Memorias del 2024 Congreso Nacional de Control Automático
8-11 de Octubre, 2024. Ciudad de México, México.

Trajectory tracking control of a mecanum-wheeled omnidirectional mobile robot implemented in ROS-Gazebo ⋆

L. López-Cancino * **M. Luna-Aguilar** * **S. Gómez-Peñate** *
I. Santos-Ruiz * **S. Molina-Domínguez** *

* *Tecnológico Nacional de México, IT Tuxtla Gutiérrez, TURIX-Dynamics Diagnosis and Control Group, Carr. Panamericana km 1080, Tuxtla Gutiérrez 29050, México.*

Abstract:
This article presents a trajectory tracking robust control for a mobile robot with Mecanum wheels. Using the sliding mode technique, a controller is developed to ensure robustness against unknown uncertainties, providing stable convergence. For stability conditions, the direct Lyapunov method is used, which guarantees the asymptotic convergence of the system. Finally, the results are evaluated at the simulation level using the nonlinear model and ROS2 in the Gazebo environment.

Keywords: Omnidirectional mobile robot, Sliding mode control, Trajectory Tracking, ROS2, Gazebo.

1. INTRODUCCIÓN

Los robots móviles omnidireccionales han ganado popularidad debido a su capacidad para moverse en cualquier dirección sin cambiar su orientación, proporcionando una gran flexibilidad y maniobrabilidad (Leong et al., 2022) esto se logra mediante la incorporación de cuatro ruedas mecanum, las cuales tienen rodillos incrustados en su circunferencia dispuestos a 45°. Estos rodillos permiten realizar movimientos diagonales y laterales, haciendo que estos robots robots sean especialmente útiles para maniobrar en espacios reducidos como fábricas (Li et al., 2018) y en aplicaciones de movilidad asistida en sillas de ruedas (Yadav et al., 2022).

Para aprovechar al máximo las capacidades de los robots móviles con ruedas mecanum (RMRM), es crucial contar con un algoritmo de control de seguimiento de trayectorias. Este algoritmo es un componente fundamental en el desarrollo de sistemas robóticos móviles que emplean este tipo de ruedas. Sin embargo, el diseño de un controlador efectivo presenta desafíos significativos debido a la dinámica compleja y no lineal del sistema (Sabouri and Asemani, 2021).

Una estrategia prometedora para abordar estos desafíos es el control por modos deslizantes (SMC, por sus siglas en inglés). Este enfoque ha emergido como una solución robusta y eficaz para problemas de control en sistemas no lineales y con incertidumbres (Lu et al., 2018). El SMC utiliza una estructura de control basada en superficies deslizantes que obliga al sistema a alcanzar y mantener una trayectoria deseada, incluso en presencia de perturbaciones externas y variaciones en los parámetros del sistema (Yadav et al., 2022; Wang et al., 2024).

En este artículo, se presenta un enfoque de control por modos deslizantes para el seguimiento de trayectorias en robots con ruedas mecanum, utilizando el modelo no lineal del sistema. Para validar los resultados del controlador, se emplea el entorno Gazebo, una plataforma de simulación robusta y versátil ampliamente utilizada en simulaciones de VANT (Wang et al., 2021; Can et al., 2022). Gazebo permite la creación de entornos complejos y realistas, tanto del entorno en el que se moverá el robot como la integración de la física del RMRM, previamente diseñado en un software de diseño mecánico. Esto facilita la evaluación y mejora del desempeño de los controladores en situaciones que serían difíciles de recrear físicamente (Yogi et al., 2023), convirtiéndose en una

⋆ Este trabajo fue desarrollado en el marco de las actividades de la red internacional denominada "Red internacional de control y cómputo aplicados" soportado por el TecNM y financiado por el Tecnológico Nacional de México a través del programa Proyectos de Investigación Científica, Desarrollo Tecnológico e Innovación, con clave de proyecto 19268.24-P.

https://doi.org/10.58571/CNCA.AMCA.2024.082

herramienta valiosa para la verificación y análisis de algoritmos de control en robótica.

2. MODELO MATEMÁTICO DEL RMRM

El RMRM cuenta con cuatro ruedas tipo mecanum como se muestra en la Figura 1, en el esquema se tienen tres marcos de referencia, $\{Y_g,\ O_g,\ X_g\}$ es el marco de referencia global en un punto de la tierra, $\{Y_m,\ O_m,\ X_m\}$ es el marco de referencia en el centro de masa del RMRM y $\{Y_{r_i},\ O_{r_i},\ X_{r_i}\}$ con $i = 1, \ldots, 4$, son los marcos de referencia con respecto el centro geométrico de cada rueda, donde V_{r_i} es la velocidad generada por el rodillo de la rueda i, ω_i es la posición angular de la rueda i, R denota el radio de la rueda y ψ_i es el ángulo de inclinación del rodillo en cada rueda.

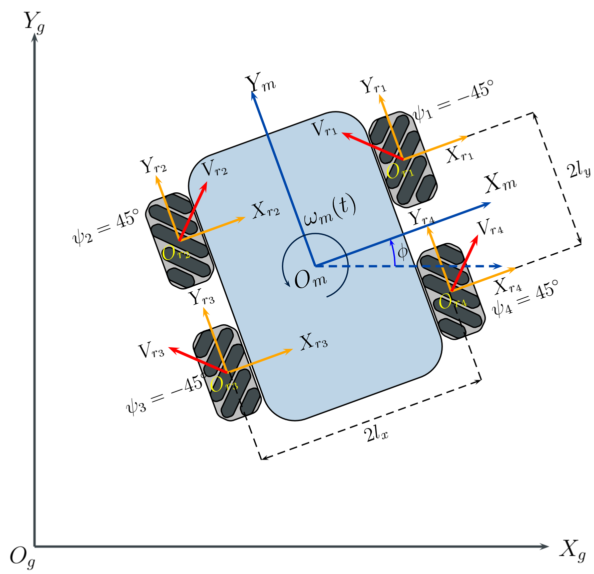

Fig. 1. Diagrama de cuerpo libre del robot omnidireccional mecanum.

Los componentes de las velocidades V_{r_i}, con respecto al marco de referencia O_{r_i} es:

$$\begin{bmatrix} \dot{x}_{r_i}(t) \\ \dot{y}_{r_i}(t) \end{bmatrix} = \begin{bmatrix} 0 & \sin(\psi_i) \\ R_r & -\cos(\psi_i) \end{bmatrix} \begin{bmatrix} \omega_i(t) \\ V_{r_i}(t) \end{bmatrix}. \tag{1}$$

Las velocidades de cada rueda en O_{ri} con respecto a O_m se puede escribir como:

$$\begin{bmatrix} \dot{x}_{m_i}(t) \\ \dot{y}_{m_i}(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} \dot{x}_{r_i}(t) \\ \dot{y}_{r_i}(t) \end{bmatrix}, \tag{2}$$

donde $\dot{x}_{m_i}(t)$ y $\dot{y}_{m_i}(t)$ son velocidades de las ruedas en el marco O_m y θ_i $(i = 1, 2, 3, 4)$ es el ángulo de rotación de O_m con respecto a O_{r_i}, para este caso $\theta_i = 0$. El vehículo se desplaza sobre terreno llano y, por tanto:

$$\begin{bmatrix} \dot{x}_{m_i}(t) \\ \dot{y}_{m_i}(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & -p_{y_i} \\ 0 & 1 & p_{x_i} \end{bmatrix} \begin{bmatrix} \dot{x}_m(t) \\ \dot{y}_m(t) \\ \dot{\omega}_m(t) \end{bmatrix}, \tag{3}$$

donde $[x_m(t)\ y_m(t)\ \omega_m(t)]^T$ es la posición del RMRM en O_m, y (p_{x_i}, p_{y_i}) determina la posición del centro de cada rueda respecto al marco de referencias O_m.

Tomando en cuenta las ecuaciones anteriores (1)-(3), se puede obtener:

$$\begin{bmatrix} \dot{\omega}_i(t) \\ V_{r_i}(t) \end{bmatrix} = \begin{bmatrix} 0 & \sin(\psi_i) \\ R_r & -\cos(\psi_i) \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & -p_{y_i} \\ 0 & 1 & p_{x_i} \end{bmatrix} \begin{bmatrix} \dot{x}_m(t) \\ \dot{y}_m(t) \\ \dot{\omega}_m(t) \end{bmatrix}. \tag{4}$$

Tomando en cuenta la primer fila de las ecuaciones denotadas en (4), la relación que existe entre las velocidades angulares de cada una de las ruedas y la velocidad del robot quedan definidas como:

$$\dot{\omega}_i(t) = \frac{1}{R} \begin{bmatrix} \frac{1}{\tan(\psi_i)} & 1 & p_{x_i} - (\frac{1}{\tan(\psi_i)})p_{y_i} \end{bmatrix} \begin{bmatrix} \dot{x}_m(t) \\ \dot{y}_m(t) \\ \dot{\omega}_m(t) \end{bmatrix}. \tag{5}$$

Nótese que en la Figura 1, los ángulos $\psi_1 = \psi_3 = -45$ y $\psi_2 = \psi_4 = 45$ y las posiciones $(p_{x_1}, p_{y_1}) = (l_x, l_y)$, $(p_{x_2}, p_{y_2}) = (-l_x, l_y)$, $(p_{x_3}, p_{y_3}) = (-l_x, -l_y)$, $(p_{x_4}, p_{y_4}) = (l_x, -l_y)$; por lo que la relación entre las velocidades con respecto a O_m y las velocidades angulares de las ruedas esta dada por:

$$\begin{bmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_3(t) \\ \omega_4(t) \end{bmatrix} = \frac{1}{R} \begin{bmatrix} -1 & 1 & l_x + l_y \\ 1 & 1 & -l_x - l_y \\ -1 & 1 & -l_x - l_y \\ 1 & 1 & l_x + l_y \end{bmatrix} \begin{bmatrix} \dot{x}_m(t) \\ \dot{y}_m(t) \\ \dot{\omega}_m(t) \end{bmatrix} \tag{6}$$

La cinemática directa del RMRM se define como (Alakshendra and Chiddarwar, 2017):

$$\overbrace{\begin{bmatrix} \dot{x}_m(t) \\ \dot{y}_m(t) \\ \dot{\omega}_m(t) \end{bmatrix} = \frac{R}{4} \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ \frac{1}{l_x+l_y} & -\frac{1}{l_x+l_y} & -\frac{1}{l_x+l_y} & \frac{1}{l_x+l_y} \end{bmatrix} \begin{bmatrix} \dot{\omega}_1(t) \\ \dot{\omega}_2(t) \\ \dot{\omega}_3(t) \\ \dot{\omega}_4(t) \end{bmatrix}}^{\text{Cinemática directa}}. \tag{7}$$

Las velocidades del RMRM en el marco global $\{X_g$, O_g, $Y_g\}$, se puede obtener mediante:

$$\begin{bmatrix} \dot{x}_g(t) \\ \dot{y}_g(t) \\ \dot{\phi}_g(t) \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\phi(t)) & -\sin(\phi(t)) & 0 \\ \sin(\phi(t)) & \cos(\phi(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R_\phi} \begin{bmatrix} \dot{x}_m(t) \\ \dot{y}_m(t) \\ \dot{\omega}_m(t) \end{bmatrix}, \tag{8}$$

donde R_ϕ es la matriz de rotación del vehículo, que relaciona las velocidades en el marco de referencia O_g con respecto a O_m.

Dado que los motores son los responsables de generar las fuerzas motrices que mueven el RMRM, el voltaje aplicado a cada motor constituye las entradas de control del sistema de seguimiento de trayectoria. En este

https://doi.org/10.58571/CNCA.AMCA.2024.082

contexto, es importante considerar la dinámica de las ruedas en función del voltaje aplicado (Monasterio-Huelin and Gutiérrez, 2020), considerando todos los motores idénticos, se tiene:

$$I_q\ddot{\omega}(t) + \frac{k_a\ k_b}{R_a}\dot{\omega}(t) + b\dot{\omega}(t) = \frac{k_a}{R_a}v(t), \qquad (9)$$

donde I_q es el momento de inercia del robot sobre su centro de gravedad, k_a es la constante de torque electromagnético, k_b es una constante electromotriz, R_a es la resistencia de armadura y b es el coeficiente de fricción, $\omega(t) = [\omega_1(t)\ \omega_2(t)\ \omega_3(t)\ \omega_4(t)]^\top$ representan las posiciones angulares de cada una de las ruedas, $v(t) = [v_1(t)\ v_2(t)\ v_3(t)\ v_4(t)]^\top$ son los voltajes de los motores.

3. DISEÑO DE CONTROL POR MODOS DESLIZANTES

Sustituyendo (7) en (8), se obtiene una representación en las coordenadas globales, proporcionando una descripción del movimiento y orientación del RMRM en O_g.

$$\dot{z}(t) = \frac{R}{4}h(\phi)\dot{\omega}(t), \qquad (10)$$

donde $z(t) = \left[\dot{x_g}(t)\ \dot{y_g}(t)\ \dot{\phi_g}(t)\right]^T$ y

$$h(\phi) = \sqrt{2}\begin{bmatrix} -\sin\phi & \cos\phi & -\sin\phi & \cos\phi \\ \cos\phi & \sin\phi & \cos\phi & \sin\phi \\ \frac{1}{\sqrt{2}(a+b)} & -\frac{1}{\sqrt{2}(a+b)} & -\frac{1}{\sqrt{2}(a+b)} & \frac{1}{\sqrt{2}(a+b)} \end{bmatrix}, \qquad (11)$$

con $\phi = \phi_g(t) + \frac{\pi}{4}$, de tal modo que la segunda derivada de $z(t)$, esta definida como:

$$\ddot{z}(t) = \frac{R}{4}h(\phi)'\dot{\phi}(t)\dot{\omega}(t) + \frac{R}{4}h(\phi)\ddot{\omega}(t), \qquad (12)$$

reescribiendo (12) de la siguiente manera:

$$\ddot{z}(t) = \frac{R}{4I_q}h(\phi)(H\dot{\phi}(t)\dot{\omega}(t) + I_q\ddot{\omega}(t)), \qquad (13)$$

donde

$$H = \begin{bmatrix} 0 & \frac{I_q}{2} & 0 & \frac{I_q}{2} \\ -\frac{I_q}{2} & 0 & -\frac{I_q}{2} & 0 \\ 0 & \frac{I_q}{2} & 0 & \frac{I_q}{2} \\ -\frac{I_q}{2} & 0 & -\frac{I_q}{2} & 0 \end{bmatrix}, \qquad (14)$$

y considerando la dinámica de los motores en (9) (Sun et al., 2021), se tiene:

$$\ddot{z}(t) = \frac{R}{4I_q}h(\phi)(H\dot{\phi}(t)\dot{\omega}(t) - \frac{ka\ kb}{Ra}\dot{\omega}(t) - b\dot{\omega}(t) + \frac{ka}{Ra}v(t)), \qquad (15)$$

definiendo una nueva entrada como:

$$u(t) = H\dot{\phi}(t)\dot{\omega}(t) - \frac{ka\ kb}{Ra}\dot{\omega}(t) - b\dot{\omega}(t) + \frac{ka}{Ra}v. \qquad (16)$$

el modelo en (15), queda expresado de la siguiente forma:

$$\ddot{z}(t) = \frac{R}{4I_q}h(\phi)u(t). \qquad (17)$$

Para el diseño de control del seguimiento de trayectoria se define el error de seguimiento $e(t)$:

$$e(t) = \begin{bmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \end{bmatrix} = z(t) - z_d(t) = \begin{bmatrix} x_g(t) - x_d(t) \\ y_g(t) - y_d(t) \\ \phi_g(t) - \phi_d(t) \end{bmatrix} \qquad (18)$$

donde $z_d(t) = [x_d(t)\ y_d(t)\ \phi_d(t)]^\top$ es la trayectoria deseada que debe seguir el RMRM.

Como parte del diseño del SMC, se define una superficie de deslizamiento $s(t)$ definida como:

$$s(t) = k_p e(t) + k_i\int e(t)\,dt + k_d\dot{e}(t) \qquad (19)$$

donde $k_p = \mathrm{diag}(k_{p1}, k_{p2}, k_{p3})$, $k_i = \mathrm{diag}(k_{i1}, k_{i2}, k_{i3})$, $k_d = \mathrm{diag}(k_{d1}, k_{d2}, k_{d3})$ son matrices de ganancias de la superficie de deslizamiento.

Mediante la resolución de la ecuación de $s(t) = 0$, ignorando todas las incertidumbres, se puede obtener la entrada de control equivalente $u_{eq}(t)$ como:

$$u_{eq}(t) = \frac{4I_q}{R}h(\phi)^{-1}k_d^{-1}(\ddot{z}_d(t)k_d - k_p\dot{e}(t) - k_i e(t)) \qquad (20)$$

Con el objetivo de adquirir una propiedad de convergencia estable y asegurar una robustez frente a incertidumbres desconocidas, se diseña una entrada de control de alcance $u_{re}(t)$ para el sistema:

$$u_{re}(t) = -h(\phi)^{-1}k\frac{s(t)}{\|s(t)\| + \epsilon}, \qquad (21)$$

donde k son los parámetros de control que se deben elegir, la señal de control es la suma de entrada de control equivalente y la de control de alcance: $u(t) = u_{eq}(t) + u_{re}(t)$.

Parámetro	Símbolo	Unidad	Valor
Masa del robot	M	kg	2.102
Momento de inercia del robot	I_q	kg m^2	0.0945
Radio de la rueda	R	m	0.05
Ancho de la plataforma	$2l_x$	m	0.22
Largo de la Plataforma	$2l_y$	m	0.36
Coeficiente de fricción	β_x	-	0.02
Fricción Viscosa	b		0.2
Coeficiente de torque del motor	k_a		0.183
Constante de fuerza electromotriz	k_b		0.208
Resistencia de armadura	R_a	Ω	4.29
Coeficiente del motor	α	N/V	0.96
Coeficiente del motor	β	kg/s	2.64

Table 1. Valores de los parámetros

3.1 Estabilidad

Para verificar la estabilidad del controlador, se propone una función de Lyapunov V dada por:

$$V(t) = s^{\top}(t)s(t), \tag{22}$$

donde la primer derivada de $s(t)$ se puede obtener de la siguiente forma:

$$\dot{s}(t) = k_p\ \dot{e}(t) + k_i\ e(t) + k_d\ \ddot{e}(t). \tag{23}$$

Sustituyendo 17, 20, 21 en 23

$$\begin{aligned}
\dot{s}(t) =& (\ddot{z}(t) - \ddot{z}_d(t))k_d + k_p\dot{e}(t) + k_i\dot{e}(t) \\
\dot{s}(t) =& \left(\frac{R}{4I_q}h(\phi)u(t) - \ddot{z}_d(t)\right)k_d + k_p\dot{e}(t) + k_i\dot{e}(t) \\
\dot{s}(t) =& \left(\frac{R}{4I_q}h(\phi)\Big(\big(\frac{4I_q}{R}h(\phi)^{-1}k_d^{-1}(\ddot{z}_d(t)k_d - k_p\dot{e}(t) \right. \\
& \left. -k_i e(t))\big) - h(\phi)^{-1}k\frac{s(t)}{\|s(t)\|+\epsilon}\Big) - \ddot{z}_d(t)\right)k_d \\
& + k_p\dot{e}(t) + k_i\dot{e}(t) \\
\dot{s}(t) =& -\frac{Rk}{4I_q}\frac{s(t)}{\|s(t)\|+\epsilon},
\end{aligned} \tag{24}$$

entonces, la derivada de la función de Lyapunov con respecto al tiempo es:

$$\begin{aligned}
\dot{V}(t) =& -s^T(t)\frac{Rk}{4I_q}\frac{s(t)}{\|s(t)\|+\epsilon} - \frac{Rk}{4I_q}\frac{s^T(t)}{\|s(t)\|+\epsilon}s(t) \\
=& -\frac{Rk}{2I_q}\frac{s^T(t)s(t)}{\|s(t)\|+\epsilon}
\end{aligned} \tag{25}$$

Para el análisis de estabilidad, necesitamos que $\dot{V}(t)$ sea negativa. Dado que R, k, I_q son positivos, y considerando que $\frac{s^T(t)\ s(t)}{\|s(t)\|+\epsilon}$ es positivo, se puede concluir que $\dot{V}(t)$ es efectivamente negativa.

4. SIMULACIÓN MATLAB, ROS-GAZEBO

En un control en modo deslizante, el objetivo es forzar a las trayectorias del estado a moverse a lo largo de una superficie deslizante $s(t) = 0$. La estructura del controlador es presentada en la Figura 2:

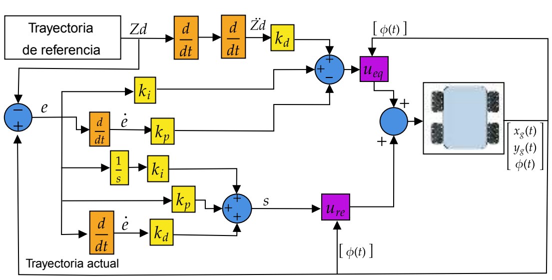

Fig. 2. Diagrama a bloques del controlador propuesto

En el diagrama del controlador, se utiliza una trayectoria deseada que esta dada por la siguiente expresión:

$$\begin{bmatrix} x_d(t) \\ y_d(t) \\ \phi_d(t) \end{bmatrix} = \begin{bmatrix} 2\cos(0.2t) \\ 1\sin(0.4t) \\ 0 \end{bmatrix}, \tag{26}$$

La ecuación (26) describe una lemniscata. Esta trayectoria ha sido utilizada tanto para la simulación en Matlab como para la simulación en Gazebo a través de Matlab-ROS. En ambas plataformas, la misma trayectoria se implementa para analizar y comparar los resultados de las simulaciones, asegurando así la coherencia y precisión de los modelos desarrollados. Donde los parámetros de sintonización para cada matriz, son los siguientes:

$$k = \begin{bmatrix} 500 & 0 & 0 \\ 0 & 500 & 0 \\ 0 & 0 & \frac{400}{l_x+l_y} \end{bmatrix}, k_p = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}, k_d = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$k_i = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}.$$

4.1 Simulación en el entorno Simulink-ROS-Gazebo

Gazebo es una plataforma de simulación robusta que ofrece un alto nivel de realismo en sus propiedades físicas, los sensores y los entornos, lo que permite una simulación precisa de robots en diversas condiciones(Azar et al., 2023), además de contar con una integración nativa con ROS, facilitando en gran manera el desarrollo y prueba de algoritmos en entornos simulados antes de implementarlo en un hardware real. En la Figura 4 se muestra el diseño del RMRM al cual se le integrara el algoritmo de control diseñado.

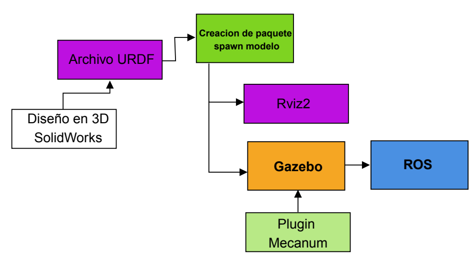

Fig. 3. Creación del entorno

En la Figura 3 se ejemplifica la creación del entorno de simulación, desde un diseño en 3D, al cual a través de una herramienta de SolidWorks se genera el archivo URDF que es el que contiene los enlaces y articulaciones de las piezas, este archivo junto con los STL de cada una de las piezas fueron implementadas en la creación del paquete, donde se creo el directorio "\robot_mecanum" en donde se adjuntan todas las piezas dentro de una carpeta llamada meshes, posteriormente fue necesario modificar las propiedades físicas dentro del archivo URDF, en donde se modificaron tanto las inercias, masas, coeficientes de rozamientos y de fricción de cada una de las piezas generadas, dentro de este mismo archivo fue necesario incluir un plugin "mecanum_drive" para el control de los movimientos del carro, en este mismo plugin fue necesaria la inclusión y modificación de

https://doi.org/10.58571/CNCA.AMCA.2024.082

los tópicos que genera el modelo dentro de Gazebo, específicamente los tópicos "\cmd_vel" y "\odom", que son necesarios par ala implementación del controlador. además de crear el archivo launch para Gazebo.

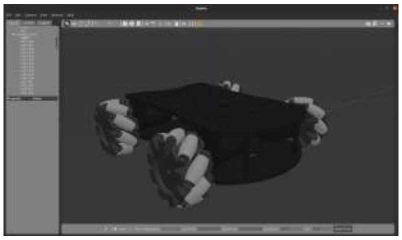

Fig. 4. Simulación en Gazebo

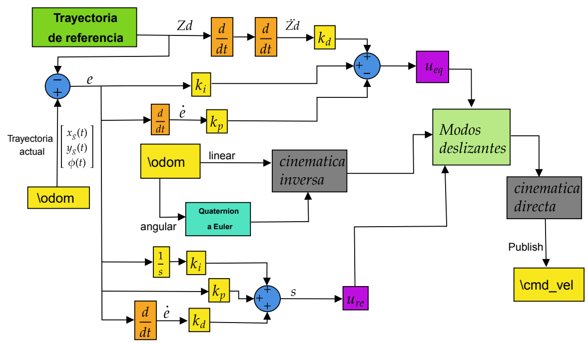

Fig. 5. Implementación en el entorno Simulink-ROS-Gazebo

En el entorno de simulación de Simulink de MATLAB, se utilizó el toolbox de ROS2 para configurar adecuadamente los bloques de suscripción y publicación a los tópicos previamente mencionados. Este proceso implicó el uso de 'bus selectors' para extraer los elementos específicos que contienen los datos relevantes de la pose del robot, así como las velocidades lineales y angulares. La configuración adecuada de estos bloques es importante ya que asegura una comunicación eficiente y precisa entre Simulink y ROS2. La Figura 5 ilustra en detalle estas configuraciones, mostrando cómo se han asignado y gestionado los datos de los tópicos para integrarse perfectamente en el entorno de simulación.

En la Figura 6 podemos apreciar la Trayectoria deseada de la ecuación (26), en color rojo la trayectoria en simulación en Matlab y en color verde la trayectoria ROS-Gazebo.

La señal de control obtenida a partir de la simulación se presenta en la Figura 8. Esta señal se encuentra dentro de un rango adecuado para su implementación en un entorno experimental real.

En la Figura 7 se detallan los errores tanto en las posiciones como en la orientación del robot. Estos errores pueden atribuirse a múltiples factores, incluyendo las propiedades físicas asignadas a cada una

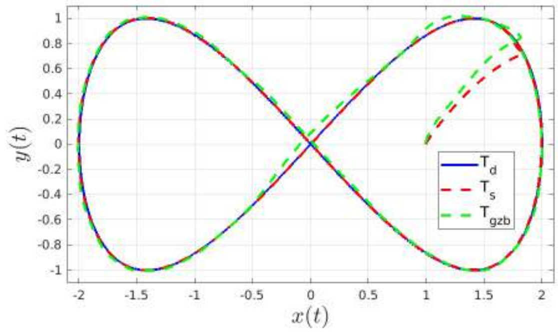

Fig. 6. Seguimiento de la trayectoria, Td=Trayectoria deseada, Ts= trayectoria en simulación y T_{gzb}=Trayectoria ROS-Gazebo

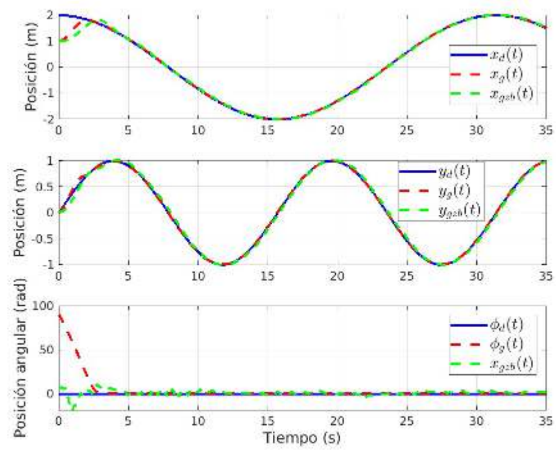

Fig. 7. Variables de posición del RMRM en simulación y en el entorno ROS-Gazebo

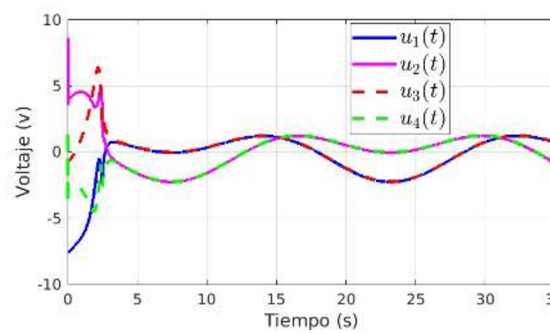

Fig. 8. Entradas de control

de las piezas del robot y las posibles pérdidas durante la conversión de datos a los comandos de velocidad (\cmd_vel). Estos parámetros fueron obtenidos a partir de cálculos precisos para cada componente, lo que asegura una representación fiel del comportamiento dinámico del robot. Sin embargo, la implementación de estos parámetros en Gazebo podría diferir de la forma en que MATLAB maneja la física, lo que explica las variaciones observadas en los resultados. El enfoque en los rodillos del carro mecanum hace que estas discrepancias sean aún más notorias, ya que el movimiento omnidireccional de estas ruedas depende en gran medida de la precisión con la que se modelen las fuerzas laterales y de deslizamiento.

5. CONCLUSIONES

En este artículo se propone un controlador por el método de modos deslizantes para el seguimiento de trayectorias en un robot móvil con ruedas mecanum (RMRM). Se ha derivado una ley de control en la superficie de deslizamiento con ganancias que permiten que el controlador sea robusto y capaz de lidiar con la dinámica no lineal. Además, se asegura una respuesta rápida y con un mínimo error en el sistema, obteniendo resultados simulados dentro de ± 5 mm, La simulación en Gazebo demuestra la viabilidad de implementar este trabajo en un entorno práctico. Las características del motor y los parámetros requeridos del robot móvil, así como los tópicos de ROS, se encuentran correctamente definidos y disponibles para su integración y control en tiempo real.

La simulación, realizada en el entorno Gazebo mediante el acoplamiento de Matlab y ROS, arrojó resultados esperados a pesar de la dinámica no lineal y las inercias generadas por los rodillos acoplados a cada una de las ruedas. Estos resultados pueden mejorarse mediante un análisis más detallado de los parámetros físicos de cada uno de los elementos que componen el modelo, logrando así un error mínimo en el seguimiento de trayectorias.

REFERENCES

Alakshendra, V. and Chiddarwar, S.S. (2017). Adaptive robust control of mecanum-wheeled mobile robot with uncertainties. *Nonlinear Dynamics*, 87, 2147–2169.

Azar, A.T., Sardar, M.Z., Ahmed, S., Hassanien, A.E., and Kamal, N.A. (2023). Autonomous robot navigation and exploration using deep reinforcement learning with gazebo and ros. In *International Conference on Advanced Intelligent Systems and Informatics*, 287–299. Springer.

Can, A., Price, J., and Montazeri, A. (2022). A nonlinear discrete-time sliding mode controller for autonomous navigation of an aerial vehicle using hector slam. *IFAC-PapersOnLine*, 55(10), 2653–2658.

Leong, J.S.L., Teo, K.T.K., and Yoong, H.P. (2022). Four wheeled mobile robots: A review. In *2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, 1–6. IEEE.

Li, Y., Dai, S., Zheng, Y., Tian, F., and Yan, X. (2018). Modeling and kinematics simulation of a mecanum wheel platform in recurdyn. *Journal of Robotics*, 2018.

Lu, X., Zhang, X., Zhang, G., and Jia, S. (2018). Design of adaptive sliding mode controller for four-mecanum wheel mobile robot. In *2018 37th Chinese Control Conference (CCC)*, 3983–3987. IEEE.

Monasterio-Huelin, F. and Gutiérrez, A. (2020). Modelado de un motordc.

Sabouri, M. and Asemani, M.H. (2021). Lpv controller design for trajectory tracking of nonholonomic wheeled mobile robots in the presence of slip. In *2021 29th Iranian Conference on Electrical Engineering (ICEE)*, 715–720. IEEE.

Sun, Z., Hu, S., He, D., Zhu, W., Xie, H., and Zheng, J. (2021). Trajectory-tracking control of mecanum-wheeled omnidirectional mobile robots using adaptive integral terminal sliding mode. *Computers & Electrical Engineering*, 96, 107500.

Wang, D., Gao, Y., Wei, W., Yu, Q., Wei, Y., Li, W., and Fana, Z. (2024). Sliding mode observer-based model predictive tracking control for mecanum-wheeled mobile robot. *ISA Transactions*.

Wang, J., Han, L., Dong, X., Li, Q., and Ren, Z. (2021). Distributed sliding mode control for time-varying formation tracking of multi-uav system with a dynamic leader. *Aerospace Science and Technology*, 111, 106549.

Yadav, P.S., Agrawal, V., Mohanta, J., and Ahmed, M.F. (2022). A robust sliding mode control of mecanum wheel-chair for trajectory tracking. *Materials Today: Proceedings*, 56, 623–630.

Yogi, S.C., Behera, L., and Nahavandi, S. (2023). Adaptive intelligent minimum parameter singularity free sliding mode controller design for quadrotor. *IEEE Transactions on Automation Science and Engineering*, 21(2), 1805–1823.