alpha \in \mathbb{R}$) es el orden de la operación. En el caso de la integral de orden fraccionaria, para $n \in \mathbb{Z}^+ - \{0\}$, a partir de la fórmula de Cauchy para la integración múltiple es

$$\begin{aligned} J^n f(t) &\triangleq \int_a^t \int_a^{\tau_1} \dots \int_a^{\tau_{n-1}} f(\tau) d\tau \dots d\tau_2 d\tau_1 \\ &= \frac{1}{(n-1)!} \int_a^t f(\tau) (t-\tau)^{n-1} d\tau, \end{aligned} \quad (1)$$

donde al considerar a $n = \alpha$ como un valor no entero $\alpha \in \mathbb{R}$, se obtiene la llamada *integral fraccionaria de Riemman-Liouville*, dada por

$$J^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_a^t f(\tau) (t-\tau)^{\alpha-1} d\tau. \quad (2)$$

donde $\Gamma(w)$ es la función Gamma de $w \in \mathbb{C}$.

Por otro lado, la derivada de orden fraccionario, vista como un operador inverso por la izquierda de la integral de orden fraccionario, admite diferentes definiciones, y entre las más comunes se encuentran la definición de Grünwald-Letnikov (GL), la de Riemann-Liouville (RL) y la de Caputo (C).

Por su parte, la definición de Grünwald-Letnikov está definida mediante:

$${}_a D_t^\alpha f(t) = \lim_{h \rightarrow 0} h^{-\alpha} \sum_{j=0}^{\left(\frac{t-a}{h}\right)} (-1)^j \binom{\alpha}{j} f(t-jh) \quad (3)$$

donde (\cdot) denota la parte entera.

La definición de Riemann-Liouville está dada por:

$${}_a D_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau \quad (4)$$

para $n-1 < \alpha < n$ y donde $\Gamma(\cdot)$ es la función gamma de Euler.

Finalmente, la definición de Caputo es:

$${}_a D_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \int_a^t \frac{d^n}{d\tau^n} \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau \quad (5)$$

En el presente documento se considerará que $a = 0$, y en general condiciones iniciales nulas, por lo que se empleará la notación

$$D^\alpha = {}_a D_t^\alpha \quad (6)$$

en los casos donde no exista ambigüedad en la variable de integración.

Para mayor conveniencia, en el dominio de las frecuencia s , empleando la Transformada de Laplace, se suele utilizar la noción del operador integro-diferencial fraccionario. La transformada de Laplace de la derivada/integral fraccionaria de Riemann-Liouville (RL) y de Caputo (C) bajo condiciones iniciales nulas para el orden α ($0 < \alpha < 1$) está dada por:

$$\mathcal{L}\{{}_a D_t^\alpha f(t)\} = s^\alpha F(s) \quad (7)$$

donde, para $\alpha < 0$ se obtiene la integración de orden fraccionario, y para $\alpha > 0$ la operación de derivación.

2.2 Control PID de orden fraccionario

En la práctica del control, se considera al controlador de orden fraccionario aún cuando el modelo de la planta puede haber sido obtenido como un modelo de orden entero en el sentido clásico, y de esta forma obtener un sistema en lazo cerrado de orden fraccionario. Dada una planta con entrada de control $u(t)$, salida $y(t)$, señal de referencia $r(t)$ y el error entre las dos últimas $e(t) = r(t) - y(t)$, es bien conocido que el control PID en su forma ideal se encuentra dado por

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} + K_i \int_0^t e(\tau) d\tau. \quad (8)$$

cuya función de transferencia se encuentra expresada mediante

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + K_d s + K_i s^{-1}. \quad (9)$$

Si se consideran operaciones de orden fraccionario en la derivación y en la integración λ y μ respectivamente, se obtiene el controlador $PI^\lambda D^\mu$:

$$u(t) = k_p e(t) + k_i D^{-\lambda} e(t) + k_d D^\mu e(t), \quad (10)$$

que resulta en la función de transferencia

$$G_{cf}(s) = k_p + \frac{k_i}{s^\lambda} + k_d s^\mu, \quad (11)$$

donde λ y μ son los órdenes fraccionarios de las partes integral y derivativa del controlador, respectivamente. Para este caso, se considerará que $\lambda \in (0, 1)$ y $\mu \in (0, 1)$.

2.3 Optimización por Enjambre de Partículas

El algoritmo de optimización por enjambre de partículas (PSO) es un método computacional inspirado en el comportamiento colectivo de poblaciones de aves o insectos que buscan soluciones óptimas mediante una función de costo. Este algoritmo evalúa iterativamente posibles soluciones, denominadas *partículas*, y ajusta sus posiciones en función de su desempeño. Las partículas, representadas por vectores, simulan movimientos influenciados por inercia, disipación y comunicación entre ellas, dirigiéndose hacia las mejores posiciones conocidas tanto local como globalmente.

Considerando el vector de parámetros de solución $\theta = (\theta_1 \theta_2 \dots \theta_p)$, se inicializan $N_p \in \mathbf{Z}$ soluciones posibles en la primera iteración $\Theta_1 = \{\theta^{(1,1)}, \theta^{(1,2)}, \dots, \theta^{(1,N_p)}\}$. Cada i -ésima solución se modela como una partícula cuya posición en un espacio p -dimensional está definida por $\theta^{(i)}$. Si cada partícula tiene una masa m y está sujeta a una fricción representada por la fuerza $F_b = b\dot{\theta}^{(i)}$, y a fuerzas externas F_1 y F_2 que describen los comportamientos cognitivo y social, la dinámica se describe mediante:

$$m\ddot{\theta}^{(i)} + b\dot{\theta}^{(i)} = F_1^{(i)}(t) + F_2^{(i)}(t).$$

La velocidad de cada partícula está dada por:

$$v^{(i)} = \dot{\theta}^{(i)},$$

por lo que se tiene:

$$m\dot{v}^{(i)} + bv^{(i)} = F_1^{(i)}(t) + F_2^{(i)}(t).$$

Discretizando mediante el método de Euler, con $\dot{v}^{(i)}(t) \approx \frac{v^{(i)}(t+h) - v^{(i)}(t)}{h}$ y definiendo $v_k^{(i)} \triangleq v^{(i)}(kh)$, se obtiene:

$$\frac{m}{h} (v_{k+1}^{(i)} - v_k^{(i)}) + bv_k^{(i)} = F_{1,k} + F_{2,k}.$$

Las fuerzas externas, que modelan la interacción entre partículas, se definen como:

$$F_1(t) \propto w_1 u_1 (\theta_{min,g} - \theta^{(i)}(t)),$$

$$F_2(t) \propto w_2 u_2 (\theta_{min,l} - \theta^{(i)}(t)),$$

donde w_1 y w_2 son pesos de diseño asignados a cada fuerza, $\theta_{min,g}$ es el mínimo global encontrado por cualquier partícula, $\theta_{min,l}$ es el mínimo local en el vecindario de $\theta^{(i)}$, y u_1 y u_2 son valores aleatorios que facilitan la exploración.

Finalmente, la actualización de la velocidad de cada partícula se expresa como:

$v_{k+1}^{(i)} = Wv_k^{(i)} + w_1 u_1 (\theta_{min,g} - \theta_k^{(i)}) + w_2 u_2 (\theta_{min,l} - \theta_k^{(i)})$, y la posición de cada partícula se actualiza mediante:

$$\theta_{k+1}^{(i)} = \theta_k^{(i)} + v_{k+1}^{(i)}.$$

3. PLANTEAMIENTO DEL PROBLEMA

3.1 Modelo Cinemático del Robot

Considerando un robot móvil con tracción diferencial que se mueve a lo largo de la trayectoria ξ sin deslizamiento,

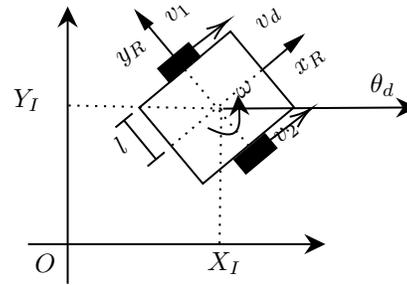


Fig. 1. Modelo del robot móvil diferencial.

con respecto del sistema de coordenadas global $\mathcal{O}_f : \{O_f, x_f, y_f\}$ como se muestra en la Fig. 1.

La relación entre las velocidades lineales v_1, v_2 y angulares del robot ω con las velocidades angulares de las ruedas ω_1 y ω_2 se determina mediante:

$$v_1 = r\omega_1 \quad (12)$$

$$v_2 = r\omega_2 \quad (13)$$

$$V_G = \frac{1}{2}(v_2 + v_1) = \frac{r}{2}(\omega_1 + \omega_2) \quad (14)$$

$$\dot{\theta} = \frac{1}{2}(v_2 - v_1) = \frac{r}{2\ell}(\omega_1 - \omega_2) \quad (15)$$

donde r es el radio de la rueda y 2ℓ es la distancia entre las dos ruedas, mientras que V_G y $\dot{\theta}$ son las velocidades lineales y angulares del robot, respectivamente.

Así, el modelo cinemático del robot en \mathcal{O}_f está dado por:

$$\dot{\mathbf{q}}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & 0 \\ \sin(\theta(t)) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_G(t) \\ \dot{\theta}(t) \end{bmatrix} \quad (16)$$

3.2 Modelo del Error Cinemático

El modelo de error describe la variación en posición y orientación del robot cuando se mueve a lo largo de la trayectoria deseada $\mathbf{q}_d(t)$ con respecto de \mathcal{O}_f , está definido por el vector de error $\mathbf{e}(t)$:

$$\mathbf{e}(t) = \mathbf{q}_d(t) - \mathbf{q}(t) = \begin{bmatrix} e_x(t) \\ e_y(t) \\ e_\theta(t) \end{bmatrix} \quad (17)$$

donde $\mathbf{q}_d(t) = \begin{bmatrix} x_d(t) \\ y_d(t) \\ \theta_d(t) \end{bmatrix}$ es la trayectoria deseada en el sistema de coordenadas global $\mathcal{C}_f\{O_{xyz}\}$.

Dado que $\mathcal{C}_r\{Gx_r y_r\}$ es el sistema de coordenadas asociado con el robot, se transforma $\mathbf{e}(t)$ de \mathcal{O}_f a \mathcal{O}_r mediante:

$$\mathbf{e}_r = R(\theta)\mathbf{e} \quad (18)$$

donde

$$R(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (19)$$

Al derivarla se obtiene:

$$\dot{\mathbf{e}}_r = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \dot{\mathbf{e}} + \begin{bmatrix} -\sin(\theta) & \cos(\theta) & 0 \\ -\cos(\theta) & -\sin(\theta) & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{e}(t) \quad (20)$$

Combinando ambas ecuaciones se obtiene:

$$\dot{\mathbf{e}}_r = \begin{bmatrix} \dot{e}_{x-r} \\ \dot{e}_{y-r} \\ \dot{e}_{\theta-r} \end{bmatrix} = \begin{bmatrix} V_d(t) \cos(e_{\phi-r}) - V_G(t) + \dot{\theta}(t)e_{y-r} \\ V_d(t) \sin(e_{\phi-r}) - \dot{\theta}(t)e_{x-r} \\ \dot{\theta}_d - \dot{\theta} \end{bmatrix} \quad (21)$$

Considerando la aproximación $\cos(e_{\theta-r}) \approx 1$ y $\sin(e_{\theta-r}) = e_{\theta-r}$, se obtiene, en forma matricial:

$$\dot{\mathbf{e}}_r = A_r \mathbf{e}_r + B \mathbf{u} \quad (22)$$

donde

$$A_r = \begin{bmatrix} 0 & \dot{\theta}_d(t) & 0 \\ -\dot{\theta}_d(t) & 0 & \dot{\theta}_d(t) \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (23)$$

con la señal de control dada por

$$\mathbf{u}(t) = \begin{bmatrix} e_V \\ e_{\dot{\theta}} \end{bmatrix} = \begin{bmatrix} V_d(t) - V_G(t) \\ \dot{\theta}_d(t) - \dot{\theta}(t) \end{bmatrix} \quad (24)$$

De manera similar al desarrollo llevado a cabo en (Thai et al., 2022; Padhy et al., 2010), se elige la ley de control PID de orden entero mediante:

$$e_V = K_{px}e_{x-r} + K_{ix} \int e_{x-r} dt + K_{dx} \frac{de_{x-r}}{dt} \quad (25)$$

$$e_{\dot{\theta}} = K_{py}e_{y-r} + K_{iy} \int e_{y-r} dt + K_{dy} \frac{de_{y-r}}{dt} + u_{\theta} \quad (26)$$

$$u_{\theta} = K_{p\theta}e_{\theta-r} + K_{i\theta} \int e_{\theta-r} dt + K_{d\theta} \frac{de_{\theta-r}}{dt} \quad (27)$$

Por su parte, al considerar operadores de orden fraccionario se obtiene el control:

$$e_V = K_{px}e_{x-r} + K_{ix}D^{-\lambda}e_{x-r} + K_{dx}D^{\mu}e_{x-r} \quad (28)$$

$$e_{\dot{\theta}} = K_{py}e_{y-r} + K_{iy}D^{-\lambda}e_{y-r} + K_{dy}D^{\mu}e_{y-r} + u_{\theta} \quad (29)$$

$$u_{\theta} = K_{p\theta}e_{\theta-r} + K_{i\theta}D^{-\lambda}e_{\theta-r} + K_{d\theta}D^{\mu}e_{\theta-r} \quad (30)$$

De esta forma, se establece la función de costo para evaluar el desempeño del controlador dada por

$$J(\Theta) = \int_0^T (e_{x-r}^2 + e_{y-r}^2) dt \quad (31)$$

donde T es el horizonte de tiempo para evaluación del desempeño y Θ es el conjunto de parámetros del controlador, que en el caso del PID de orden entero es

$$\Theta_{PID} = (K_{px}, K_{ix}, K_{dx}, K_{py}, K_{iy}, K_{dy}, K_{p\theta}, K_{i\theta}, K_{d\theta}),$$

y

$$\Theta_{FOPID} = (K_{px}, K_{ix}, K_{dx}, K_{py}, K_{iy}, K_{dy}, K_{p\theta}, K_{i\theta}, K_{d\theta}, \lambda, \mu)$$

el conjunto de parámetros a sintonizar para el caso del FOPID.

4. RESULTADOS NUMÉRICOS

Con el fin de obtener una sintonización y comparación del desempeño entre el PID y el FOPID, y simplificar el esquema de optimización, se consideró un conjunto reducido de parámetros para el caso del PID mediante $K_{px} = K_{py}$, $K_{ix} = K_{iy}$, $K_{dx} = K_{dy}$ y $K_{i\theta} = K_{d\theta} = 0$. Para el modelo se consideraron los parámetros del robot QUANSER©QBot 2, donde $r = 0.05$ m y $\ell = 0.15$ m. En todos los casos, se consideró una optimización de (31) con $T = 60$ s, siguiendo una trayectoria descrita por $x_d = \sin(0.2t)$, $y_d = \sin(0.4t)$, así como intervalos de búsqueda de parámetros dada por $K_{px} \in [15, 25]$, $K_{ix} \in [15, 25]$, $K_{dx} \in [0.5, 2]$ y $K_{p\theta} \in [0.5, 2]$. En el caso del controlador FOPID, se consideraron además los intervalos $\mu, \lambda \in [0.8, 0.99]$. Las simulaciones fueron realizadas mediante SIMULINK en MATLAB 2023b, y en particular los operadores de orden fraccionario fueron implementados mediante FOMCON Toolbox for MATLAB (Tepljakov, 2020).

Con el fin de evaluar el desempeño con condiciones un tanto más apegadas a un ambiente experimental, y considerando que este sistema será evaluado posteriormente mediante un esquema de maestro-esclavo mediante control por visión, se empleó un tiempo de muestreo cercano de 200 ms para determinar la acción de control.

Tras aplicar el algoritmo PSO a ambos controladores, se obtuvieron los parámetros mostrados en la Tabla 1. En las Figuras 2 y 3 se muestra el desempeño logrado al sintonizar el controlador PID de orden entero, mientras que en las Figuras 4 y 5. En el primer caso el error RMS con respecto al seguimiento de la trayectoria dado con la evaluación de (31) fue de 0.0920, mientras que en el segundo se disminuye a 0.0815, lo que supone una mejora de más del 11 % en el desempeño.

	$K_{px,y}$	$K_{ix,i}$	$K_{dx,y}$	$K_{p\theta}$	λ	μ
PID	19.72	18.07	1.80	1.72	-	-
FOPID	16.78	21.92	0.6684	1.98	0.9537	0.8425

Table 1. Resultados tras aplicar el PSO para la sintonización del control PID y FOPID.

Para evaluar el desempeño ante diferentes trayectorias se evaluó también el seguimiento ante $x_d = \cos(0.4t) - 1$, $y_d = \sin(0.4t)$, los cuales se muestran en las Fig. 6 para el PID de orden entero y en la Fig. 7 para el FOPID, donde se puede observar cómo, en el caso del control de orden fraccionario, la convergencia es relativamente más lenta en el caso fraccionario que en el entero.

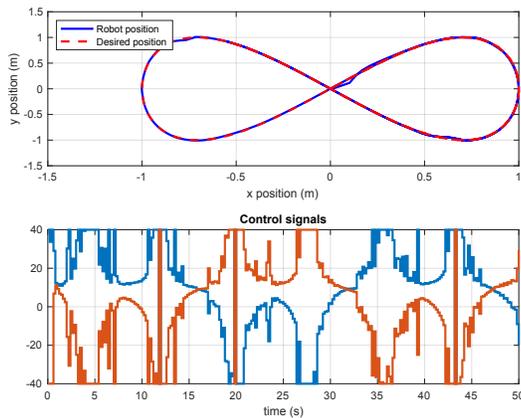


Fig. 2. Resultados del control PID sintonizado mediante el algoritmo PSO. Se muestra la trayectoria deseada y las señales de control enviadas a las dos ruedas

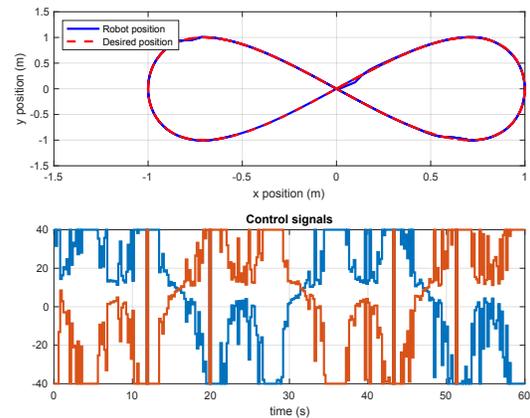


Fig. 4. Resultados del control PID de orden fraccionario sintonizado mediante el algoritmo PSO. Se muestra la trayectoria deseada y las señales de control enviadas a las dos ruedas.

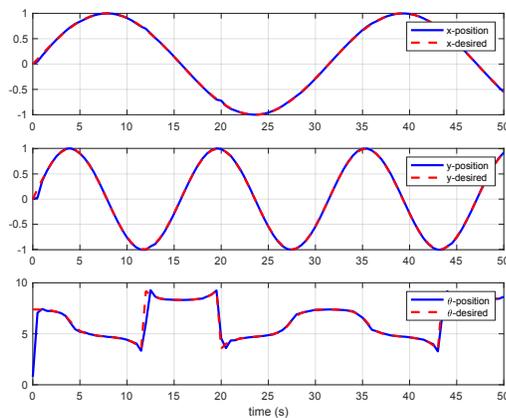


Fig. 3. Resultados del control PID sintonizado mediante el algoritmo PSO. Se muestra la trayectoria deseada contra el tiempo en las posiciones espaciales y el ángulo de orientación.

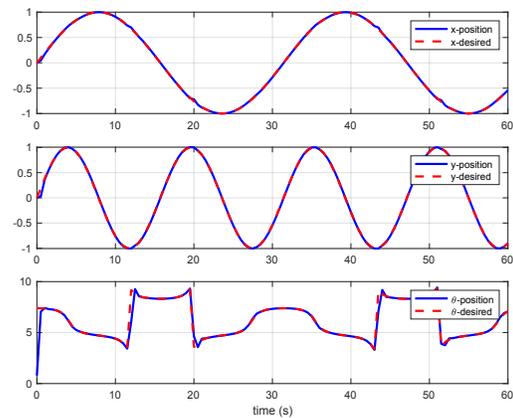


Fig. 5. Resultados del control PID de orden fraccionario sintonizado mediante el algoritmo PSO. Se muestra la trayectoria deseada contra el tiempo en las posiciones espaciales y el ángulo de orientación.

5. CONCLUSIONES

En este trabajo se presentó una metodología para la sintonización de un controlador PID de orden fraccionario utilizando el método de optimización por enjambre de partículas, aplicado a un robot móvil de dos ruedas diferenciales, y se compararon tanto la metodología como los resultados del controlador optimizado con aquellos obtenidos para un PID de orden entero. Los resultados numéricos muestran que el controlador propuesto mejora la precisión y estabilidad del seguimiento de trayectoria en comparación con los controladores PID tradicionales. Es de notar que, la complejidad en su implementación y los mayores requisitos computacionales representan un desafío importante en el ámbito computacional, en vista de que requieren un mayor tiempo de cálculo que su contraparte entera.

Como trabajo futuro se considera llevar a cabo la implementación de estos controladores en una plataforma experimental para validar los resultados obtenidos en simulaciones y evaluar su desempeño en condiciones reales, en vista de que se cuenta con parte de la plataforma. Esta validación experimental permitirá evaluar y mejorar la elección de parámetros y limitaciones físicas de los actuadores y sensores.

6. AGRADECIMIENTOS

Los autores desean agradecer su apoyo para la realización de este trabajo a la Universidad Autónoma de la Ciudad de México (Colegio de Ciencia y Tecnología) mediante el proyecto UACM-CCYT-2023-IMP-05.

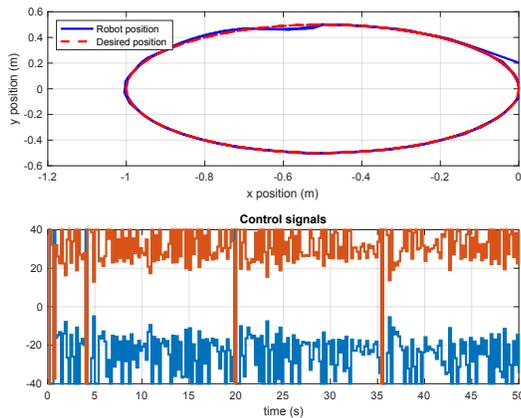


Fig. 6. Resultados del control PID de orden entero sintonizado mediante el algoritmo PSO para una trayectoria circular.

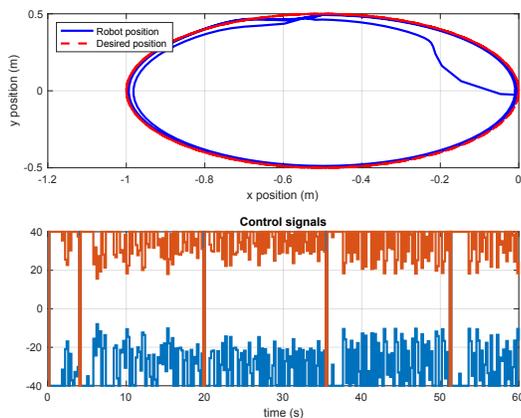


Fig. 7. Resultados del control PID de orden fraccionario sintonizado mediante el algoritmo PSO para una trayectoria circular.

REFERENCES

- Abajo, M.R., Sierra-García, J.E., and Santos, M. (2022). Evolutive tuning optimization of a pid controller for autonomous path-following robot. In *16th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2021)*, 451–460. Springer.
- Abed, A.M., Rashid, Z.N., Abedi, F., Zeebaree, S.R., Sahib, M.A., Mohamad Jawad, A.J., Redha Ibrahim, G.A., Maher, R.A., Abdulkareem, A.I., Ibrahim, I.K., et al. (2022). Trajectory tracking of differential drive mobile robots using fractional-order proportional-integral-derivative controller design tuned by an enhanced fruit fly optimization. *Measurement and Control*, 55(3-4), 209–226.
- Barsan, A. (2019). Position control of a mobile robot through pid controller. *Acta Universitatis Cibiniensis. Technical Series*, 71(1), 14–20.
- Kamel, M.A. and Zhang, Y. (2014). Developments and challenges in wheeled mobile robot control. In *International Conference on Intelligent Unmanned Systems (ICIUS)*.
- Martinez-Alonso, E.S. and González-Olvera, M.A. (2016). Control fraccionario para un péndulo invertido. In *Memorias del Congreso Nacional de Control Automático*.
- Meng, J., Liu, A., Yang, Y., Wu, Z., and Xu, Q. (2018). Two-wheeled robot platform based on pid control. In *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*, 1011–1014. IEEE.
- Mohamed, M.J. and Hamza, M.K. (2019). Design pid neural network controller for trajectory tracking of differential drive mobile robot based on pso. *Engineering and Technology Journal*, 37(12A), 574–583.
- Monje, C.A., Vinagre, B.M., Feliu, V., and Chen, Y. (2008). Tuning and auto-tuning of fractional order controllers for industry applications. *Control engineering practice*, 16(7), 798–812.
- Padhy, P.K., Sasaki, T., Nakamura, S., and Hashimoto, H. (2010). Modeling and position control of mobile robot. In *2010 11th IEEE International Workshop on Advanced Motion Control (AMC)*, 100–105. IEEE.
- Pak, Y.J., Kong, Y.S., and Ri, J.S. (2023). Robust pid optimal tuning of a delta parallel robot based on a hybrid optimization algorithm of particle swarm optimization and differential evolution. *Robotica*, 41(4), 1159–1178.
- Parhi, D.R. (2005). Navigation of mobile robots using a fuzzy logic controller. *Journal of intelligent and robotic systems*, 42, 253–273.
- Pati, C.S. and Kala, R. (2017). Vision-based robot following using pid control. *Technologies*, 5(2), 34.
- Shah, P. and Agashe, S. (2016). Review of fractional pid controller. *Mechatronics*, 38, 29–41.
- Somefun, O.A., Akingbade, K., and Dahunsi, F. (2021). The dilemma of pid tuning. *Annual Reviews in Control*, 52, 65–74.
- Tepljakov, A. (2020). Fomcon toolbox for matlab.
- Thai, N.H., Ly, T.T.K., Thien, H., and Dzung, L.Q. (2022). Trajectory tracking control for differential-drive mobile robot by a variable parameter pid controller. *International Journal of Mechanical Engineering and Robotics Research*, 11(8), 614–621.
- Xie, D., Wang, S., and Wang, Y. (2018). Trajectory tracking control of differential drive mobile robot based on improved kinematics controller algorithm. In *2018 Chinese Automation Congress (CAC)*, 2675–2680. IEEE.