



DOI: <https://doi.org/10.23857/dc.v10i1.3749>

Ciencias de la Educación
Artículo de Investigación

*Implementación de un modelo matemático de ecuaciones diferenciales para la
simulación del movimiento de las ruedas de un robot con diferentes velocidades
impuestas mediante MATLAB/Simulink*

*Implementation of a mathematical model of differential equations for the
simulation of the movement of the wheels of a robot with different speeds imposed
using MATLAB/Simulink*

*Implementação de um modelo matemático de equações diferenciais para simulação
do movimento das rodas de um robô com diferentes velocidades impostas utilizando
MATLAB/Simulink*

Alex Marcelo Moreta Chango ^I
ammoreta.felc@unach.edu.ec
<https://orcid.org/0009-0008-5760-6245>

Néstor Augusto Estrada Brito ^{II}
nestor.estrada@epoch.edu.ec
<https://orcid.org/0000-0002-4100-7351>

Correspondencia: ammoreta.felc@unach.edu.ec

***Recibido:** 29 de diciembre de 2023 ***Aceptado:** 10 de enero de 2024 * **Publicado:** 26 de febrero de 2024

- I. Licenciado en Ciencias de la Educación, Profesor de Ciencias Exactas, Universidad Nacional De Chimborazo, Riobamba, Ecuador.
- II. Ingeniero, Docente investigador, Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador.

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

Resumen

Este trabajo explora la eficacia de los modelos matemáticos que utilizan ecuaciones diferenciales en MATLAB Simulink y los algoritmos de programación para simular el movimiento de las ruedas en robots. A través de un análisis comparativo, evaluamos las ventajas y limitaciones de cada enfoque en términos de precisión, flexibilidad y eficiencia computacional. El estudio revela que los modelos matemáticos proporcionan una representación precisa del comportamiento del robot basada en principios físicos, lo que permite una simulación precisa del movimiento. Sin embargo, los algoritmos de programación ofrecen una mayor flexibilidad y adaptabilidad, permitiendo la implementación de estrategias de control complejas. En última instancia, la elección entre modelos matemáticos y algoritmos de programación depende de los requisitos del proyecto, las capacidades del sistema y las preferencias del diseñador. Un enfoque combinado puede ofrecer lo mejor de ambos mundos, aprovechando las fortalezas de cada método para lograr simulaciones de robots más precisas y eficientes.

Palabras Claves: Modelos matemáticos; Ecuaciones diferenciales; Simulación de movimiento; MATLAB/Simulink; Control de robots.

Abstract

This work explores the effectiveness of mathematical models using differential equations in MATLAB Simulink and programming algorithms to simulate wheel motion in robots. Through comparative analysis, we evaluate the advantages and limitations of each approach in terms of accuracy, flexibility, and computational efficiency. The study reveals that mathematical models provide an accurate representation of robot behavior based on physical principles, allowing for accurate simulation of movement. However, programming algorithms offer greater flexibility and adaptability, allowing the implementation of complex control strategies. Ultimately, the choice between mathematical models and programming algorithms depends on the project requirements, system capabilities, and designer preferences. A combined approach can offer the best of both worlds, leveraging the strengths of each method to achieve more accurate and efficient robot simulations.

Keywords: mathematical models; Differential equations; Motion simulation; MATLAB/Simulink; Robot control.

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

Resumo

Este trabalho explora a eficácia de modelos matemáticos usando equações diferenciais no MATLAB Simulink e algoritmos de programação para simular o movimento de rodas em robôs. Através de análises comparativas, avaliamos as vantagens e limitações de cada abordagem em termos de precisão, flexibilidade e eficiência computacional. O estudo revela que os modelos matemáticos fornecem uma representação precisa do comportamento do robô com base em princípios físicos, permitindo uma simulação precisa do movimento. Porém, os algoritmos de programação oferecem maior flexibilidade e adaptabilidade, permitindo a implementação de estratégias de controle complexas. Em última análise, a escolha entre modelos matemáticos e algoritmos de programação depende dos requisitos do projeto, das capacidades do sistema e das preferências do designer. Uma abordagem combinada pode oferecer o melhor dos dois mundos, aproveitando os pontos fortes de cada método para obter simulações de robôs mais precisas e eficientes.

Palavras-chave: modelos matemáticos; Equações diferenciais; Simulação de movimento; MATLAB/Simulink; Controle do robô.

Introducción

Para el estudio y análisis de sistemas físicos es necesario la utilización de herramientas como el modelo matemático y la simulación. Estas herramientas permiten representar de manera formal sistemas físicos sobre situaciones reales, para el estudio de su comportamiento y mediante modelos de simulación para obtener unos resultados deseados.

Los sistemas físicos se consideran manipuladores, que necesitan interactuar con entradas y salidas a través de componentes o subsistemas externos. La relación de entrada-salida para componentes y subsistemas son regulables mediante funciones de transferencia, que se definen en términos de las variables del sistema.

Para representar un modelado primero se debe realizar la formulación del modelo matemático, después la verificación del modelo y por último la validación de dicho modelo. Simulink es una herramienta muy útil que nos permite representar un modelado físico y sistemas de control mediante diagramas de bloques y sus resultados son analíticos o aproximados según corresponda el modelo.

El comportamiento de dichos sistemas se define mediante funciones de transferencia, operaciones matemáticas, elementos de Matlab y señales.

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

Simulink dispone de una serie de utilidades que facilitan la visualización, análisis y guardado de los resultados de una simulación.

La robótica, como disciplina en constante evolución, ha experimentado un significativo avance en las últimas décadas, consolidándose como un campo multidisciplinario esencial en diversos sectores de la sociedad moderna. El estudio del movimiento de las ruedas de un robot es fundamental para comprender y optimizar su rendimiento en aplicaciones prácticas. En este sentido, la aplicación de modelos matemáticos basados en ecuaciones diferenciales emerge como una herramienta crucial para simular y analizar con precisión dichos movimientos.

Los trabajos de Craig (2005) y Siciliano y Khatib (2008) resaltan la importancia de abordar la mecánica y el control en la robótica, destacando la necesidad de implementar modelos matemáticos precisos para la simulación de movimientos. Spong, Hutchinson y Vidyasagar (2005) complementan esta perspectiva al subrayar la relevancia de los modelos basados en ecuaciones diferenciales para capturar la dinámica intrínseca de los robots.

En esta investigación, se enfocará en la implementación de un modelo matemático específico para simular el movimiento de las ruedas de un robot en respuesta a diferentes velocidades impuestas. La elección de MATLAB/Simulink como entorno de simulación se justifica a través de la versatilidad de la herramienta, como se destaca en el trabajo de Corke (2017), que explora algoritmos fundamentales en MATLAB para la robótica.

La validación del modelo propuesto será realizada mediante comparaciones con datos experimentales y resultados de la literatura, siguiendo los principios establecidos por Khalil (2002) en el estudio de sistemas no lineales. La relevancia de los modelos matemáticos en la robótica moderna, como señalan Lynch y Park (2017), es crucial para avanzar en el diseño y control de robots en entornos dinámicos y desafiantes.

La estructura de este trabajo se basa en la revisión de la literatura, donde se abordarán los conceptos teóricos fundamentales según los trabajos de Ogata (2010), Franklin, Powell y Emami-Naeini (2015), y Åström y Murray (2010). La metodología de desarrollo e implementación del modelo en MATLAB/Simulink será guiada por las pautas establecidas por MathWorks (2024) y Simulink (2024), asegurando la precisión y la validez de los resultados.

Bajo esta perspectiva, este documento presenta el desarrollo e implementación de un modelo matemático basado en ecuaciones diferenciales para simular el movimiento de las ruedas de un robot con diferentes velocidades utilizando MATLAB/Simulink. Los objetivos específicos establecidos al

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

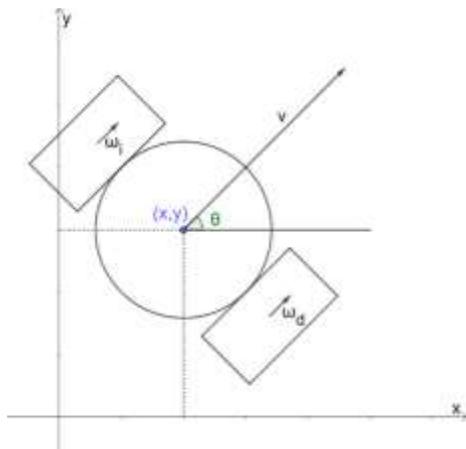
inicio del estudio, que incluyen la creación de un modelo matemático con ecuaciones diferenciales, la realización de un diagrama de bloques para su simulación en Simulink y la comparación del movimiento de las ruedas de un robot que utiliza modelos matemáticos con ecuaciones diferenciales frente a un robot que utiliza algoritmos con ecuaciones trigonométricas, se han abordado satisfactoriamente. Los resultados obtenidos demuestran la eficacia del modelo propuesto en la representación precisa del movimiento de las ruedas del robot, ofreciendo una herramienta valiosa para el diseño y análisis de sistemas robóticos. La implementación de este modelo contribuye al avance en el campo de la robótica, proporcionando una base sólida para futuras investigaciones y aplicaciones prácticas en la simulación y control de robots.

Desarrollo

Para la realización de las simulaciones, se utilizó una computadora ASUS con un procesador Intel Core i7 de 13^a generación y 32 GB de RAM

En la figura 1. Se puede observar las variables utilizadas para obtener las ecuaciones correspondientes que describen el movimiento del robot descritas en (Olson, 2010):

Figura 1: Diagrama de variables del robot



$$\begin{cases} \dot{x} = \frac{R}{2} \cos(\theta) \omega_d + \frac{R}{2} \cos(\theta) \omega_i, \\ \dot{y} = \frac{R}{2} \sin(\theta) \omega_d + \frac{R}{2} \sin(\theta) \omega_i, \\ \dot{\theta} = \frac{R}{L} \omega_d - \frac{R}{L} \omega_i. \end{cases}$$

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

Donde (W_d, W_i) son las velocidades angulares de la rueda derecha e izquierda correspondientemente, y (R, L, θ) son el radio, la separación entre las dos ruedas y la orientación, (x, y) indican la posición del robot en coordenadas.

En cambio, para la simulación del movimiento de las ruedas de un robot se utilizaron las siguientes líneas de programación en py.

```
# Parámetros del control proporcional
```

```
Kp_linear = 10.0
```

```
target_x = 10.0
```

```
target_y = 10.0
```

```
current_x = 0.0
```

```
current_y = 0.0
```

```
current_theta = 0.0
```

```
# Datos para la gráfica
```

```
x_positions = [current_x]
```

```
y_positions = [current_y]
```

```
# Simulación del movimiento del robot
```

```
for _ in range(100):
```

```
    # Calcular el error en la posición
```

```
    error_x = target_x - current_x
```

```
    error_y = target_y - current_y
```

```
    # Calcular el ángulo hacia el objetivo
```

```
    target_theta = atan2(error_y, error_x)
```

```
    # Calcular el error en el ángulo
```

```
    error_theta = target_theta - current_theta
```

```
    # Calcular las velocidades de las ruedas
```

```
    linear_velocity = Kp_linear * sqrt(error_x**2 + error_y**2)
```

```
    # Simular el movimiento del robot
```

```
    current_x += linear_velocity * cos(current_theta)
```

```
    current_y += linear_velocity * sin(current_theta)
```

```
    current_theta += 0.1 # Actualizar el ángulo de manera arbitraria para simular movimiento
```

```
    # Almacenar datos para la gráfica
```

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

```
x_positions.append(current_x)
y_positions.append(current_y)
# Graficar el movimiento lineal
plt.plot(x_positions, y_positions, label='Movimiento Lineal')
plt.scatter(target_x, target_y, color='red', marker='x', label='Objetivo')
plt.title('Movimiento Lineal del Robot')
plt.xlabel('Posición X')
plt.ylabel('Posición Y')
plt.legend()
```

Modelado construido en Simulink

Primero se partió de la ecuación diferencial de θ' , se utilizó un “add”, “gain” y un “product” de simulink para obtener la resta de las velocidades de las ruedas $(W_d - W_i) * (R/L)$, a esto se unió un “integrator” para obtener el valor de θ .

La ecuación diferencial de y' , sería: $(W_d + W_i) * \text{sen}(\theta) * (R/2)$, para lo cual se utilizó un “add”, “product”, “sen”, “gain” para obtener dicha ecuación, luego se adjuntó un “integrator” para obtener la variable y

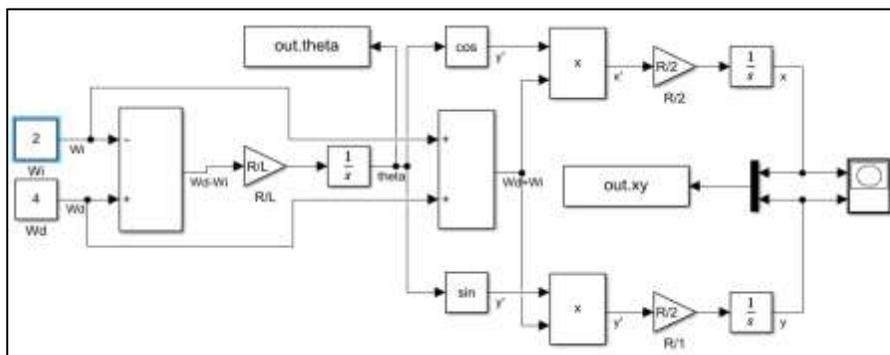
La ecuación diferencial de x' , sería $(W_d + W_i) * \text{cos}(\theta) * (R/2)$, para lo cual se utilizó un “add”, “product”, “cos”, “gain” para obtener dicha ecuación, luego se unió a un “integrator” para obtener x .

Una vez obtenido los variables (x, y, θ) , se utilizó un “xy graph” para conseguir una representación gráfica del movimiento de las ruedas, y “mux”, “output” para conseguir los resultados en forma de vectores.

En la Figura 2 se describe el modelado del movimiento de un robot en simulink mediante diagramas de bloques que representan las ecuaciones mencionadas con anterioridad.

Figura 2 Diagrama de bloques del sistema

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink



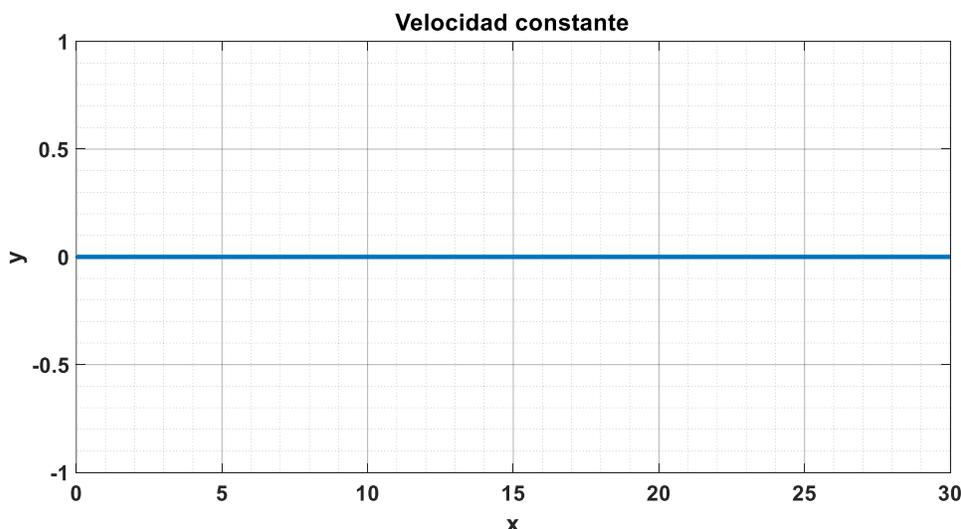
Resultados

Los valores de: $R = 1$, $L = 3$ fueron utilizados para todas las simulaciones. Los resultados representados en las tablas son obtenidos de la simulación desde el primer valor con saltos de 5 puntos hasta el valor final, con la finalidad de visualizar de mejor manera su respuesta.

Velocidad angular constante para ambas ruedas

Se utilizó un valor de 2 tanto para la velocidad W_d como para W_i y se pudo observar que, al utilizar una velocidad constante en la simulación, la gráfica obtenida del movimiento es una línea recta como se indica en la Figura 3 cuyos valores obtenidos están representados en la Tabla 1. El valor del ángulo es cero por lo que se restan las velocidades iguales en la ecuación diferencial de θ' . El tiempo de ejecución fue de 0.000002354s.

Figura 3: Movimiento del robot con una velocidad constante para ambas ruedas mediante Ecuaciones Diferenciales



Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las
ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

Tabla 1: Valores de x, y con velocidades

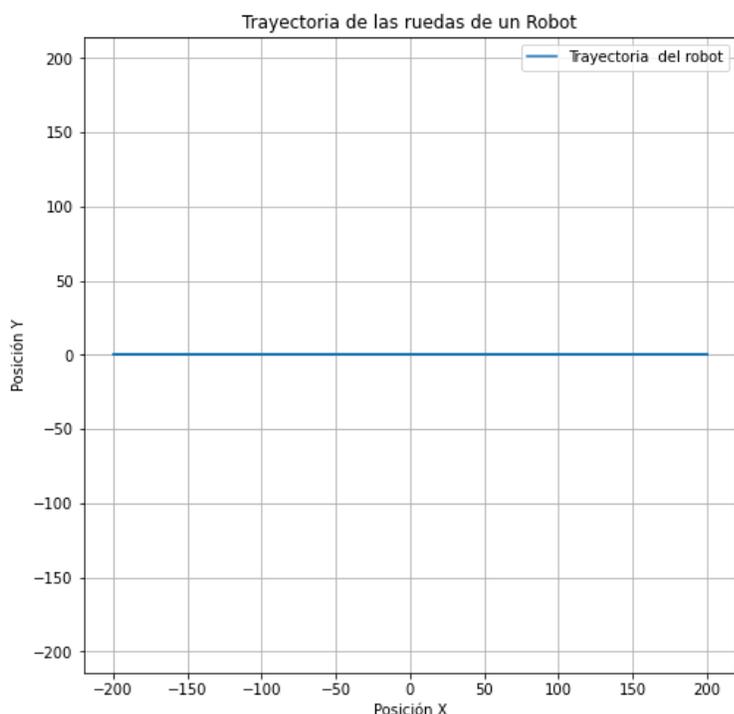
x	y
0	0
0.1569	0
3.1569	0
6.1569	0
9.1569	0
12.1569	0
15.1569	0
18.1569	0
21.1569	0
24.1569	0
27.1569	0
30.0000	0

constantes

En la Figura 4. Se puede observar el movimiento de las ruedas de un robot con velocidades constantes, mediante el uso de algoritmos. El tiempo de ejecución fue de 0.00005467s.

Figura 4: Movimiento del robot con una velocidad constante para ambas ruedas mediante algoritmos

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink



Velocidad constante de diferente magnitud para cada rueda

La Figura 4 describe el movimiento del robot cuando se utiliza velocidades de $W_d = 2$, $W_i = 4$, se puede ver que recorren una misma figura cuando sus velocidades son diferentes pero constantes. Si la velocidad de la rueda izquierda es mayor que la derecha, se obtiene valores negativos de "y", θ , la Figura 5 describe el comportamiento del ángulo y sus valores están descritos en las Tablas 2, 3. El tiempo de ejecución fue de 0.00000457

Figura 4: Movimiento del robot con velocidades constantes $W_i > W_d$

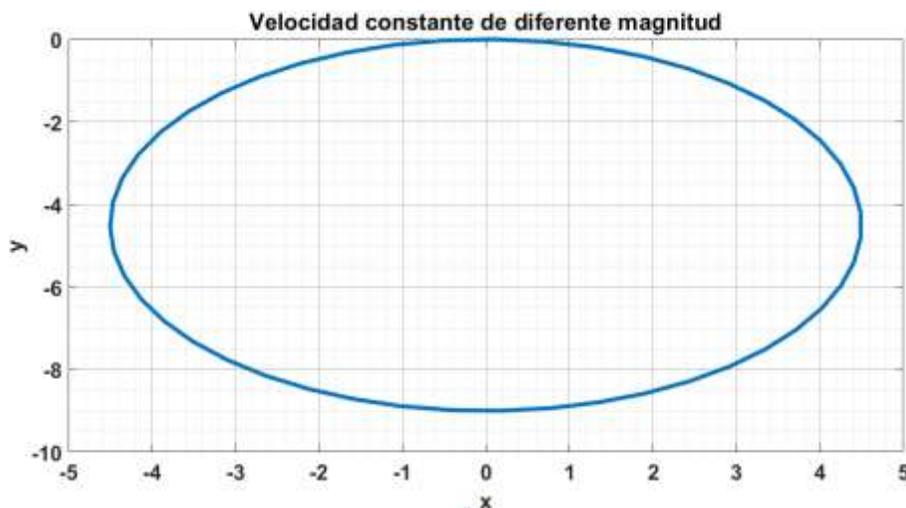


Figura 5: Comportamiento del ángulo con velocidades constantes $W_i > W_d$

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

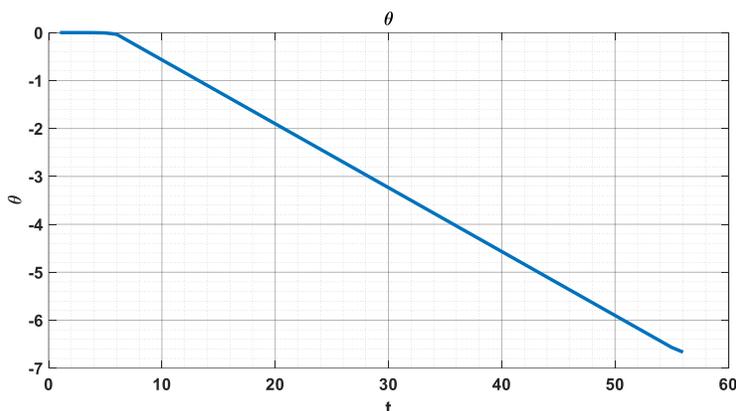


Tabla 2: Valores de x, y con $W_i > W_d$

x	y
0	0
0.15691	-0.0027365
2.9043	-1.0627
4.408	-3.5946
4.0241	-6.5142
1.9169	-8.5713
-1.0111	-8.8849
-3.5061	-7.3209
-4.4997	-4.5488
-3.5665	-1.7559
-1.1059	-0.13802
1.6837	-0.32685

Tabla 3: Valores de Θ con $W_i > W_d$

t	θ
0	0
0.0523	-0.0349
1.0523	-0.7015
2.0523	-1.3682
3.0523	-2.0349
4.0523	-2.7015
5.0523	-3.3682
6.0523	-4.0349
7.0523	-4.7015
8.0523	-5.3682
9.0523	-6.0349
10.0000	-6.6667

La Figura 7 describe el movimiento del robot cuando se utiliza velocidades de $W_d = 4$, $W_i = 2$, se puede ver que recorren una misma figura cuando sus velocidades son diferentes pero constantes. Si la velocidad de la rueda derecha es mayor que la izquierda, se obtiene valores positivos de “y”, θ , la Figura 7 describe el comportamiento del ángulo y sus valores están descritos en las Tablas 4, 5.

Figura 7: Movimiento del robot con velocidades constantes $W_d > W_i$

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

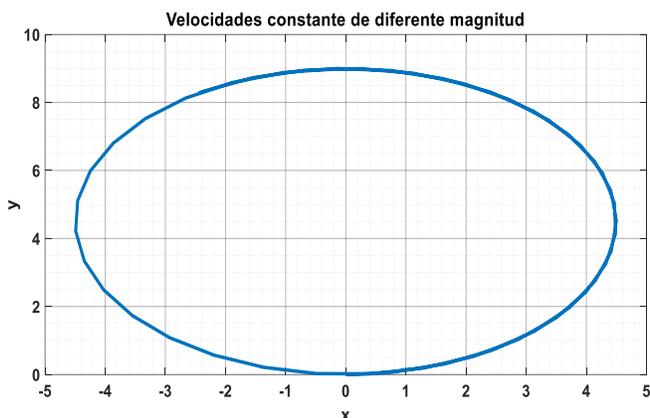


Figura 6: Comportamiento del ángulo con velocidades constantes $W_d > W_i$

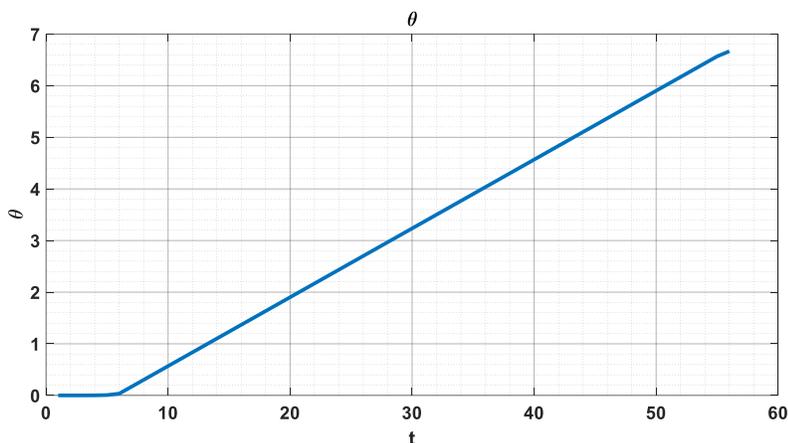


Tabla 4: Valores de Θ con $W_d > W_i$

x	y
0	0
0.1569	0.0027
3.7232	1.9726
4.1384	6.2674
0.7488	8.9373
-3.3293	7.5275
-4.3464	3.3343
-1.3674	0.2128
2.8687	1.0329
4.4674	5.0407
1.9588	8.5513

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

-2.3507	8.3372
---------	--------

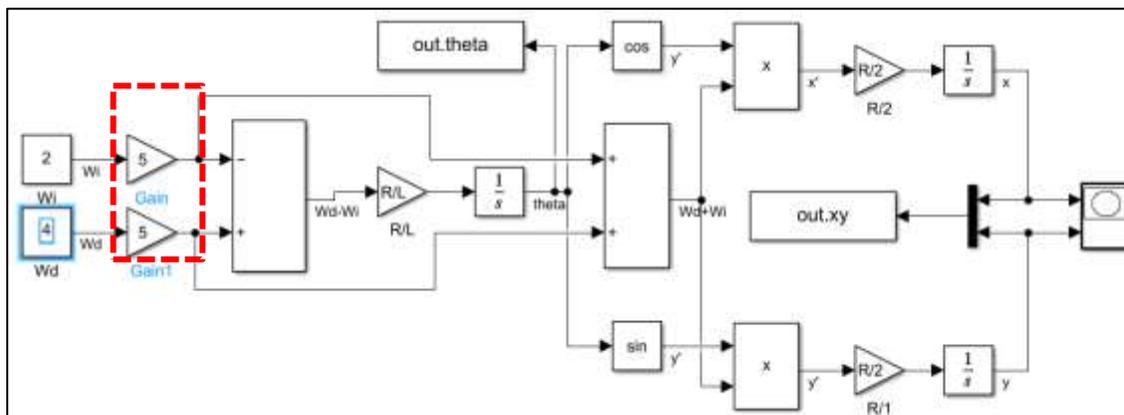
Tabla 4: Valores de x , y con $W_d > W_i$

t	θ
0	0
0.0523	0.0349
1.0523	0.7015
2.0523	1.3682
3.0523	2.0349
4.0523	2.7015
5.0523	3.3682
6.0523	4.0349
7.0523	4.7015
8.0523	5.3682
9.0523	6.0349
10.0000	6.6667

Recorrer la misma figura a mayor velocidad

Para recorrer la misma figura a mayor velocidad se implementó unos valores constantes mediante “gain” de simulink con un valor de 5, y $W_d=4$, $W_i=2$, como se indica en la Figura 9.

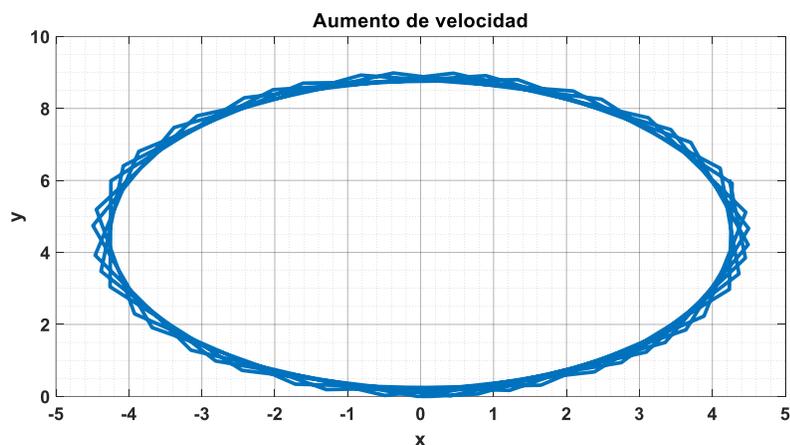
Figura 7: Diagrama de bloques del sistema con aumento de velocidad



Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

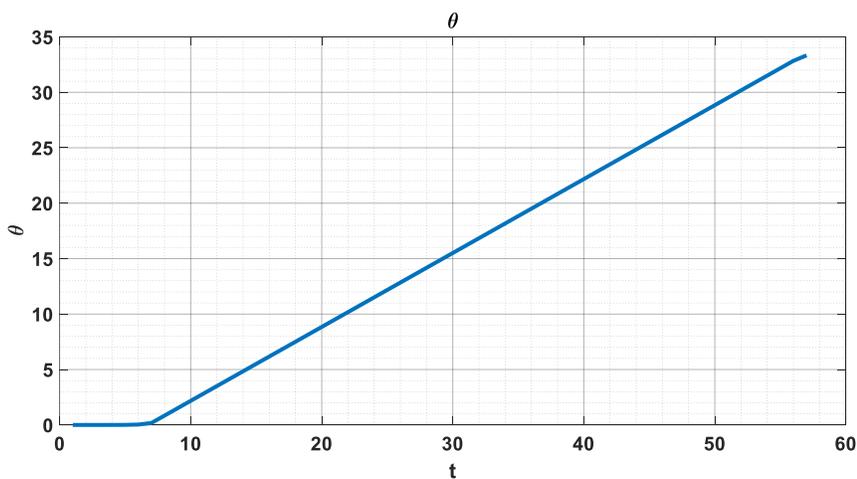
En la Figura 10 se puede observar que el robot a recorrido la misma forma varias veces en el mismo tiempo de simulación que el apartado 4.3, esto es debido a la incorporación de una ganancia constante en ambas velocidades de las ruedas.

Figura 8: Movimiento del robot con un aumento de velocidad



En la Figura 11 se describe el comportamiento del ángulo con el aumento de la velocidad, sus valores de (x, y) y el ángulo representado en vectores están representados en las tablas 5, 6

Figura 9: Comportamiento del ángulo con un aumento de velocidad



Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

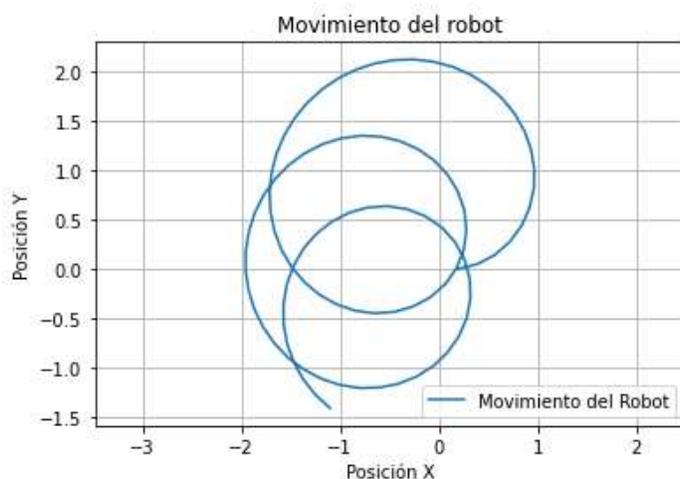
Tabla 5: Valores de x, y con aumento de velocidad Tabla 6: Valores de x, y con aumento de velocidad

x	y
0	0
0.1569	0.0027
1.3320	8.7983
-0.4885	0.0266
-0.3730	8.9845
1.2208	0.1687
-2.0238	8.5192
2.7526	0.9401
-3.3806	7.4701
3.8847	2.2285
-4.2463	5.9895
4.4524	3.8470

t	θ
0	0
0.0523	0.0349
1.0523	2.8411
2.0523	6.1744
3.0523	9.5078
4.0523	12.8411
5.0523	16.1744
6.0523	19.5078
7.0523	22.8411
8.0523	26.1744
9.0523	29.5078
10.0000	32.8411

En la figura 12, se muestra el recorrido que realiza el robot mediante algoritmos a diferentes velocidades cada rueda, pero constantes el cual se puede observar que presento un pequeño fallo y no pudo estar estable. El tiempo de ejecución fue de 0.0002567

Figura 10: Movimiento del robot mediante algoritmos con diferentes velocidades en cada rueda.



Movimiento del robot sobre su propio eje

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

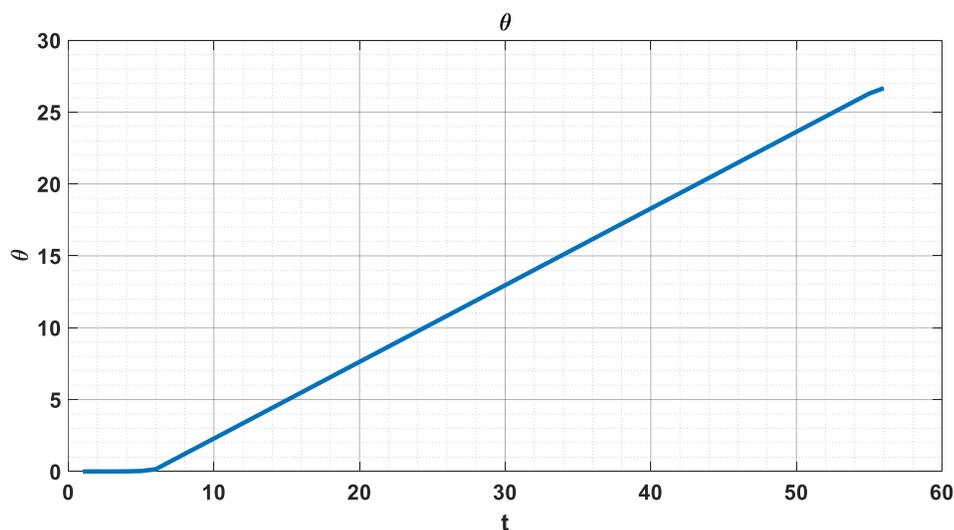
Para que el robot pueda girar sobre su propio eje las velocidades de las ruedas deben ser iguales, pero con signo contrario, para este caso se utilizó velocidades de $W_d = 4$, $W_i = -4$.

La Figura 13 se encuentra vacía ya que el robot no recorre ninguna posición en ninguno de sus ejes (x, y), pero se puede dar cuenta que el robot está en movimiento gracias a la gráfica del ángulo mostrado en la Figura 12, mientras sigue pasando el tiempo de simulación sigue incrementando el ángulo. Sus valores se encuentran representados en la Tabla 8. El tiempo de ejecución fue de 0.00000345.

Figura 13: Movimiento del robot sobre su propio eje



Figura 11: Comportamiento de θ con $W_d=4$, $W_i=-4$



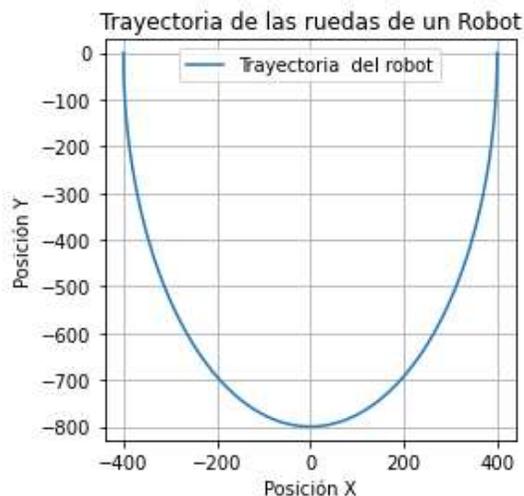
Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

Tabla 7: Valores de Θ cuando el robot se mueve en su propio eje

t	theta
0	0
0.1177	0.1569
1.1177	2.8236
2.1177	5.4903
3.1177	8.1569
4.1177	10.8236
5.1177	13.4903
6.1177	16.1569
7.1177	18.8236
8.1177	21.4903
9.1177	24.1569
10.0000	26.6667

Como se puede observar en la figura 16, el movimiento del robot es diferente usando algoritmos que, al usar un modelo matemático con ecuaciones diferenciales, se puede ver que el robot pierde su estabilidad. El tiempo de ejecución fue de 0.008539

Figura 12: Movimiento de un robot sobre su propio eje

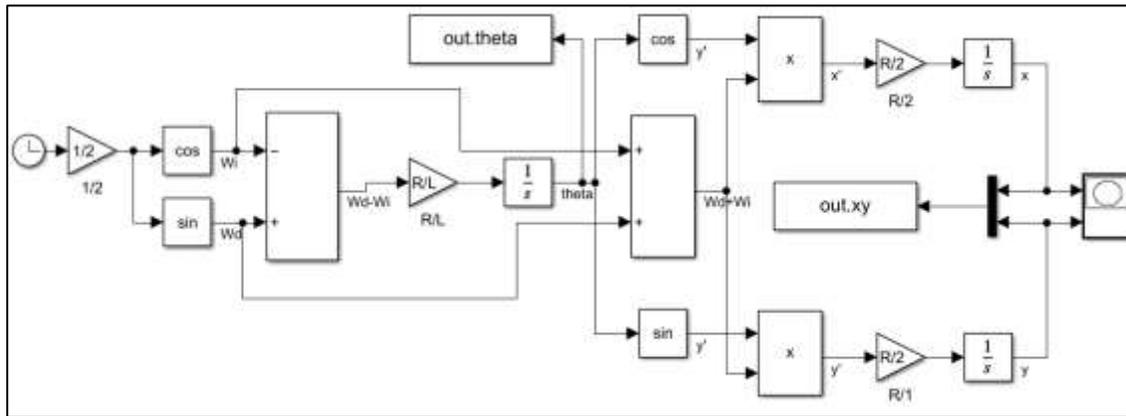


Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

Recorrido del robot con velocidades sinusoidales.

En este apartado se utilizó un stop time de 15 seg y velocidades de: $W_d = \sin\left(\frac{t}{2}\right)$, $W_i = \cos\left(\frac{t}{2}\right)$ como se puede ver en el diagrama de simulink de la Figura 16.

Figura 13: Diagrama de bloques del sistema con velocidades sinusoidales



El recorrido que realiza el robot se aprecia en la Figura 14, el comportamiento del ángulo es parecido a una sinc como se aprecia en la Figura 15, los valores de las coordenadas y el ángulo están representados en las Tablas 9, 10. El tiempo de ejecución fue de 0.089324

Figura 14: Movimiento del robot con velocidades sinusoidales.

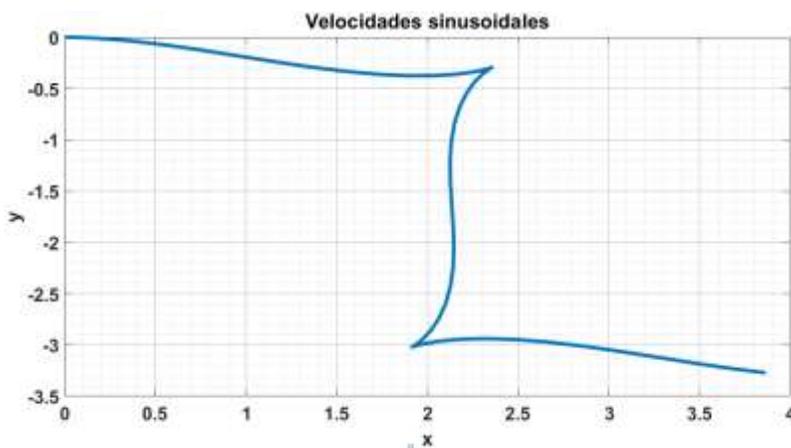


Figura 15: Comportamiento del ángulo con velocidades sinusoidales.

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

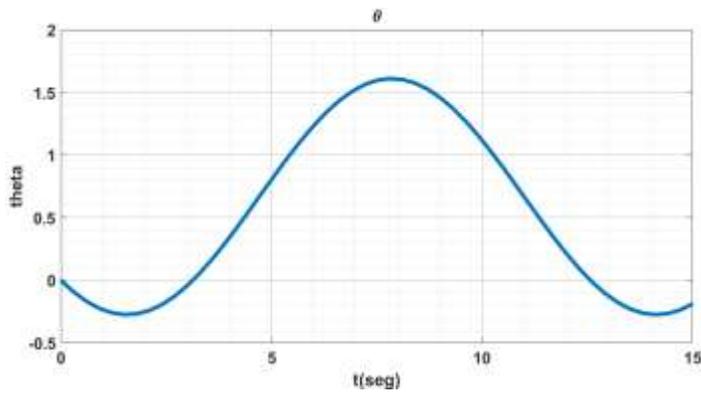


Tabla 8: Valores de (x, y) con velocidad sinusoidal

x	y
0	0
0.1683	-0.0085
1.1439	-0.2349
2.0399	-0.3748
2.3568	-0.2942
2.1786	-0.6777
2.1341	-1.6300
2.1007	-2.5984
1.9167	-3.0197
2.1951	-2.9445
3.0725	-3.0669

Tabla 9: Comportamiento de Θ con velocidad sinusoidal

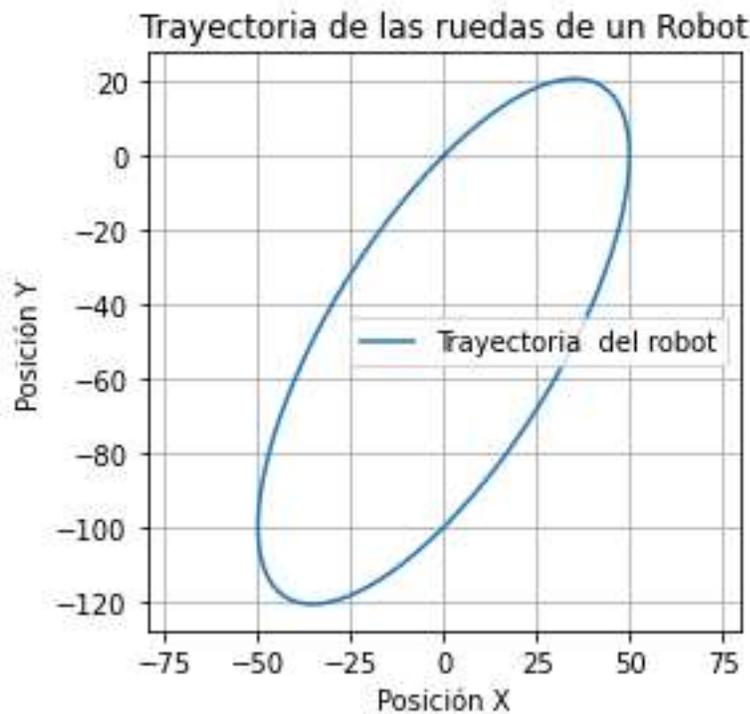
t	y
0	0
0.3139	-0.0960
1.8139	-0.2692
3.3139	0.0598
4.8139	0.7145

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

6.3139	1.3435
7.8139	1.6093
9.3139	1.3693
10.8139	0.7522
12.3139	0.0892
13.8139	-0.2639

En la figura 19 observamos que, al utilizar algoritmos y con velocidades sinusoidales tuvo un mejor funcionamiento que al utilizar el modelo matemático con ecuaciones diferenciales. El tiempo de ejecución fue de 1.0003245

Figura 16: Movimiento con velocidades sinusoidales

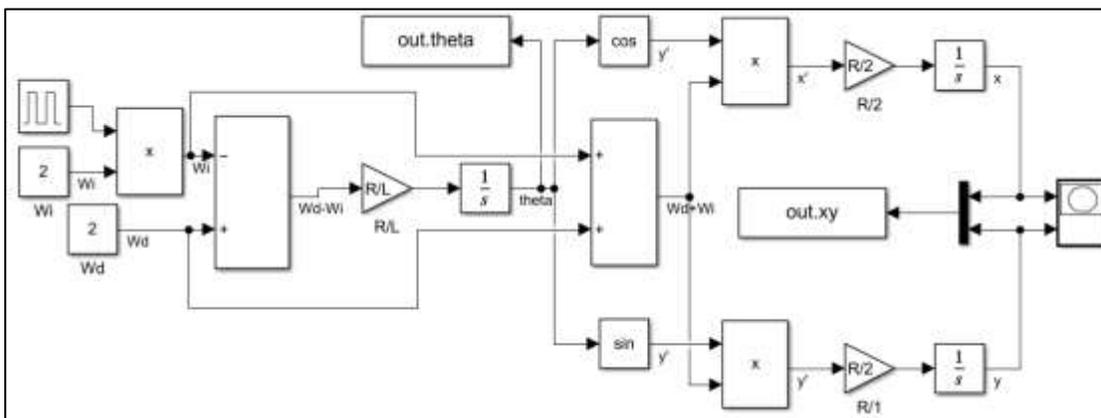


Simulación de un fallo en la rueda izquierda

Para conseguir un fallo a la rueda izquierda se implementó un pulso en el cuál cada dos segundos deja de funcionar 4 décimas de ese tiempo. Para ello se aplica 2 segundos al periodo del pulso y el 80% al porcentaje del periodo, con esto conseguimos que funcione 1.6 seg y deje de funcionar 0.4 seg. La Figura 20 muestra la implementación del fallo.

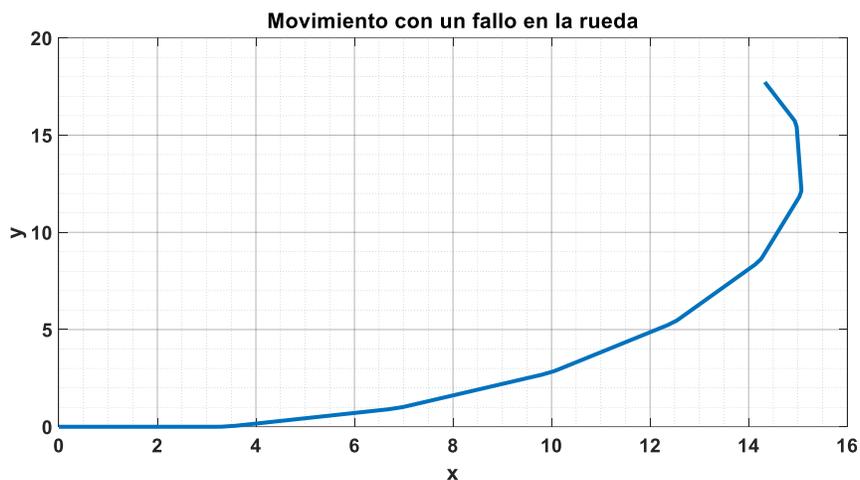
Figura 17: Diagrama de bloques del sistema con un fallo en la rueda izquierda

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink



El movimiento del robot y el comportamiento del ángulo se muestra en las Figura 21, 22 y sus valores están representados en las tablas 11, 12.

Figura 21: Movimiento del robot con un fallo en la rueda izquierda



Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

Figura 18: Comportamiento del ángulo con un fallo en la rueda izquierda

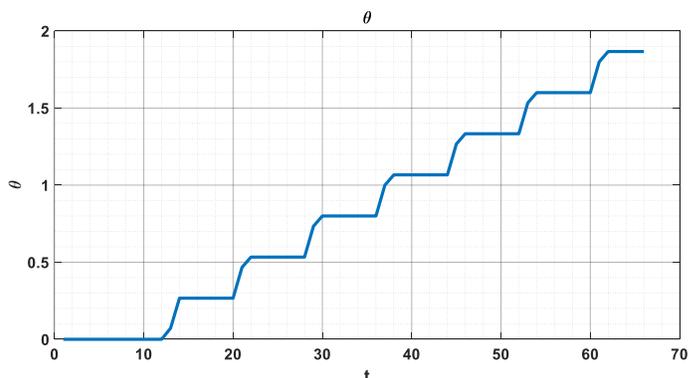


Tabla 10: Comportamiento de Θ con un fallo en W_i

x	y
0	0
0.1569	0
3.1569	0
4.7529	0.3692
6.9618	1.0036
9.1162	2.2718
10.5365	3.3555
12.3480	5.2206
13.4548	7.1172
14.2756	8.7147
14.9813	11.6306
15.0348	13.4211
14.9378	15.7172
14.3287	17.7269

Tabla 11: Valores de (x, y) con un fallo en W_i

t (seg)	θ
0	0

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

0.0785	0
1.5785	0
2.6000	0.2667
3.9000	0.4667
5.2000	0.5333
6.3000	0.8000
7.6000	0.8000
8.9000	1.0667
10.0000	1.3333
11.5000	1.3333
12.6000	1.6000
13.9000	1.8000
15.0000	1.8667

Eficiencia computacional

En la Tabla 13. Se puede observar que, los tiempos de ejecución de la simulación de modelos matemáticos con ecuaciones diferenciales presentan mejores resultados referente al uso de los algoritmos de programación.

Tabla 12: Eficiencia computacional

	Eficiencia computacional	
	Modelo matemático con ecuaciones diferenciales	Algoritmos de programación
Movimiento Lineal	0,000002354	0,00005467
Movimiento elíptico	0,00000457	0,0002567
Movimiento en su propio eje	0,00000345	0,008539
Movimiento con velocidades sinusoidales	0,089324	1,0003245

Desviación causada por el fallo de la rueda

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

Para poder observar la desviación causada por el fallo de la rueda izquierda es necesario realizar una comparación de la Tabla 1 con la Tabla 12 que corresponden con los valores que toman los ángulos, en la Tabla 14 se puede ver la comparación de estos valores con los 15 segundos de simulación.

Tabla 13: Comparativa de la desviación del ángulo

t (seg)	Θ Normal	Θ con Fallo
0 a 1.6	0	0
1.7076	0	0.0718
2 a 3.6	0	0.2667
3.9	0	0.4667
4 a 5.6	0	0.5333
5.9	0	0.7333
6 a 7.6	0	0.8
7.9	0	1
8 a 9.6	0	1.0667
9.9	0	1.2667
10 a 11.6	0	1.3333
11.9	0	1.5333
12 a 13.6	0	1.6
13.9	0	1.8
14 a 15	0	1.8667

Conclusiones

Referente a la desviación que sufre el movimiento del robot debido a una falla en la rueda izquierda se puede concluir que, el ángulo tiene un desvío de 0.2667 en 0.4 segundos de cada dos segundos, el 1.6 segundos restantes de los 2 segundos se mantiene constante el ángulo, como se puede verificar en la Tabla 14.

conclusión, el uso de modelos matemáticos con ecuaciones diferenciales para simular las ruedas de un robot en MATLAB Simulink ofrece una poderosa herramienta para diseñar, analizar y optimizar el comportamiento del robot en diversas situaciones y entornos. Al emplear ecuaciones diferenciales que describen el movimiento y la dinámica de las ruedas, podemos obtener una representación precisa y detallada de cómo el robot se moverá y responderá a diferentes comandos de control.

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

Además, al utilizar MATLAB Simulink, se facilita la implementación de estos modelos, permitiendo una simulación interactiva y visual que ayuda a comprender mejor el comportamiento del robot. Esto facilita la experimentación con diferentes parámetros y estrategias de control, lo que puede llevar a mejoras significativas en el diseño y desempeño del robot en la práctica.

El uso de modelos matemáticos con ecuaciones diferenciales en MATLAB Simulink proporciona una plataforma versátil y poderosa para el desarrollo y análisis de sistemas de control de robots, permitiendo a los ingenieros y diseñadores crear robots más eficientes, precisos y funcionales. En conclusión, al comparar el uso de modelos matemáticos con ecuaciones diferenciales en MATLAB Simulink y algoritmos de programación para simular el movimiento de las ruedas de un robot, se observa que ambos enfoques tienen sus ventajas y limitaciones.

El uso de modelos matemáticos ofrece una representación precisa y detallada del comportamiento del robot, permitiendo una simulación precisa del movimiento basada en principios físicos. Esto facilita la comprensión del sistema y la predicción de su comportamiento en diferentes situaciones.

Por otro lado, los algoritmos de programación ofrecen una mayor flexibilidad y adaptabilidad, lo que permite la implementación de estrategias de control más complejas y específicas para el robot. Además, los algoritmos de programación pueden ser más eficientes en términos de tiempo de cálculo y recursos computacionales en comparación con los modelos matemáticos complejos.

En última instancia, la elección entre modelos matemáticos y algoritmos de programación depende de los requisitos específicos del proyecto, las capacidades del sistema y las preferencias del diseñador. En algunos casos, puede ser beneficioso combinar ambos enfoques para aprovechar sus fortalezas individuales y lograr una simulación más precisa y eficiente del movimiento del robot.

Referencias

- Craig, J. J. (2005). *Introduction to Robotics: Mechanics and Control*. Pearson Prentice Hall.
- Siciliano, B., & Khatib, O. (2008). *Springer Handbook of Robotics*. Springer.
- Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2005). *Robot Modeling and Control*. Wiley.
- Corke, P. (2017). *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer.
- Lynch, K. M., & Park, F. C. (2017). *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press.
- MathWorks. (2024). *MATLAB Documentation*.
- Simulink. (2024). *Simulink Documentation*.

Implementación de un modelo matemático de ecuaciones diferenciales para la simulación del movimiento de las
ruedas de un robot con diferentes velocidades impuestas mediante MATLAB/Simulink

- American Psychological Association. (2020). Publication Manual of the American Psychological Association (7th ed.). American Psychological Association.
- Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing. Pearson.
- Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (2015). Feedback Control of Dynamic Systems. Pearson.
- Åström, K. J., & Murray, R. M. (2010). Feedback Systems: An Introduction for Scientists and Engineers. Princeton University Press.
- Khalil, H. K. (2002). Nonlinear Systems. Pearson.
- Lewis, F. L., Dawson, D. M., & Abdallah, C. T. (2011). Control of Robot Manipulators. Macmillan International Higher Education.
- Dorf, R. C., & Bishop, R. H. (2016). Modern Control Systems. Pearson.
- Ogata, K. (2010). Modern Control Engineering. Pearson.
- Olson, E. S. (2010). Analysis and design of a two-wheeled robot with multiple user interface inputs and vision feedback control. The university of texas at arlington.
- Gil Rodriguez, Manuel, (2003). Introducción a Matlab y Simulink para ciencias e ingeniería.