



UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ

CENTRO DE INVESTIGACIÓN Y ESTUDIOS DE
POSGRADO

FACULTAD DE INGENIERÍA

MODELADO Y CONTROL DE UN ROBOT MÓVIL DE DOS
RUEDAS CON AUTOBALANCE

T E S I S

QUE PARA OBTENER EL GRADO DE
MAESTRO EN INGENIERÍA ELÉCTRICA

OPCIÓN: CIBERNÉTICA, CONTROL Y ROBÓTICA

P R E S E N T A:

ING. ARTURO ANTONIO MUÑOZ FLORES

A S E S O R:

DR. RICARDO ÁLVAREZ SALAS

C O A S E S O R:

DR. JUAN ANTONIO CÁRDENAS GALINDO

SAN LUIS POTOSÍ, S.L.P.

FEBRERO DE 2026



Universidad Autónoma de San Luis Potosí

Facultad de Ingeniería



Centro de Investigación y Estudios de Posgrado

Maestría en Ingeniería Eléctrica

Opción: Cibernética, Control y Robótica

“Modelado y Control de un Robot Móvil de dos Ruedas con Autobalance”

Presenta:

Ing. Arturo Antonio Muñoz Flores

Sinodales:

Dr. Ricardo Álvarez Salas
(Asesor de Tesis)

Dr. Juan Antonio Cárdenas Galindo
(Co-asesor de tesis)

Dr. Mario Arturo González García

Dr. Mauro Eduardo Maya Méndez

Dr. Luis Alberto González Murillo
(Invitado)

Dra. Elvia Ruth Palacios Hernández
(Suplente)

San Luis Potosí, S.L.P.

Febrero de 2026.

20 de noviembre de 2025

ING. ARTURO ANTONIO MUÑOZ FLORES
P R E S E N T E.

En atención a su solicitud de Temario, presentada por los **Dres. Ricardo Álvarez Salas y Juan Antonio Cárdenas Galindo**, Asesor y Coasesor de la Tesis que desarrollará Usted, con el objeto de obtener el Grado de **Maestro en Ingeniería Eléctrica**. Me es grato comunicarle que en la sesión del H. Consejo Técnico Consultivo celebrada el día 20 de noviembre del presente, fue aprobado el Temario propuesto:

TEMARIO:

"Modelado y Control de un Robot Móvil de dos Ruedas con Autobalance"

Introducción.

1. Modelado Dinámico del Robot Móvil de dos Ruedas con Autobalance.
2. Desarrollo de Algoritmos de Control para el Robot Móvil de dos Ruedas con Autobalance.
3. Resultados en Simulación.
4. Descripción de la Plataforma y Pruebas Experimentales.

Conclusiones.

Anexos.

Referencias.

"MODOS ET CUNCTARUM RERUM MENSURAS AUDEBO"

ATENTAMENTE

DR. EMILIO JORGE GONZÁLEZ GALVÁN
DIRECTOR



www.uaslp.mx

Av. Manuel Nava 8
Zona Universitaria • CP 78290
San Luis Potosí, S.L.P.
tel. (444) 826 2330 al39
fax (444) 826 2336

Ccp. Archivo
-bvlg

"1945-2025: 80 años formando profesionales de la Ingeniería en beneficio de la sociedad"



UASLP
Universidad Autónoma
de San Luis Potosí



FACULTAD DE
INGENIERÍA



CENTRO DE
**INVESTIGACIÓN
Y ESTUDIOS
DE POSGRADO**

UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ
FACULTAD DE INGENIERÍA
Área de Investigación y Estudios de Posgrado

D E C L A R A C I Ó N

El presente trabajo que lleva por título:

Modelado y Control de un Robot Móvil de dos Ruedas con Autobalance

se realizó en el periodo abril de 2024 a enero de 2026 bajo la dirección del Dr. Ricardo Álvarez Salas y el Dr. Juan Antonio Cárdenas Galindo.

Originalidad

Por este medio aseguro que he realizado el trabajo reportado, y la escritura de este documento de tesis. para fines académicos sin ayuda indebida de terceros y sin utilizar otros medios más que los indicados.

Las referencias e información tomadas directa o indirectamente de otras fuentes se han definido en el texto como tales y se ha dado el debido crédito a las mismas.

El autor exime a la UASLP de las opiniones vertidas en este trabajo escrito y asume la responsabilidad total del mismo.

Este trabajo no ha sido sometido como tesis o trabajo terminal a ninguna otra institución nacional o internacional en forma parcial o total, exceptuando el caso cuando existe un convenio específico de doble titulación celebrado entre ambas instituciones.

Se autoriza a la UASLP para que divulgue este documento para fines académicos.

El autor del trabajo escrito, Arturo Antonio Muñoz Flores.

ÍNDICE GENERAL

Índice general	VII
Índice de tablas	IX
Índice de figuras	XI
Nomenclatura	XV
Resumen	XXIII
Introducción	1
1 Modelado Dinámico del Robot Móvil de dos Ruedas con Autobalance	15
1.1. Introducción	15
1.2. Modelo mediante el enfoque de Euler Lagrange	16
1.3. Modelo mediante el principio de trabajo virtual	31
1.4. Comparación de técnicas de modelado	40
1.5. Verificación mediante simulación del modelo obtenido	41
1.6. Cinemática de un robot móvil de arquitectura diferencial	42
2 Desarrollo de Algoritmos de Control para el Robot Móvil de dos Ruedas con Autobalance	47
2.1. Introducción	47
2.2. Linealización del modelo del robot móvil	48
2.3. Diseño del controlador por retroalimentación de estados	56
2.4. Implementación del controlador por retroalimentación de estados	63
2.5. Diseño del controlador no lineal para el seguimiento de trayectoria	64
2.6. Implementación del controlador para seguimiento de trayectoria	70
3 Resultados en Simulación	75
3.1. Introducción	75
3.2. Resultados del controlador por retroalimentación de estados	76
3.3. Resultados en simulación para el seguimiento de trayectoria	82
4 Descripción de la Plataforma y Pruebas Experimentales	85
4.1. Introducción	85
4.2. Descripción del robot a utilizar, Elegoo Tumbler	86
4.3. Resultados del control de equilibrio	91

4.4.	Comparación entre controladores para el equilibrio	91
4.5.	Resultados experimentales para el seguimiento de trayectoria	96
4.6.	Comparación de controladores para el equilibrio y seguimiento de trayectoria	98
Conclusiones		111
Anexos		
A	Cálculo de los parámetros mecánicos del robot	113
A.1.	Cálculo de los parámetros mecánicos	113
B	Sistemas no holonómicos	123
B.1.	Sistemas no holonómicos	123
Referencias		133

ÍNDICE DE TABLAS

1.1.	Naturaleza de los enfoques de Euler-Lagrange y Trabajo Virtual.	40
2.1.	Resultados de las pruebas para validar la odometría implementada.	71
2.2.	Resultados para validar el sentido de giro de las ruedas en la odometría.	72
4.1.	Tabla de verdad para el driver de los motores.	88
4.2.	Comparación de los resultados entre el controlador implementado de fábrica, E1, y el controlador por retroalimentación de estados E2.	95
4.3.	Comparación de los tiempos de procesamiento en los controladores E1 y E2 con una frecuencia del oscilador interno de 16 MHz.	98
4.4.	Datos usados para las pruebas	99
4.5.	Comparación de los tiempos de procesamiento en los controladores R1 y R2 con una frecuencia del oscilador interno de 16 MHz.	103
4.6.	Comparación de los controladores para el seguimiento de una trayectoria circular.	105
4.7.	Duración de la batería para cada controlador.	109
A.1.	Valores calculados y simulados de los parámetros mecánicos.	120

ÍNDICE DE FIGURAS

1.	Robot a utilizar en el trabajo de tesis, Robot Elegoo Tumbler.	4
2.	Tipos de modelado para los robots de dos ruedas de autobalance.	5
1.1.	Esquema del péndulo invertido.	16
1.2.	Ángulo de inclinación del péndulo invertido.	20
1.3.	Velocidad angular del péndulo invertido.	20
1.4.	Diagrama de la rueda.	21
1.5.	Distancia recorrida por la rueda.	23
1.6.	Velocidad de la rueda.	23
1.7.	Diagrama del robot de autobalance.	24
1.8.	Esquema de fuerzas del péndulo invertido.	35
1.9.	Resultados de la simulación del robot Elegoo Tumbler: a) Distancia recorrida. b) Velocidad lineal. c) Ángulo de inclinación. d) Velocidad angular.	42
1.10.	Robot móvil de arquitectura diferencial.	43
1.11.	Seguimiento de trayectoria del robot móvil de arquitectura diferencial.	46
1.12.	Velocidades del robot móvil de arquitectura diferencial: a) Velocidad lineal. b) velocidad angular.	46
2.1.	Modelo de un robot móvil tipo diferencial.	64
3.1.	Esquema del sistema en lazo abierto en Simulink.	76
3.2.	Respuestas del sistema en lazo abierto en tiempo continuo: a) Distancia recorrida. b) Velocidad lineal. c) Ángulo de inclinación. d) Velocidad angular.	77
3.3.	Esquema del sistema en lazo cerrado en Simulink	77
3.4.	Respuestas del sistema en lazo cerrado en tiempo continuo: a) Distancia recorrida. b) Velocidad lineal. c) Ángulo de inclinación. d) Velocidad angular	78
3.5.	Ley de control del sistema en lazo cerrado en tiempo continuo.	79
3.6.	Esquema del sistema discretizado en lazo abierto en Simulink.	79
3.7.	Respuestas del sistema en lazo abierto en tiempo discreto: a) Distancia recorrida. b) Velocidad lineal. c) Ángulo de inclinación. d) Velocidad angular.	80
3.8.	Esquema del sistema discretizado en lazo abierto en Simulink.	80
3.9.	Respuestas del sistema en lazo cerrado en tiempo discreto: a) Distancia recorrida. b) Velocidad lineal. c) Ángulo de inclinación. d) Velocidad angular.	81
3.10.	Ley de control del sistema discretizado en lazo cerrado.	81
3.11.	Muestras de la ley de control del sistema discretizado en lazo cerrado.	82
3.12.	Simulación del seguimiento de trayectoria del robot.	82
3.13.	Velocidades en simulación del seguimiento de trayectoria del robot.	83

4.1.	Robot Elegoo Tumbller	87
4.2.	Posiciones del robot Elegoo Tumbller.	89
4.3.	Controles del robot Elegoo Tumbller.	90
4.4.	Conexiones generales del robot Elegoo Tumbller.	91
4.5.	Filtro de Kalman implementado en el robot.	92
4.6.	Robot manteniéndose en equilibrio.	93
4.7.	Control de equilibrio E1.	94
4.8.	Control de equilibrio E2.	94
4.9.	Medidas del ángulo de inclinación del robot con el controlador E1.	95
4.10.	Medidas del ángulo de inclinación del robot con el controlador por E2.	96
4.11.	Medidas de la ley de control del motor derecho con el controlador E1.	96
4.12.	Medidas de la ley de control del motor derecho con el controlador E2.	97
4.13.	Medidas de la ley de control del motor izquierdo con el controlador E1.	97
4.14.	Medidas de la ley de control del motor izquierdo con el controlador E2.	98
4.15.	Tiempo de procesamiento del controlador E1.	99
4.16.	Tiempo de procesamiento del controlador E2.	100
4.17.	Prueba de seguimiento de trayectoria controlador R1.	101
4.18.	Prueba de seguimiento de trayectoria controlador R2.	102
4.19.	Controlador R1.	103
4.20.	Controlador R2.	104
4.21.	Ángulo de inclinación durante el seguimiento de trayectoria del controlador R1.	104
4.22.	Ángulo de inclinación durante el seguimiento de trayectoria del controlador R2.	105
4.23.	Ley de control del motor derecho durante el seguimiento de trayectoria del controlador R1.	105
4.24.	Ley de control del motor derecho durante el seguimiento de trayectoria del controlador R2.	106
4.25.	Ley de control del motor izquierdo durante el seguimiento de trayectoria del controlador R1	106
4.26.	Ley de control del motor izquierdo durante el seguimiento de trayectoria del controlador R2.	107
4.27.	Seguimiento de trayectoria del controlador R1.	107
4.28.	Seguimiento de trayectoria del controlador R2.	108
4.29.	Tiempo de procesamiento del controlador R1.	108
4.30.	Tiempo de procesamiento del controlador R2.	109
A.1.	Robot Elegoo Tumbller	114
A.2.	Prisma rectangular.	114
A.3.	Cilindro.	115
A.4.	Semicilindro.	115
A.5.	Teorema de ejes paralelos.	116
A.6.	Masa del motor.	117
A.7.	Tensor de inercia del motor en SolidWorks.	118
A.8.	Parámetros mecánicos del Robot obtenidos mediante SolidWorks.	119
A.9.	Parámetros mecánicos de las ruedas obtenidos mediante SolidWorks.	120
B.1.	Partícula insertada en una barra cilíndrica.	124
B.2.	Ligaduras por el método de activación.	124
B.3.	Robot holonómico	127

B.4.	Modelo cinemático de un robot móvil tipo automóvil.	129
B.5.	Restricción cinemática de no deslizamiento lateral en un robot móvil tipo automóvil.	129

NOMENCLATURA

Nomenclatura	Descripción
Nomenclatura del Capítulo 1	
$m_{\text{péndulo}}$	Masa concentrada del péndulo invertido.
$l_{\text{péndulo}}$	Distancia del centro de masa del péndulo invertido respecto al marco de referencia.
$\theta_{\text{péndulo}}$	Ángulo de inclinación del péndulo invertido respecto al eje vertical, siendo el sentido positivo el sentido horario.
$I_{\text{péndulo}}$	Momento de inercia del péndulo invertido.
$v_{\text{péndulo}}$	Velocidad lineal del péndulo invertido.
$\omega_{\text{péndulo}}$	Velocidad angular del péndulo invertido.
$x_{\text{péndulo}}$	Coordenada en el eje x del centro de masa del péndulo invertido.
$y_{\text{péndulo}}$	Coordenada en el eje y del centro de masa del péndulo invertido.
g	Aceleración de la gravedad.
$K_{\text{péndulo}}$	Energía cinética del péndulo invertido.
$V_{\text{péndulo}}$	Energía potencial del péndulo invertido.
$L_{\text{péndulo}}$	Lagrangiano del péndulo invertido.
$c_{\omega \text{ péndulo}}$	Coefficiente de fricción del movimiento de rotación del péndulo respecto a su base.
$q_{\text{péndulo}}$	Coordenada generalizada del sistema, correspondiente al ángulo de inclinación del péndulo invertido $\theta_{\text{péndulo}}$.
$\dot{q}_{\text{péndulo}}$	Primera derivada de la coordenada generalizada del sistema, correspondiente a la velocidad angular $\omega_{\text{péndulo}}$.
$\tau_{\text{péndulo}}$	Entrada de control del sistema.
$x_{1 \text{ péndulo}}$	Estado uno del sistema, correspondiente al ángulo de inclinación del péndulo invertido $\theta_{\text{péndulo}}$.
$x_{2 \text{ péndulo}}$	Estado dos del sistema, correspondiente a la velocidad angular del péndulo invertido.
$\dot{x}_{1 \text{ péndulo}}$	Primera derivada del estado uno del sistema.
$\dot{x}_{2 \text{ péndulo}}$	Primera derivada del estado dos del sistema.
R_{rueda}	Radio de la rueda.
F_{rueda}	Fuerza aplicada a la rueda.
m_{rueda}	Masa concentrada de la rueda.

Nomenclatura	Descripción
v_{rueda}	Velocidad lineal de la rueda.
I_{rueda}	Momento de inercia de la rueda respecto a su centro de masa.
ω_{rueda}	Velocidad angular de la rueda.
x_{rueda}	Distancia recorrida por la rueda respecto al marco de referencia.
K_{rueda}	Energía cinética de la rueda.
V_{rueda}	Energía potencial de la rueda respecto al marco de referencia.
L_{rueda}	Lagrangiano de la rueda.
q_{rueda}	Coordenada generalizada del sistema, correspondiente a la distancia recorrida por la rueda x_{rueda} .
$\dot{q}_{\text{rueda}}$	Primera derivada de la coordenada generalizada del sistema, correspondiente a la velocidad de la rueda v_{rueda} .
$x_{1 \text{ rueda}}$	Estado uno del sistema, correspondiente a la distancia recorrida por la rueda x_{rueda} .
$x_{2 \text{ rueda}}$	Estado dos del sistema, correspondiente a la velocidad de la rueda v_{rueda} .
$\dot{x}_{1 \text{ rueda}}$	Primera derivada del estado uno del sistema.
$\dot{x}_{2 \text{ rueda}}$	Primera derivada del estado dos del sistema.
R	Radio de las ruedas del robot.
m_r	Masa concentrada del conjunto de ruedas del robot.
m_h	Masa concentrada del cuerpo del robot.
l	Distancia del eje de las ruedas al centro de masa del cuerpo del robot m_h .
θ	Ángulo de inclinación del robot respecto al eje vertical, considerando positivo el sentido horario.
d_r	Distancia del centro de las ruedas del robot respecto al marco de referencia.
v	Velocidad lineal de las ruedas del robot.
$\mathbf{r}_H$	Vector de posición del centro de masa del cuerpo del robot respecto al marco de referencia.
$\mathbf{r}_L$	Vector de posición del centro de las ruedas del robot respecto al marco de referencia.
I_r	Momento de inercia del conjunto de ruedas del robot respecto a su eje de rotación.
I_p	Momento de inercia del cuerpo del robot respecto al eje de rotación de las ruedas.
ω_r	Velocidad angular de las ruedas del robot.
ω_p	Velocidad angular del cuerpo del robot.
K_r	Energía cinética del conjunto de ruedas del robot.
K_p	Energía cinética del cuerpo del robot.
V_r	Energía potencial del conjunto de ruedas del robot.
V_p	Energía potencial del cuerpo del robot.

Nomenclatura	Descripción
L_p	Lagrangiano del robot.
$\mathcal{R}_p$	Función de disipación de Rayleigh, asociada a las pérdidas de energía por fricción.
c_v	Coefficiente de fricción de las ruedas respecto al suelo sobre el cual se desplaza el robot.
c_ω	Coefficiente de fricción del movimiento de rotación del cuerpo del robot respecto a la unión con las ruedas.
τ_1	Entrada de la ecuación de la sumatoria de fuerzas, correspondiente a la fuerza τ/R .
τ	Par total generado por los motores del robot.
τ_2	Entrada de la ecuación de la sumatoria de pares, considerada nula, es decir, $\tau_2 = 0$.
q_1	Primera coordenada generalizada del sistema, correspondiente a la distancia recorrida por las ruedas d_r .
q_2	Segunda coordenada generalizada del sistema, correspondiente al ángulo de inclinación del robot θ .
$\dot{q}_1$	Primera derivada de la primera coordenada generalizada, correspondiente a la velocidad de las ruedas del robot v .
$\dot{q}_2$	Primera derivada de la segunda coordenada generalizada, correspondiente a la velocidad angular del robot ω_p .
x_1	Estado uno del sistema, correspondiente a la distancia recorrida por las ruedas del robot d_r .
x_2	Estado dos del sistema, correspondiente a la velocidad de las ruedas del robot v .
x_3	Estado tres del sistema, correspondiente al ángulo de inclinación del robot θ .
x_4	Estado cuatro del sistema, correspondiente a la velocidad angular del robot ω_p .
$\dot{x}_1$	Primera derivada del estado uno del sistema.
$\dot{x}_2$	Primera derivada del estado dos del sistema.
$\dot{x}_3$	Primera derivada del estado tres del sistema.
$\dot{x}_4$	Primera derivada del estado cuatro del sistema.
f_i	Fuerza aplicada en un desplazamiento virtual.
δr_i	Desplazamiento virtual, infinitesimal y reversible.
τ_i	Vector de fuerzas o pares generalizados asociados a las coordenadas generalizadas del robot. Representa los esfuerzos equivalentes en el espacio articular que producen el mismo trabajo virtual que las fuerzas aplicadas en el espacio cartesiano.
F_i	Wrench (vector de fuerzas generalizadas en el espacio cartesiano) aplicado al cuerpo i .

Nomenclatura	Descripción
$\mathbf{J}_i$	Jacobiano geométrico del cuerpo i , que relaciona las velocidades generalizadas del sistema con la velocidad lineal y angular del cuerpo.
m_i	Masa del cuerpo i .
$\mathbf{r}_{gi}$	Vector de posición del centro de masa del cuerpo i .
$\ddot{\mathbf{r}}_{gi}$	Vector de aceleración lineal del centro de masa del cuerpo i .
$\mathbf{I}_i$	Tensor de inercia del cuerpo i .
$\boldsymbol{\omega}_i$	Vector de velocidad angular del cuerpo i .
$\dot{\boldsymbol{\omega}}_i$	Vector de aceleración angular del cuerpo i .
$\mathbf{J}_v^T$	Jacobiano lineal del cuerpo i .
$\mathbf{J}_\omega^T$	Jacobiano angular del cuerpo i .
τ_{rot}	Momento de inercia del cuerpo i .
$\mathbf{q}$	Vector de coordenadas generalizadas.
$\dot{\mathbf{q}}$	Vector de la primera derivada de las coordenadas generalizadas.
$\ddot{\mathbf{q}}$	Vector de la segunda derivada de las coordenadas generalizadas.
$\dot{v}$	Aceleración de las ruedas del robot.
$\dot{\omega}_p$	Aceleración angular de la inclinación del robot.
$\mathbf{J}_{vL}$	Jacobiano lineal que corresponde a las ruedas del robot.
$\mathbf{J}_{vH}$	Jacobiano lineal que corresponde al cuerpo del robot.
$\mathbf{a}_L$	Vector de las aceleraciones de las ruedas del robot.
$\mathbf{a}_H$	Vector de las aceleraciones del cuerpo del robot.
ω_L	Velocidad angular de las ruedas.
ω_H	Velocidad angular del cuerpo del robot.
α_L	Es la aceleración angular de las ruedas del robot.
α_H	Es la aceleración angular del cuerpo del robot.
$\mathbf{f}_L$	Vector de fuerzas inerciales en las ruedas del robot.
$\mathbf{f}_H$	Vector de fuerzas inerciales en el cuerpo del robot.
τ_L	Par inercial en las ruedas del robot.
τ_H	Par inercial del cuerpo del robot.
$\mathbf{J}_{\omega L}$	Jacobiano angular de las ruedas del robot.
$\mathbf{J}_{\omega H}$	Jacobiano angular del cuerpo del robot.
$\mathbf{J}_L$	Jacobiano, considerando la parte lineal y angular, de las ruedas del robot.
$\mathbf{J}_H$	Jacobiano, considerando la parte lineal y angular, del cuerpo del robot.
$\mathbf{W}_L$	Wrench de las ruedas del robot.
$\mathbf{W}_H$	Wrench del cuerpo del robot.
τ_{inercial}	Vector de fuerzas o pares inerciales del robot.
f_{gH}	Fuerza gravitacional que actúa en el cuerpo del robot.
f_v	Fuerza debida a la fricción presente en el robot.
f_ω	Par debido a la fricción presente en el robot.
$\tau_{\text{disipativa}}$	Vector de fuerzas y pares debido a la fricción en el robot.

Nomenclatura	Descripción
ρ_{rob}	Radio de rotación del centro de rotación CR al centro de masa del robot.
s_{rob}	Distancia del centro de masa del robot al centro de la rueda.
$\rho_{i rob}$	Radio de rotación del centro de rotación CR a la rueda izquierda.
$\rho_{d rob}$	Radio de rotación del centro de rotación CR a la rueda derecha.
$v_{i rob}$	Velocidad lineal de la rueda izquierda.
$v_{d rob}$	Velocidad lineal de la rueda derecha.
ω_{rob}	Velocidad angular del robot, considerando la rotación sobre su propio eje.
v_{rob}	Velocidad lineal del robot.
$\omega_{i rob}$	Velocidad angular de la rueda izquierda.
$\omega_{d rob}$	Velocidad angular de la rueda derecha.
x_{rob}	Es la coordenada en el eje x del centro de masa del robot.
y_{rob}	Es la coordenada en el eje y del centro de masa del robot.
$\dot{x}_{m rob}$	Velocidad en el eje x respecto al marco de referencia móvil.
$\dot{y}_{m rob}$	Velocidad en el eje y respecto al marco de referencia móvil.
$\dot{x}_{rob}$	Velocidad en el eje x respecto al marco de referencia fijo.
$\dot{y}_{rob}$	Velocidad en el eje y respecto al marco de referencia fijo.
Nomenclatura adicional del Capítulo 2	
$\mathbf{A}_{cont}$	Matriz jacobiana evaluada en el punto de equilibrio y con los valores obtenidos en 1.92.
$\mathbf{B}_{cont}$	Matriz jacobiana correspondiente a la entrada, evaluada en el punto de equilibrio y con los valores obtenidos en 1.92.
$\mathbf{x}_{pe}$	Vector del punto de equilibrio utilizado en el proceso de linealización.
$\mathbf{MC}_{cont}$	Matriz de controlabilidad del sistema en tiempo continuo.
$\bar{\mathbf{C}}_{cont}, \mathbf{P}_{cont}^{-1}$	Matrices o vectores auxiliares utilizados durante el procedimiento de Ackermann.
$\mathbf{k}_{cont}$	Vector de retroalimentación en tiempo continuo.
τ_{cont}	Entrada de control diseñada para el sistema en tiempo continuo, siendo el par generado por los motores del robot.
$\mathbf{f}_{cont}$	Fuerza que permite al robot mantenerse en equilibrio.
$\mathbf{A}_{disc}$	Matriz jacobiana evaluada en el punto de equilibrio y discretizada mediante un retenedor de orden cero.
$\mathbf{B}_{disc}$	Matriz jacobiana correspondiente a la entrada, evaluada en el punto de equilibrio y discretizada mediante un retenedor de orden cero.
$\mathbf{MC}_{disc}$	Matriz de controlabilidad del sistema en tiempo discreto.
$\bar{\mathbf{C}}_{disc}, \mathbf{P}_{disc}$	Matrices o vectores auxiliares utilizados durante el procedimiento de Ackermann para el sistema en tiempo discreto.
$\mathbf{k}_{disc}$	Vector de retroalimentación en tiempo discreto.
τ_{disc}	Entrada de control diseñada para el sistema en tiempo discreto, siendo el par generado por los motores del robot.

Nomenclatura	Descripción
$\mathbf{f}_{disc}$	Fuerza que permite al robot mantenerse en equilibrio y que corresponde a la ley de control implementada en el microcontrolador del robot.
x_{rob}	Coordenada en el eje x del centro de masa del robot.
y_{rob}	Coordenada en el eje y del centro de masa del robot.
$\dot{x}_{rob}$	Velocidad en el eje x del centro de masa del robot.
$\dot{y}_{rob}$	Velocidad en el eje y del centro de masa del robot.
ϕ	Ángulo de rotación del robot sobre el eje perpendicular al plano $x-y$.
a_{rob}	Punto que indica una distancia en adelante respecto al centro de masa del robot. Es usado para el diseño del controlador de seguimiento de trayectoria.
$\mathbf{h}_{rob}$	Vector de las coordenadas del punto a_{rob} , medidas mediante la odometría.
$h_{x\ rob}$	Coordenada en el eje x del punto a_{rob} .
$h_{y\ rob}$	Coordenada en el eje y del punto a_{rob} .
$\dot{\mathbf{h}}_{rob}$	Vector de las velocidades medidas del punto a_{rob} .
$\dot{h}_{x\ rob}$	Velocidad en el eje x del punto a_{rob} .
$\dot{h}_{y\ rob}$	Velocidad en el eje y del punto a_{rob} .
$\mathbf{J}_{rob}$	Matriz jacobiana del robot de arquitectura diferencial en función del ángulo de rotación ϕ del robot y de la distancia a_{rob} .
$\mathbf{h}_{d\ rob}$	Vector de las coordenadas deseadas del punto a_{rob} , correspondientes a la trayectoria de referencia a seguir.
$h_{d\ x\ rob}$	Coordenada en x del vector $\mathbf{h}_{d\ rob}$.
$h_{d\ y\ rob}$	Coordenada en y del vector $\mathbf{h}_{d\ rob}$.
t_{circ}	Variable de tiempo utilizada para generar los valores de la trayectoria de referencia $\mathbf{h}_{d\ rob}$.
$\dot{\mathbf{h}}_{d\ rob}$	Vector de las velocidades de referencia del punto a_{rob} , correspondientes a la trayectoria a seguir.
$\dot{h}_{d\ x\ rob}$	Velocidad en el eje x del vector $\dot{\mathbf{h}}_{d\ rob}$.
$\dot{h}_{d\ y\ rob}$	Velocidad en el eje y del vector $\dot{\mathbf{h}}_{d\ rob}$.
$\mathbf{e}_h$	Error entre las coordenadas seguidas por el robot $\mathbf{h}_{rob}$ y las coordenadas de referencia $\mathbf{h}_{d\ rob}$.
V_{rob}	Función candidata de Lyapunov propuesta para desarrollar el controlador de seguimiento de trayectoria.
$\dot{V}_{rob}$	Derivada de la función candidata de Lyapunov.
$\mathbf{K}_{rob}$	Matriz de las constantes del controlador de seguimiento de trayectoria, construida a partir del vector $\mathbf{k}_{rob}$ y la matriz $\mathbf{I}_{rob}$.
$\mathbf{k}_{rob}$	Vector de las constantes del controlador de seguimiento de trayectoria.
$\mathbf{I}_{rob}$	Matriz identidad.
D_L	Distancia recorrida por la rueda izquierda.

Nomenclatura	Descripción
D_R	Distancia recorrida por la rueda derecha.
N_L	Número de vueltas de la rueda izquierda.
N_R	Número de vueltas de la rueda derecha.
Δ pulsos	Incremento de los pulsos del encoder de cada motor.
PPR	Número de pulsos que hay en una revolución de cada motor.
L_{rob}	Distancia entre las ruedas del robot.
$\Delta\phi$	Incremento del ángulo de rotación del robot.
ΔS_{rob}	Incremento de la distancia recorrida por el robot.
ϕ_k	Valor actual del ángulo de rotación ϕ .
ϕ_{k-1}	Valor anterior del ángulo de rotación ϕ .
$x_{rob\ k-1}$	Coordenada anterior del centro de masa del robot en el eje x .
$y_{rob\ k-1}$	Coordenada anterior del centro de masa del robot en el eje y .
$\omega_{L\ rob}$	Velocidad angular que debe tener la rueda izquierda. Se calcula a partir de las señales de control v_{rob} y ω_{rob} .
$\omega_{R\ rob}$	Velocidad angular que debe tener la rueda derecha. Se calcula a partir de las señales de control v_{rob} y ω_{rob} .
Nomenclatura adicional del Capítulo 4	
PWM	Pulse Width Modulation o Modulación por Ancho de Pulso.
IEC	Integral del Error Cuadrático.
$IVABSC$	Integral del Valor Absoluto de la Señal de Control.
$IVAVSC$	Integral del Valor Absoluto de la Variación de la Señal de Control.

RESUMEN

Este trabajo de tesis está enfocado en estudiar un robot móvil tipo diferencial de dos ruedas y con autobalance. Las tareas principales de este tipo de robots son mantener su equilibrio y moverse en un entorno, de manera similar a otros robots diferenciales. Normalmente la estructura de un robot diferencial consiste en dos ruedas actuadas y una tercera rueda loca, lo que permite mantener el equilibrio sin ningún controlador. Sin embargo, el robot que se utiliza en este trabajo de investigación solo tiene dos ruedas actuadas y no utiliza una tercera rueda de apoyo, lo que provoca que requiera un control para mantenerse en equilibrio. Por lo tanto, en este tipo de robots se pueden evaluar algoritmos de control tanto para el equilibrio como para el seguimiento de trayectorias. Esto requiere la obtención del modelo dinámico y cinemático del robot a utilizar y el diseño de los algoritmos de control. Para el equilibrio se propone un controlador por retroalimentación de estados. Es necesario mencionar que el robot tiene implementado de fábrica un controlador PI-PD. Para el seguimiento de trayectoria se propone un controlador basado en técnicas de Lyapunov usando de referencia una trayectoria circular. Con los controladores diseñados se procedió a realizar pruebas para comparar el desempeño de los controladores. Primero se realizan pruebas únicamente con los controladores del equilibrio y posteriormente se comparan dos casos: el primero considerando la combinación de un controlador PI-PD con el basado en técnicas de Lyapunov y el segundo corresponde al controlador por retroalimentación de estados combinado con el basado en técnicas de Lyapunov. El trabajo realizado dio como resultado un modelo dinámico y cinemático del robot, además de algoritmos de control para el equilibrio y seguimiento de trayectoria de un robot móvil de dos ruedas con autobalance. El contenido de este documento se divide en cuatro capítulos; en el primero se obtiene el modelo dinámico considerando únicamente el equilibrio, así como el modelo cinemático de un robot móvil tipo diferencial. En el segundo capítulo se muestra el proceso para diseñar los controladores de retroalimentación de estados para controlar el equilibrio del robot y el controlador basado en técnicas de Lyapunov, para el seguimiento de trayectoria. El tercer capítulo muestra los resultados de las simulaciones. El cuarto capítulo presenta una descripción del robot utilizado y las

pruebas experimentales realizadas. Finalmente se presentan las conclusiones del trabajo de investigación y los anexos que presentan información necesaria para el desarrollo de la tesis.

INTRODUCCIÓN

El uso de los robots móviles se ha incrementado en los últimos años debido a la versatilidad para realizar diversas tareas, como el transporte de materiales, navegar en espacios con obstáculos y usos sociales como el entretenimiento. Entre los robots de este tipo se encuentran los que usan piernas o ruedas para su locomoción. Sin embargo, los robots con piernas tienen una estructura mecánica muy compleja y esto lleva a que sus algoritmos de control sean igual de complicados. En contraste, los robots de autobalance de dos ruedas tienen un diseño más sencillo y permite controladores menos complejos que los utilizados en los robots con piernas. Es importante remarcar que los robots con tres o más ruedas pueden garantizar su estabilidad estática y esto los convierte en sistemas estables. Por este motivo, su control se basa principalmente en la cinemática y dinámica del movimiento. No obstante, esta misma característica de los convierte en sistemas sobre restringidos y pueden llegar a requerir un sistema de amortiguamiento para casos donde el suelo es irregular. Por el contrario, los robots que cuentan únicamente con dos ruedas, al tener que mantener su propio equilibrio, se clasifican como sistemas inestables, subactuados y no lineales. Estas propiedades los hacen más difíciles de controlar; en particular, la subactuación implica que existen menos entradas de control que grados de libertad del sistema. A pesar de ello, los robots de dos ruedas con autobalance ofrecen ventajas significativas respecto a otros robots con ruedas.

En particular, los robots de dos ruedas con autobalance resultan particularmente útiles en espacios reducidos, ya que el movimiento de rotación puede ser sobre su propio centro de masa debido a un mecanismo diferencial que une las dos ruedas, o bien con un motor en cada rueda. El hecho de que el robot se estabilice continuamente le permite corregir perturbaciones externas, mientras que un robot estáticamente estable podría volcarse ante una perturbación similar. Estos robots son adecuados para desplazarse por pasillos o esquinas estrechas, así como en superficies inclinadas sin perder el equilibrio. Al contar únicamente con dos ruedas, éstas pueden ser de mayor tamaño, lo que facilita su desplazamiento en terrenos irregulares. Este tipo de robots pueden contar con actuadores extras, como brazos manipuladores, sin embargo, esto incrementa la complejidad del modelo dinámico del sistema y, en consecuencia, del control. Para llevar a cabo las acciones de control, es necesario utilizar sensores que midan los parámetros esenciales del robot,

como las velocidades de los motores, el ángulo de inclinación del robot y la velocidad del robot. Estos sensores permiten mantener el equilibrio del robot y controlar su movimiento. No obstante, para lograr una interacción adecuada con el entorno se requiere sensores adicionales, como ultrasónicos o infrarrojos, que permiten detectar obstáculos y percibir características del entorno.

Objetivos y alcance de la investigación

Objetivo de la investigación

Realizar un análisis comparativo de algoritmos de control en función del consumo de energía de la batería.

Objetivos particulares

- Desarrollar el modelo dinámico del robot considerando el movimiento lineal.
- Obtener el modelo cinemático del robot de arquitectura diferencial.
- Calcular los parámetros mecánicos del robot.
- Diseñar un controlador por retroalimentación de estados para controlar el equilibrio del robot.
- Diseñar un controlador no lineal basado en técnicas de Lyapunov para el seguimiento de trayectoria de un robot móvil de arquitectura diferencial.
- Comparar el desempeño del controlador diseñado con respecto al controlador original del robot.
- Combinar el controlador de equilibrio con el de seguimiento de trayectoria.
- Comparar el conjunto de los controladores de equilibrio y de trayectoria.

Hipótesis del trabajo de investigación

La aplicación de técnicas de control moderno permite mejorar el desempeño del robot móvil en tareas de equilibrio con respecto al control originalmente implementado. Además, integrando la odometría es posible incorporar un controlador para seguimiento de trayectorias basado en técnicas de Lyapunov con el controlador propuesto. Finalmente,

mediante una estrategia de conmutación de las acciones de control es posible integrar los dos algoritmos para satisfacer los objetivos de equilibrio, seguimiento de trayectorias y realizar un análisis comparativo en función del consumo de energía de la batería.

Estado del Arte

Modelado y control

La presente tesis se centra en el estudio de un robot de autobalance de dos ruedas cada una equipada con un motor, que se desplaza únicamente en un terreno plano. Este robot es de la marca Elegoo de modelo Tumbler mostrado en la figura (1). En trabajos previos enfocados en el control de este tipo de robots, es indispensable contar con un modelo dinámico adecuado. Dicho modelo puede obtenerse mediante la formulación de Euler-Lagrange o a partir del enfoque de Newton. Estos modelos integran la cinemática con la dinámica, proporcionando una descripción completa del sistema. Pero a su vez el modelo se divide en dos partes principales: la primera corresponde al movimiento de balanceo en relación con el desplazamiento lineal del robot, mientras que la segunda describe el movimiento de rotación. Dependiendo del enfoque, se puede tener un modelo que acople estas dos dinámicas o un modelo desacoplado que solo considere el balanceo y movimiento lineal. Además, se puede trabajar con un modelo no lineal o un modelo linealizado alrededor de un punto de operación. Otra alternativa consiste en obtener el modelo mediante la técnica de identificación de caja negra, aunque la mayoría de los trabajos emplean modelos obtenidos por ecuaciones diferenciales. En el estudio [1] se muestran modelados que se pueden resumir en la figura (2).

Para el control se tiene una amplia gama de controladores que se pueden usar. El más básico es el uso de controladores PI, PD y PID. Sin embargo, es más común usar controladores lineales y no lineales aplicado a un modelo en espacio de estados. El propio robot Elegoo Tumbler contiene un controlador PD para mantener el balance del cuerpo del robot, un PI para controlar la velocidad, un PD para el control de equilibrio, finalmente todas las salidas de estos lazos entran a un controlador PID. Un estudio que también usa un controlador PID para el balance se muestra en [2] donde se usa un robot con una unidad de medición inercial MPU6050 que cuenta con giroscopio y acelerómetro para medir la inclinación y velocidad angular del cuerpo del robot y estas señales entran al controlador PID que se implementa con un Arduino Uno. También se le agrega una conexión inalámbrica mediante Bluetooth. Para los motores se tienen drivers con puentes H que inyectan el voltaje necesario a los motores de acuerdo con el control. Se usa Proteus



Figura 1. Robot a utilizar en el trabajo de tesis, Robot Elegoo Tumbler.

en conjunto de Matlab para poder sintonizar el controlador. Como resultado se obtiene un robot de dos ruedas que es capaz de mantenerse en balance.

En [3] se presenta un análisis detallado del modelado y control de un robot de autobalance. El modelo se obtiene mediante el enfoque de Newton y diagramas de cuerpo libre de las ruedas y cuerpo del robot. De esta manera se analizan todas las fuerzas que actúan en las ruedas, permitiendo la relación del movimiento lineal y de rotación del robot. Al analizar el circuito de los dos motores de DC se obtiene la ecuación diferencial que permite realizar la conexión entre la parte eléctrica y mecánica del robot. La importancia de esto radica en que los voltajes de los motores es la señal de salida del controlador. Finalmente, las ecuaciones diferenciales no lineales del modelado se linealizan en un punto de operación con un ángulo de inclinación cercano a cero. Al linealizar el sistema se observa que el modelo se puede desacoplar en el movimiento lineal del robot y el de rotación. Este artículo se concentra solo en controlar el balance y el movimiento lineal. De esta manera el modelo se simplifica mucho. Para el control se hace uso de un regulador lineal cuadrático (LQR por sus siglas en inglés) para retroalimentar los estados. Además, se mezcla la técnica del LQR con el control por modos deslizantes, en este caso se considera una perturbación

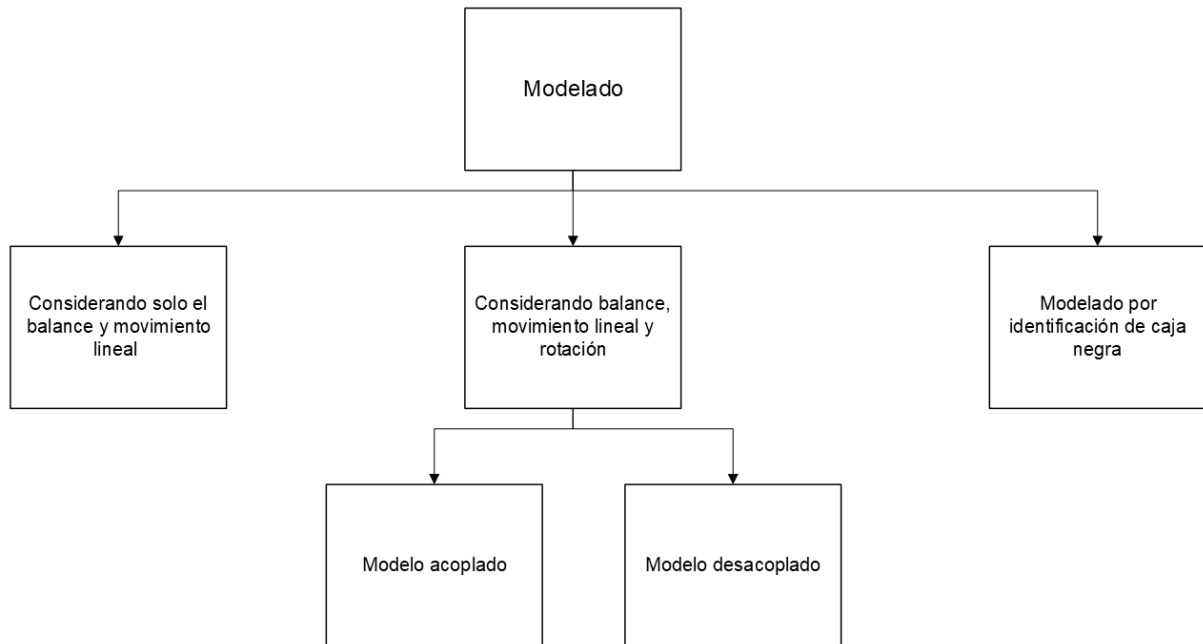


Figura 2. Tipos de modelado para los robots de dos ruedas de autobalance.

externa que afecta el comportamiento del robot. Para los resultados se simulan los modelos con el controlador LQR y el controlador LQR con modos deslizantes. Se muestra que la mezcla de LQR con modos deslizantes tuvo un mejor desempeño al estabilizar más rápido el sistema, aunque estuviera presente la perturbación.

En el artículo [4] se usa un robot Lego Mindstorms EV3 controlado mediante una superficie dinámica invariante. El modelado es del tipo acoplado ya que incluye el movimiento de rotación y el lineal. El controlador está compuesto por un lazo interno de tipo LQR que retroalimenta la posición y el ángulo de rotación, y un lazo externo basado en la superficie dinámica, cuya entrada es la posición de referencia. Este lazo incorpora algoritmos de auto sintonización que ajustan automáticamente los parámetros del controlador. El algoritmo de control se implementa en Matlab/Simulink y se evalúa utilizando diferentes trayectorias, con y sin auto sintonización. Como resultado los algoritmos de auto sintonización mejoran significativamente la precisión con la que el robot sigue las trayectorias. La parte de los algoritmos de auto sintonización se muestra en más detalle en [5].

Se observa que en [6] se utiliza un robot de autobalance usando un control robusto PD-PI en un ambiente percibido (SD por sus siglas en inglés). Para mejorar el comportamiento del robot se hace uso de un filtro de Kalman para mejorar la estabilización del sistema y su dinámica de navegación dentro del ambiente. También se agrega una comunicación inalámbrica mediante Bluetooth para usar el robot con un celular o con una computadora. Para poder implementar el controlador se usa un microcontrolador de STM32 de 32 bits.

Una de las ventajas más importantes del controlador implementado en este artículo es que a diferencia de un PID, es menos susceptible al ruido presente en los sensores para percibir el ambiente donde se mueve el robot. Para el modelado se usa el enfoque de energía de Euler-Lagrange siendo un modelo acoplado y no lineal. Además, se incorpora el modelo eléctrico al mecánico. Posteriormente se linealiza respecto a un ángulo de inclinación cercano a cero. El controlador PD está destinado a controlar balance del robot. El controlador PI se concentra en controlar la velocidad de los motores. Para controlar la rotación se usa un controlador proporcional (P). Para poder medir la inclinación y velocidad del cuerpo del robot se usa una unidad inercial MPU6050, sin embargo, las mediciones que se obtienen tienden a contener mucho ruido, para evitar se usa un algoritmo de filtro de Kalman que estime la velocidad e inclinación del robot. Para que el robot pueda percibir el ambiente en donde se encuentra se implementan sensores ultrasónicos, se simula este comportamiento mediante Matlab con Robot Toolbox. Para la sintonización de los controladores se simula el sistema con distintos valores de sintonización y se escoge la mejor opción. Finalmente se realizan experimentos comparando el comportamiento de este robot con otro con un controlador difuso. Como resultado, el control robusto presenta una estabilización más rápida.

En el artículo [7] se muestra una técnica de visión para controlar estos tipos de robots. Se trabaja con un robot que aparte de tener la función de autobalance cuenta con brazos manipuladores. Estos actuadores extras cambian la dinámica del modelo total a comparación de otros artículos. Se implementan cámaras RGB-D que mediante una red neuronal convolucional profunda (ConvNet) para que el robot sea capaz de navegar en un ambiente determinado. Para poder llevar a cabo todo el procesamiento que requiere este robot tiene incorporado una PC, lo cual le da un mayor poder de procesamiento a diferencia de otros robots analizados en otros estudios. Se realiza un modelo matemático que permita analizar el movimiento lineal y de rotación del robot, también, se considera la dinámica propia de los brazos manipuladores. Las técnicas de aprendizaje por refuerzo se usan para las tareas de que el robot se mueva en línea recta y en esquinas. Para el entrenamiento de la red neuronal se adquieren un conjunto de imágenes, así como rutas de pruebas que debe seguir el robot. Para los experimentos se usan dos trayectorias, una recta dentro de un pasillo y otra con una esquina de 90 grados. Primero se implementa en el mismo ambiente que se usa para el entrenamiento. También se prueba en un ambiente completamente nuevo como lo es una fábrica con máquinas de CNC, y el robot fue capaz de moverse de manera adecuada en esta fábrica.

En [8] se implementa un robot de autobalance, con la diferencia de que se considera una masa variable, esta masa se realiza mediante pesas de distintos valores que se

pueden agregar al robot. Se realiza un modelado por el enfoque de Euler-Lagrange y posteriormente es linealizado respecto a un ángulo cercano a cero. Y para la parte de control se usa un algoritmo de “machine learning” basado en un controlador adaptativo con lógica difusa proporcional integral (AFL-PI por sus siglas en inglés), para poder realizar todos estos cálculos se usa un microcontrolador Cortex M3. Durante la experimentación se recibe retroalimentación de los sensores integrados en una unidad BNO055 IMU que contiene un acelerómetro, giroscopio y un magnetómetro, con esto se puede medir la inclinación y velocidad del cuerpo del robot. De los algoritmos de “machine learning” utilizados se encuentran una red artificial neuronal (ANN), análisis discriminante lineal (LDA), “support vector machine” (SVM) y “k-Nearest Neighbors” (k-NN). De todos los algoritmos el que da mejores resultados es la red neuronal artificial (ANN), y al aplicarlo al robot con la carga variables, en conjunto con el controlador AFL-PI es capaz de mantener el balance del robot a pesar del cambio de masa. El desempeño de la estrategia de control implementada en este artículo ha demostrado ser mejor que la técnica clásica de control integral con lógica difusa.

Analizando [9] se muestra un modelado del robot con el enfoque de Newton. Al modelado se le aplica una técnica de control robusto en ROS basado en la técnica de control activo de rechazo de perturbaciones (ADRC). Para poder implementar estas técnicas de control se usa un Arduino Mega como una unidad de procesamiento de bajo nivel y una Raspberry Pi 4 como una unidad de procesamiento complementaria que procesa los comandos de alto nivel. El sistema cuenta con una unidad inercial MPU6050. Aparte de las medidas de esta unidad se cuenta con un observador para estimar perturbaciones externas o parámetros que el modelo dinámico no pueda aportar con exactitud. Para el control de ambos motores y la velocidad de estos se usa un PID por cada motor. La respuesta al implementar el controlador propuesto en este artículo provocó que fuera más rápida, suave, capaz de eliminar las perturbaciones y estabiliza al robot.

El estudio reportado en [10] utiliza un robot de autobalance con la característica que cuenta con ruedas tipo omnidireccionales, estas ruedas permiten el movimiento omnidireccional del robot. Este tipo movimiento provoca que el modelado sea diferente a los estudios anteriormente mencionados. Para controlar este robot se hace uso de un control predictivo basado en modelo.

Consumo de energía en un robot móvil

En el estudio [11] los autores realizan un modelo de consumo de energía para un robot móvil. El robot utilizado tiene dos ruedas con motores y una tercera rueda loca, de

esta manera se trabaja con un sistema que no tiene los problemas de inestabilidad de autobalance. La razón de realizar este artículo es encontrar un modelo de energía que implemente las pérdidas en todo el robot, ya que artículos previos no consideraban algunas pérdidas. Encontrar un modelo de este tipo permite que posteriormente se pueda optimizar el consumo de la energía de la batería. Antes de realizar el modelo de energía se modela la cinemática del robot, ya que algunas pérdidas están directamente relacionadas con la energía cinética. Las otras pérdidas son de los motores, los componentes electrónicos, la fricción y misceláneas.

De esta manera se obtiene la ecuación (1).

$$E_{battery} = E_{dc} + E_{kinetics} + E_{friction} + E_{electronics} \quad (1)$$

donde

- $E_{battery}$ es la energía disponible en la batería.
- E_{dc} es la energía disipada por los motores.
- $E_{kinetics}$ son las perdidas por energía cinética.
- $E_{friction}$ son las perdidas por fricción.
- $E_{electronics}$ son las perdidas en los componentes electrónicos.

Las pérdidas del motor se deben principalmente a las pérdidas eléctricas en la armadura y a la eficiencia de la caja reductora. La energía cinética depende de la velocidad de los motores. La energía perdida por la fricción depende del material donde este moviéndose el robot. Las pérdidas por componentes electrónicos se deben a la energía que consumen todos los componentes del control y drivers que tiene el robot. Muchos de los valores necesarios para obtener las pérdidas se obtienen midiendo directamente. Las corrientes en los motores mediante un sensor ACS712 y se introduce a un Arduino Mega 2560 mientras el robot está en movimiento, con estas corrientes se pueden calcular la energía consumida por los motores. Para medir las pérdidas de la caja reductora se debe desacoplar de los motores y se analiza el tipo de reductor para verificar el nivel de la eficiencia dada por estudios previos o por información del fabricante. Para las pérdidas de la energía cinética se usan encoders para medir la velocidad rotacional de los motores y posteriormente calcular la velocidad lineal, con la velocidad lineal ya se puede calcular la energía cinética. Para las pérdidas por fricción se toman medidas con los motores sin ruedas en comparativa con las ruedas sobre la superficie a trabajar, y de manera experimental se demuestra la validez del modelo de energía propuesto.

En el estudio [12] se realiza el análisis de la energía consumida por un robot de dos ruedas con motores. En este caso se implementan distintos algoritmos de control de seguimiento de trayectoria a un robot de dos ruedas con motor y una rueda loca. En cada algoritmo de control se registra el consumo de energía para analizar cuál tiene el menor consumo. Sin embargo, se demuestra que el consumir menos energía tiene como consecuencia perder en otros aspectos como la velocidad del robot o la precisión con la que sigue una trayectoria programada. Los criterios utilizados para comparar la energía consumida son los siguientes: consumo de energía cinética, distancia recorrida, la desviación del robot respecto a la trayectoria programada y el tiempo que le toma al robot recorrer la trayectoria. Las trayectorias que se utilizan cuentan con giros bruscos o trayectorias discontinuas, esto debido a que en estudios anteriores se utilizan trayectorias más ideales como círculos. Además, los controladores para robots móviles se concentran solo en controlar la cinemática, en este artículo se muestra que implementar el modelo dinámico junto el cinemático al controlador permite un control más completo. Asimismo, se utiliza un modelo de energía (1).

Se realiza un proceso de optimización con dos funciones objetivo, una a partir de la energía cinética (2). Esto debido a los criterios que se utilizan en este artículo.

$$\sum E_k = \sum_{t_0}^{t_n} \left(\frac{1}{2} m(v(t))^2 + \frac{1}{2} I(\omega(t))^2 \right) \quad (2)$$

donde

- m es la masa del objeto analizar.
- $v(t)$ es la velocidad lineal del objeto.
- I es el momento de inercia del objeto.
- $\omega(t)$ es la velocidad angular del objeto.

La distancia recorrida por el robot, descrita por la ecuación (3), es la segunda función objetivo necesaria para realizar el proceso de optimización.

$$S = \int_{t_0}^{t_n} |v(t)| dt \quad (3)$$

Los aspectos para controlar dependen del objetivo que se desee. Algunos ejemplos son el control de seguimiento de trayectoria, trayectoria, posición y seguimiento de punto a punto. En el caso de los robots móviles el aspecto mayor preocupación es el seguimiento de trayectoria. Los controladores evaluados son los siguientes: control por retroalimentación

de estados, Dubin, Lyapunov, geometría, Robins/Mathew y Kanayama. Al analizar los resultados se demuestra que el controlador por retroalimentación de estados es el que consume menos energía, pero tiene una mayor diferencia respecto a la trayectoria programada. Además, los controladores basados en Lyapunov y geometría fueron los que permiten que el robot siga las trayectorias cuadrada y discontinua sin desviarse de las mismas. Pero estos controladores consumieron más energía en comparación con los otros controladores utilizados. Otro factor que se considera al evaluar el consumo de energía es el tiempo de duración de la batería. Por lo tanto, además de los criterios utilizados en la literatura, se emplea el tiempo de duración de la batería para evaluar los algoritmos de control.

Seguimiento de trayectoria

El seguimiento de trayectoria en robots móviles de arquitectura diferencial ha sido un tema ampliamente estudiado en la literatura. Sin embargo, cuando se consideran robots con autobalance con dos ruedas, el problema de seguir una trayectoria se vuelve considerablemente más complejo debido a la necesidad de mantener el equilibrio dinámico mientras se realiza el desplazamiento en el plano. En el caso de los robots móviles de arquitectura diferencial convencional, el seguimiento de trayectoria se ha abordado principalmente desde una perspectiva cinemática. Un trabajo representativo es el mostrado en [13], quien propone un controlador basado en errores de pose y funciones de Lyapunov, garantizando convergencia asintótica para sistemas no holonómicos. Este enfoque ha servido como base para múltiples desarrollos posteriores y es ampliamente citado en la literatura de control cinemático de robots móviles.

Posteriormente, en [14] se introduce un esquema de seguimiento de trayectoria suave para robots móviles, formulando el error en un marco de referencia móvil y demostrando estabilidad mediante Lyapunov. Aunque este enfoque se centra en robots diferenciales sin restricciones de balance, su estructura ha sido adoptada en múltiples trabajos posteriores como lazo externo de seguimiento.

Una transición desde el control cinemático hacia uno dinámico se realiza en el trabajo [15]. Se propone un controlador adaptativo para robots móviles diferenciales que incorpora incertidumbres dinámicas y fricción, logrando seguimiento de trayectoria con compensación adaptativa.

En el caso de los robots de autobalance, la mayor cantidad de los trabajos se enfocan en mantener el equilibrio del robot. Sin embargo, uno de los primeros trabajos que integra el seguimiento de trayectoria y el balance es presentado en [16]. Los autores desarrollan un

controlador capaz de mantener el equilibrio del péndulo invertido mientras el robot sigue trayectorias planificadas, validando su propuesta mediante experimentos en un robot real. Este artículo es considerado un referente pionero en navegación de robots tipo Segway.

En el trabajo presentado en [17] se usa un control de modos deslizantes con backstepping para un grupo de robots móviles con el fin de lograr el seguimiento de una trayectoria. El grupo de robots se compone de robots con autobalance y robots móviles convencionales de arquitectura diferencial. Las trayectorias analizadas son línea recta y un círculo, con los robots en formación triangular. Sin embargo, los resultados son presentados únicamente en simulación.

En [18] se desarrolla un control de navegación planificada para un robot móvil de servicio con autobalance. Se hace uso de un modelo linealizado, con este modelo se diseñan un rastreador de trayectoria cinemático y dos controladores adaptativos para rastrear las órdenes de velocidad y giro. Se utiliza un generador de trayectorias con reconocimiento humano para proporcionar las trayectorias deseadas entre las personas y el robot. En este caso se desarrollan pruebas experimentales en una plataforma desarrollada de forma casera y cuenta con una DSP para las tareas de control.

En [19] presenta un novedoso robot con autobalance capaz de reconfigurarse morfológicamente de forma autónoma entre configuraciones de dos y cuatro ruedas, permitiendo que pueda superar obstáculos pasando por arriba o por debajo de estos. Se usa un controlador PID adaptativo para garantizar la estabilidad durante las transiciones de las configuraciones del robot. Se realizan pruebas experimentales en una plataforma desarrollada de manera casera. Sin embargo, no se analiza el seguimiento de trayectoria, pero sí se controla el ángulo de inclinación y el funcionamiento de la transición de las dos configuraciones del robot.

En [20] se diseña e implementa completamente un nuevo robot híbrido de dos ruedas con autobalance, capaz de rastrear objetos y evitar obstáculos de forma eficiente. Se utiliza el marco del Internet de las Cosas (IoT) para el control y la monitorización remotos mediante una interfaz inalámbrica. Para medir la proximidad de la posición de un objeto respecto al robot, se usa un sistema de visión junto con un sensor ultrasónico. Para la evasión de obstáculos se implementa un controlador por lógica difusa. Finalmente se muestran las pruebas experimentales realizadas, se hace uso de MATLAB Fuzzy Toolbox con un paquete de Arduino, además de una tarjeta Raspberry Pi.

En [21] se aborda el seguimiento de trayectoria para un robot con autobalance. Se obtiene el modelo del robot y el controlador multivariable desacoplado con respecto a la velocidad lineal del robot, el ángulo de dirección y el equilibrio del robot. Sin embargo, solo se presentan resultados en simulación.

En [22] se desarrolla un control en cascada para que un robot con autobalance pueda seguir un perfil de velocidad mientras se mantiene el equilibrio. Este controlador se compara con un control por modos deslizantes.

En [23] se desarrolla un controlador jerárquico de modos deslizantes combinado con un control de modos deslizantes convencional para un robot de dos ruedas con autobalance. El controlador se divide en dos componentes: el controlador jerárquico controla la velocidad y estabiliza el vehículo. El controlador de modos deslizantes convencional controla el ángulo de inclinación. Mostrando mediante simulación que el enfoque utilizado es superior al método convencional de modos deslizantes en términos de tiempo de respuesta y reducción de errores durante el control estable de la posición, el ángulo de cabeceo y el ángulo de rotación. Además, se implementa un algoritmo de árbol aleatorio de exploración rápida para la planificación de trayectorias, permitiendo que el robot navegue por lugares complejos. Esta propuesta permite al robot seguir la trayectoria deseada al realizar las simulaciones pertinentes.

Brecha de investigación

A partir del análisis de la literatura, se observa que, si bien existen numerosos trabajos sobre estabilización de robots con autobalance y una amplia cantidad de estudios sobre seguimiento de trayectoria en robots diferenciales, los trabajos que integran explícitamente ambas tareas y las validan experimentalmente son relativamente escasos. Debido a la complejidad del sistema y de los algoritmos de control utilizados, algunos artículos no implementan pruebas experimentales, solo en simulación.

Por lo tanto, aún existe una brecha importante en cuanto a implementaciones experimentales completas que integren control basado en modelos dinámicos y seguimiento de trayectorias en robots con autobalance de bajo costo, lo que justifica la relevancia del presente trabajo.

Justificación del trabajo

Los robots de dos ruedas con autobalance representan una clase de sistemas subactuados, no lineales e inherentemente inestables, cuyo control plantea desafíos significativos desde el punto de vista teórico y experimental. Aunque el problema de estabilización del equilibrio ha sido ampliamente estudiado, la integración efectiva de esta tarea con el seguimiento de trayectoria continúa siendo un tema de investigación activo y con brechas en la literatura. En particular, en la realización de pruebas experimentales implementadas en plataformas

de bajo costo, didácticas y de arquitectura abierta, además de analizar el desempeño energético de la batería.

En este contexto, el presente trabajo de tesis aborda el problema de control de un robot móvil de dos ruedas con autobalance de arquitectura diferencial, tomando como plataforma experimental el robot Elegoo Tumbler, y realiza aportaciones relevantes tanto desde el punto de vista del diseño de control como desde la evaluación experimental del desempeño, considerando el equilibrio del robot y el seguimiento de una trayectoria circular.

En primer lugar, se logró diseñar e implementar un controlador por retroalimentación de estados basado en un modelo dinámico del robot, capaz de mantener el equilibrio del sistema alrededor de la posición vertical.

En segundo lugar, se desarrolló un controlador de seguimiento de trayectoria basado en técnicas de Lyapunov, diseñado a partir del modelo cinemático del robot diferencial, que garantiza la convergencia asintótica del error hacia una trayectoria circular deseada. Aplicar este tipo de controladores en robots de dos ruedas con autobalance requiere un cuidadoso acoplamiento con la dinámica de balance, lo que incrementa la complejidad del problema.

En tercer lugar, se diseña una estrategia de conmutación entre el controlador de equilibrio y el controlador de seguimiento de trayectoria, permitiendo que el robot mantenga el equilibrio dinámico mientras ejecuta el movimiento requerido para seguir la trayectoria deseada. Esta estrategia refleja un enfoque jerárquico y práctico. Implementado y validado en una plataforma real de bajo costo, lo cual no ha sido reportado en trabajos previos.

Adicionalmente, el trabajo presenta una comparación experimental en términos de desempeño entre los controladores desarrollados que son los siguientes:

1. Controlador E1. Controlador PI-PD para el equilibrio implementado de fábrica en el robot Elegoo Tumbler.
2. Controlador E2. Controlador por retroalimentación de estados para el equilibrio.
3. Controlador R1. Controlador PI-PD (con sintonización distinta al implementado de fábrica) para el equilibrio combinado con el controlador de seguimiento de trayectoria basado en técnicas de Lyapunov.
4. Controlador R2. Controlador por retroalimentación de estados para el equilibrio combinado con el controlador de seguimiento de trayectoria basado en técnicas de Lyapunov.

La comparación entre los controladores R1 y R2 se realiza no solo en términos del error de seguimiento de trayectoria, sino también considerando el consumo energético del sistema

medido a través de la duración de la carga de la batería, lo cual constituye una aportación particularmente relevante.

Por lo tanto, este trabajo no solo contribuye al diseño e implementación de estrategias de control para robots de dos ruedas con autobalance, sino que también introduce un criterio de evaluación adicional basado en el consumo energético, proporcionando una perspectiva más completa del desempeño del sistema. Esta combinación de modelado, control, implementación experimental y análisis energético refuerza la relevancia del presente estudio.

Organización del trabajo de tesis

En el Capítulo 1 se muestra el proceso para obtener el modelo dinámico del robot considerando el movimiento lineal. Se usan los enfoques de Euler-Lagrange y de trabajo virtual. El enfoque del trabajo virtual se utiliza ya que no ha sido reportado que se use para este tipo de robots y se compara con respecto al de Euler-Lagrange. Asimismo, se obtiene el modelo cinemático de un robot diferencial, que será utilizado para diseñar el controlador de seguimiento de trayectoria.

El Capítulo 2 se presenta el procedimiento para diseñar e implementar los controladores por retroalimentación de estados y el basado en técnicas de Lyapunov. Además, se explica la forma de acoplar el control de equilibrio con el de seguimiento de trayectoria.

El Capítulo 3 se muestran los resultados obtenidos en las simulaciones y la discusión de los mismos.

El Capítulo 4 se expone la descripción de la plataforma que se utiliza y los resultados de las pruebas experimentales realizadas.

Finalmente, se presentan las conclusiones del trabajo de investigación, donde se detallan las soluciones y áreas de oportunidad en los diferentes temas tocados por este trabajo.

MODELADO DINÁMICO DEL ROBOT MÓVIL DE DOS RUEDAS CON AUTOBALANCE

1.1. Introducción

Este capítulo tiene como principal objetivo mostrar la obtención de los modelos del robot de dos ruedas con autobalance. El contenido se organiza de la siguiente manera.

- En la sección 1.2, se muestra el procedimiento para obtener el modelo del robot considerando únicamente el movimiento para mantener el equilibrio mediante el enfoque de Euler-Lagrange. Se establecen los dos subsistemas que componen al robot y el modelo final. Por lo tanto, hay tres subsecciones:
 - En la subsección 1.2.1, se modela el primer subsistema que compone al robot, un péndulo invertido.
 - En la subsección 1.2.2, se modela el segundo subsistema siendo una rueda.
 - En la subsección 1.2.3, se modela el robot de dos ruedas con autobalance combinando los dos subsistemas de las subsecciones 1.2.1 y 1.2.2. Las ecuaciones obtenidas en esta parte se usan para diseñar el controlador de equilibrio del robot.
- En la sección 1.3, se obtiene el modelo del robot de dos ruedas con autobalance mediante el enfoque de trabajo virtual. Resultando en las mismas ecuaciones obtenidas en 1.2.3.
- En la sección 1.4, se realiza una comparación entre el enfoque de Euler-Lagrange y el de trabajo virtual.

- En la sección 1.5, se verifica, mediante simulación, el comportamiento del modelo del robot de dos ruedas con autobalance. Para esto se usan las ecuaciones
- En la sección 1.6, se modela un robot de arquitectura diferencial. Este modelo se usa para diseñar el controlador para el seguimiento de trayectoria.

1.2. Modelo mediante el enfoque de Euler Lagrange

Para el modelado dinámico del robot de autobalance se hace uso principalmente de las técnicas de Euler-Lagrange. El sistema consiste principalmente de dos subsistemas, un péndulo invertido y una rueda. Estos dos subsistemas se pueden modelar para comprender de mejor manera como se comporta el sistema. Posteriormente se unen los dos subsistemas para obtener el modelado del sistema de equilibrio del robot. Para validar que el modelo sea correcto se analizan los resultados obtenidos gráficamente buscando que dicho comportamiento sea congruente con la física del sistema.

1.2.1. Modelo del péndulo invertido

Para comenzar con el primer modelado se trabaja con el péndulo invertido, para ello se usa el esquema mostrado en la figura (1.1).

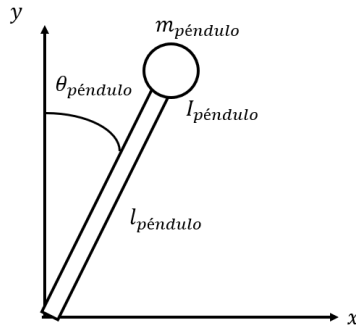


Figura 1.1. Esquema del péndulo invertido.

donde

- $m_{p\acute{e}ndulo}$ es la masa concentrada del péndulo invertido.
- $l_{p\acute{e}ndulo}$ es la longitud del centro de masa del péndulo respecto al marco de referencia.
- $\theta_{p\acute{e}ndulo}$ indica el ángulo de inclinación del péndulo respecto al eje vertical, siendo el sentido positivo el sentido horario.

- $I_{\text{péndulo}}$ es el momento de inercia del péndulo.
- $v_{\text{péndulo}}$ es la velocidad lineal del péndulo invertido.
- $\omega_{\text{péndulo}}$ es la velocidad angular del péndulo invertido.
- $x_{\text{péndulo}}$ es la coordenada en el eje x del centro de masa del péndulo invertido.
- $y_{\text{péndulo}}$ es la coordenada en el eje y del centro de masa del péndulo invertido.
- g es la aceleración de la gravedad.
- $K_{\text{péndulo}}$ es la energía cinética del péndulo invertido.
- $V_{\text{péndulo}}$ es la energía potencial del péndulo invertido.
- $L_{\text{péndulo}}$ es el Lagrangiano del péndulo invertido.
- $c_{\omega \text{ péndulo}}$ es el coeficiente de fricción del movimiento de rotación del péndulo respecto a la base del péndulo.
- $q_{\text{péndulo}}$ es la coordenada generalizada del sistema, que corresponde al ángulo de inclinación del péndulo invertido $\theta_{\text{péndulo}}$.
- $\dot{q}_{\text{péndulo}}$ es la primer derivada de la coordenada generalizada del sistema, que corresponde a $\omega_{\text{péndulo}}$.
- $\tau_{\text{péndulo}}$ indica la entrada de control al sistema.
- $x_{1 \text{ péndulo}}$ corresponde al estado uno del sistema, siendo el ángulo de inclinación del péndulo invertido $\theta_{\text{péndulo}}$.
- $x_{2 \text{ péndulo}}$ corresponde al estado dos del sistema, siendo la velocidad angular del péndulo invertido.
- $\dot{x}_{1 \text{ péndulo}}$ es la primer derivada del estado uno del sistema.
- $\dot{x}_{2 \text{ péndulo}}$ es la primer derivada del estado dos del sistema.

Para poder realizar el modelado se deben establecer las energías cinéticas y potenciales del péndulo usando el método dado por [24]. Primero se establecen las coordenadas generalizadas del sistema $q_1 = \theta$ y $\dot{q}_2 = \dot{\theta} = \omega$. En las ecuaciones (1.1) y (1.2) se muestran los cálculos para obtener la energía cinética y potencial respectivamente.

$$K_{\text{péndulo}} = \frac{1}{2} m_{\text{péndulo}} v_{\text{péndulo}}^2 + \frac{1}{2} I_{\text{péndulo}} \omega_{\text{péndulo}}^2 \quad (1.1)$$

$$V_{\text{péndulo}} = m_{\text{péndulo}} \begin{bmatrix} 0 & g & 0 \end{bmatrix} \begin{bmatrix} l_{\text{péndulo}} \sin \theta_{\text{péndulo}} \\ l_{\text{péndulo}} \cos \theta_{\text{péndulo}} \\ 0 \end{bmatrix} = m_{\text{péndulo}} g l_{\text{péndulo}} \cos \theta_{\text{péndulo}} \quad (1.2)$$

En la ecuación (1.1) se observa que la energía cinética corresponde a dos tipos de velocidad, la velocidad lineal y angular del péndulo. En la parte de la velocidad angular el término I_p indica el momento de inercia del péndulo.

Con las energías definidas en las ecuaciones (1.1) y (1.2) se procede a calcular el Lagrangiano que se muestra en la ecuación (1.3).

$$\begin{aligned} L_{\text{péndulo}} &= K_{\text{péndulo}} - V_{\text{péndulo}} \\ &= \frac{1}{2} m_{\text{péndulo}} v_{\text{péndulo}}^2 + \frac{1}{2} I_{\text{péndulo}} \omega_{\text{péndulo}}^2 - m_{\text{péndulo}} g l_{\text{péndulo}} \cos \theta_{\text{péndulo}} \end{aligned} \quad (1.3)$$

Para poder continuar con el procedimiento se procede a plantear el Lagrangiano en términos de las coordenadas generalizadas. Este procedimiento se plantea en la ecuaciones (1.4) que describen la cinemática del péndulo invertido.

$$\begin{aligned} x_{\text{péndulo}} &= l_{\text{péndulo}} \sin \theta_{\text{péndulo}} \\ y_{\text{péndulo}} &= l_{\text{péndulo}} \cos \theta_{\text{péndulo}} \\ \dot{x}_{\text{péndulo}} &= l_{\text{péndulo}} \cos(\theta_{\text{péndulo}}) \dot{\theta}_{\text{péndulo}} = \omega_{\text{péndulo}} l_{\text{péndulo}} \cos \theta_{\text{péndulo}} \\ \dot{y}_{\text{péndulo}} &= -l_{\text{péndulo}} \sin(\theta_{\text{péndulo}}) \dot{\theta}_{\text{péndulo}} = -\omega_{\text{péndulo}} l_{\text{péndulo}} \sin \theta_{\text{péndulo}} \end{aligned} \quad (1.4)$$

Con la cinemática se establece el término $v_{\text{péndulo}}^2$ en función de las coordenadas generalizadas, esto se muestra en la ecuación (1.5).

$$\begin{aligned} v_{\text{péndulo}}^2 &= \dot{x}_{\text{péndulo}}^2 + \dot{y}_{\text{péndulo}}^2 \\ &= \omega_{\text{péndulo}}^2 l_{\text{péndulo}}^2 (\cos^2(\theta_{\text{péndulo}}) + \sin^2(\theta_{\text{péndulo}})) = \omega_{\text{péndulo}}^2 l_{\text{péndulo}}^2 \end{aligned} \quad (1.5)$$

De esta manera el Lagrangiano queda con la forma mostrada en (1.6).

$$L_{\text{péndulo}} = \frac{1}{2} m_{\text{péndulo}} \omega_{\text{péndulo}}^2 l_{\text{péndulo}}^2 + \frac{1}{2} I_{\text{péndulo}} \omega_{\text{péndulo}}^2 - m_{\text{péndulo}} g l_{\text{péndulo}} \cos \theta_{\text{péndulo}} \quad (1.6)$$

Asimismo, se incluye la función de disipación de Rayleigh para modelar las fuerzas de fricción actuantes en la base del péndulo. Esta función se detalla en la ecuación (1.7).

$$\mathcal{R}_{\text{péndulo}} = \frac{1}{2} c_{\omega \text{ péndulo}} \omega_{\text{péndulo}}^2 \quad (1.7)$$

En la función de disipación de Rayleigh el término $c_{\omega \text{ péndulo}}$ es el coeficiente de fricción del movimiento de rotación del péndulo respecto a la base. Lo siguiente es realizar las operaciones del enfoque de Euler-Lagrange, descritas por (1.8).

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L_{\text{péndulo}}}{\partial \dot{q}_{\text{péndulo}}} \right) - \frac{\partial L_{\text{péndulo}}}{\partial q_{\text{péndulo}}} + \frac{\partial \mathcal{R}_{\text{péndulo}}}{\partial \dot{q}_{\text{péndulo}}} \\ = \frac{d}{dt} \left(\frac{\partial L_{\text{péndulo}}}{\partial \omega_{\text{péndulo}}} \right) - \frac{\partial L_{\text{péndulo}}}{\partial \theta_{\text{péndulo}}} + \frac{\partial \mathcal{R}_{\text{péndulo}}}{\partial \omega_{\text{péndulo}}} = \tau_{\text{péndulo}} \end{aligned} \quad (1.8)$$

Al realizar las operaciones de (1.8) se obtiene (1.9).

$$\begin{aligned} (m_{\text{péndulo}} l_{\text{péndulo}}^2 + I_{\text{péndulo}}) \dot{\omega}_{\text{péndulo}} - m_{\text{péndulo}} g l_{\text{péndulo}} \sin \theta_{\text{péndulo}} \\ + c_{\omega \text{ péndulo}} \omega_{\text{péndulo}} = \tau_{\text{péndulo}} \end{aligned} \quad (1.9)$$

El término $\tau_{\text{péndulo}}$ indica la entrada de control al sistema, concretamente es el par que se introduce al péndulo. Y de esta manera se obtiene la ecuación que describe el comportamiento del sistema.

La ecuación diferencial de (1.9) se debe pasar a espacio de estados para poder realizar las simulaciones. Para esto se usa el cambio de variable mostrado en (1.10).

$$\begin{aligned} x_{1 \text{ péndulo}} &= \theta_{\text{péndulo}} \\ x_{2 \text{ péndulo}} &= \dot{x}_{\text{péndulo}} = \omega_{\text{péndulo}} \end{aligned} \quad (1.10)$$

Y se obtiene el modelo de espacio de estados mostrado en la ecuación (1.11).

$$\begin{aligned} \dot{x}_{1 \text{ péndulo}} &= x_{2 \text{ péndulo}} \\ \dot{x}_{2 \text{ péndulo}} &= \dot{\omega}_{\text{péndulo}} = \frac{\tau_{\text{péndulo}} + m_{\text{péndulo}} g l_{\text{péndulo}} \sin \theta_{\text{péndulo}} - c_{\omega \text{ péndulo}} \omega_{\text{péndulo}}}{m_{\text{péndulo}} l^2 + I_p} \end{aligned} \quad (1.11)$$

En la figura (1.2) se muestran el ángulo de salida del péndulo invertido, se puede observar que después de un tiempo el péndulo llega a un ángulo de $\pi \approx 3.141592$, lo cual tiene sentido ya que indica que el péndulo invertido queda en posición totalmente hacia abajo.

Y en la figura (1.3) se muestra la velocidad angular del péndulo, mostrando que debido a la fricción la velocidad va disminuyendo, lo cual también es un comportamiento razonable.

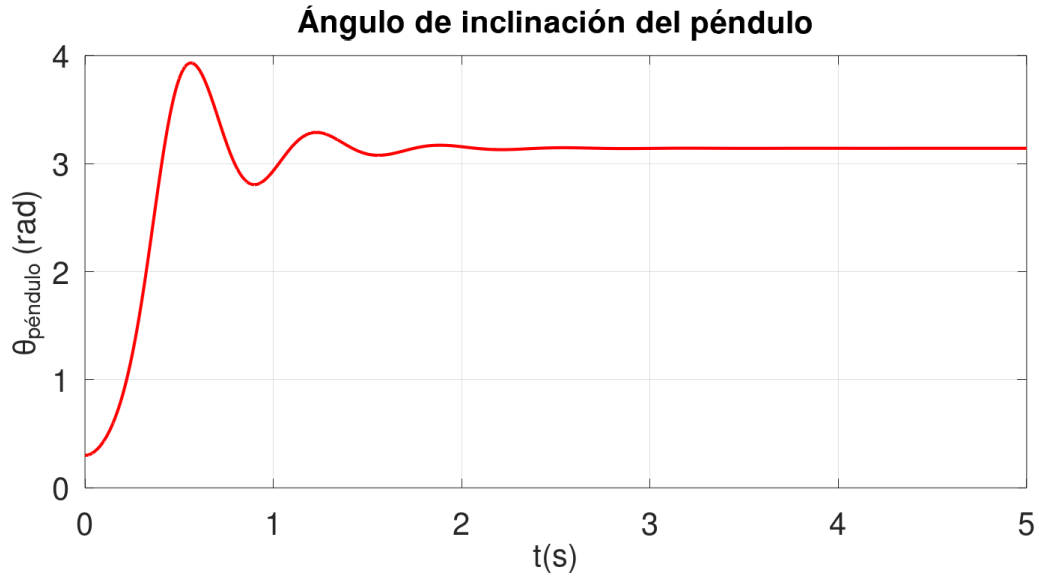


Figura 1.2. Ángulo de inclinación del péndulo invertido.

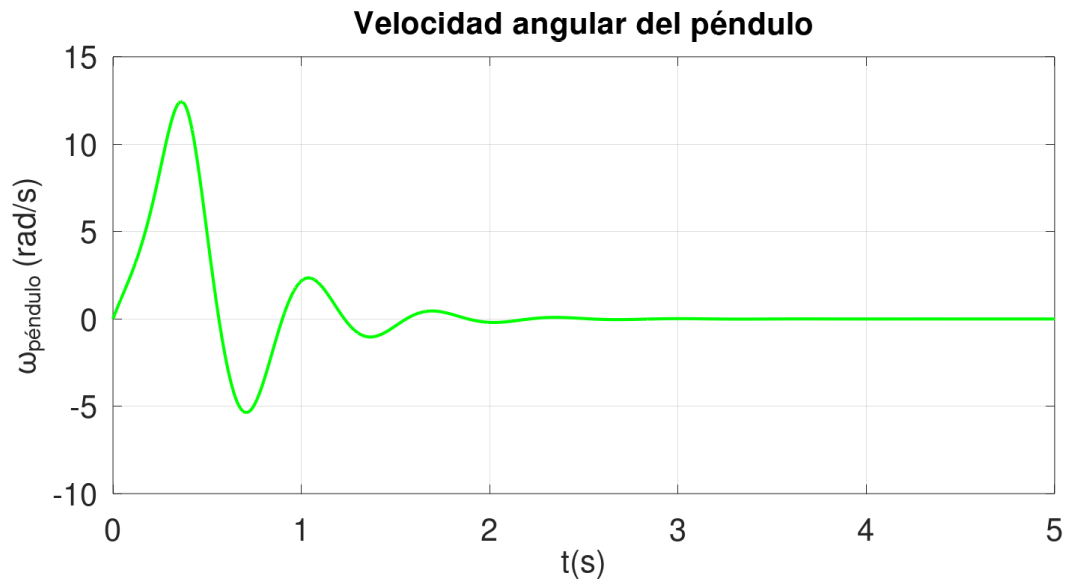


Figura 1.3. Velocidad angular del péndulo invertido.

1.2.2. Modelo de la rueda

El otro sistema que incluye el robot corresponde a un conjunto de ruedas, que se modelan como una sola rueda que se muestra en la figura (1.4).

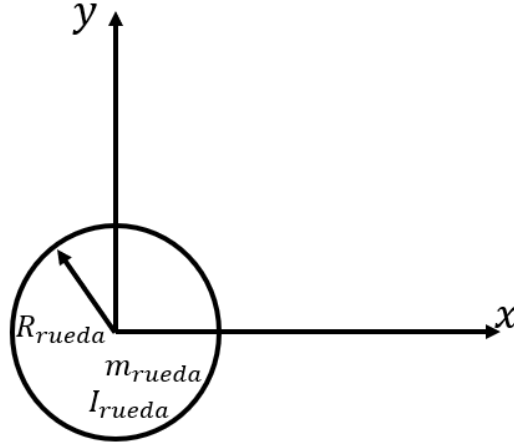


Figura 1.4. Diagrama de la rueda.

De manera análoga al procedimiento realizado con el péndulo invertido, se obtienen la energía cinética y potencial del sistema mostradas en las ecuaciones (1.12) y (1.13). La energía potencial es cero debido a que la rueda se encuentra justo en el marco de referencia. Para comprender las ecuaciones de mejor manera se establece la nomenclatura a utilizar.

- R_{rueda} es el radio de la rueda.
- F_{rueda} es la fuerza que es aplicada a la rueda.
- m_{rueda} es la masa concentrada de la rueda.
- v_{rueda} es la velocidad de la rueda.
- I_{rueda} es el momento de inercia respecto al centro de masa de la rueda.
- ω_{rueda} es la velocidad angular de la rueda.
- x_{rueda} es la distancia recorrida por la rueda respecto al marco de referencia.
- K_{rueda} es la energía cinética de la rueda.
- V_{rueda} es la energía potencial de la rueda respecto al marco de referencia.
- L_{rueda} es el Lagrangiano de la rueda.
- q_{rueda} es la coordenada generalizada del sistema, que corresponde a la distancia recorrida x_{rueda} de la rueda.
- $\dot{q}_{\text{rueda}}$ es la primera derivada de la coordenada generalizada del sistema, que corresponde a la velocidad de la rueda v_{rueda} .

- $x_{1\text{rueda}}$ corresponde al estado uno del sistema, siendo la distancia recorrida x_{rueda} de la rueda.
- $x_{2\text{rueda}}$ corresponde al estado dos del sistema, siendo la velocidad de la rueda v_{rueda} .
- $\dot{x}_{1\text{rueda}}$ es la primera derivada del estado uno del sistema.
- $\dot{x}_{2\text{rueda}}$ es la primera derivada del estado dos del sistema.

$$K_{\text{rueda}} = \frac{1}{2}m_{\text{rueda}}v_{\text{rueda}}^2 + \frac{1}{2}I_{\text{rueda}}\omega_{\text{rueda}}^2 \quad (1.12)$$

$$V_{\text{rueda}} = 0 \quad (1.13)$$

Al obtener el Lagrangiano se usa el momento de inercia de un cilindro en el parámetro I_{rueda} , este procedimiento se muestra en (1.14).

$$\begin{aligned} L_{\text{rueda}} = K_{\text{rueda}} - V_{\text{rueda}} &= \frac{1}{2}m_{\text{rueda}}v_{\text{rueda}}^2 + \frac{1}{2}I_{\text{rueda}}\omega_{\text{rueda}}^2 - 0 \\ &= \frac{1}{2}m_{\text{rueda}}v_{\text{rueda}}^2 + \frac{1}{2}\left(\frac{1}{2}m_{\text{rueda}}R_{\text{rueda}}^2\right)\left(\frac{v_{\text{rueda}}^2}{R_{\text{rueda}}^2}\right)^2 \\ &= \frac{3}{4}m_{\text{rueda}}v_{\text{rueda}}^2 \end{aligned} \quad (1.14)$$

Con el Lagrangiano se procede a utilizar la ecuación mostrada en (1.15).

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial L_{\text{rueda}}}{\partial \dot{q}_{\text{rueda}}}\right) - \frac{\partial L_{\text{rueda}}}{\partial q_{\text{rueda}}} \\ = \frac{d}{dt}\left(\frac{\partial L_{\text{rueda}}}{\partial v_{\text{rueda}}}\right) - \frac{\partial L_{\text{rueda}}}{\partial x_{\text{rueda}}} = F_{\text{rueda}} \end{aligned} \quad (1.15)$$

Al aplicar las operaciones que se muestran en (1.15) se obtiene la ecuación diferencial mostrada en (1.16).

$$\frac{3}{2}m_{\text{rueda}}\ddot{x}_{\text{rueda}} = F_{\text{rueda}} \quad (1.16)$$

Se procede obtener la representación del sistema en espacio de estados para poder realizar las simulaciones. Los estados se definen en (1.17).

$$\begin{aligned} x_{1\text{rueda}} &= x_{\text{rueda}} \\ x_{2\text{rueda}} &= \dot{x}_{1\text{rueda}} = v_{\text{rueda}} \end{aligned} \quad (1.17)$$

De esta forma se obtiene las ecuaciones de la representación en espacio de estados mostrada en (1.18).

$$\begin{aligned}\dot{x}_{1\text{rueda}} &= x_{2\text{rueda}} \\ \dot{x}_{2\text{rueda}} &= \frac{F_{\text{rueda}}}{3m_{\text{rueda}}}\end{aligned}\tag{1.18}$$

Con este modelo se pueden realizar las simulaciones para analizar si el modelo es correcto. Los resultados de las simulaciones se muestran en las figuras (1.5) y (1.6).

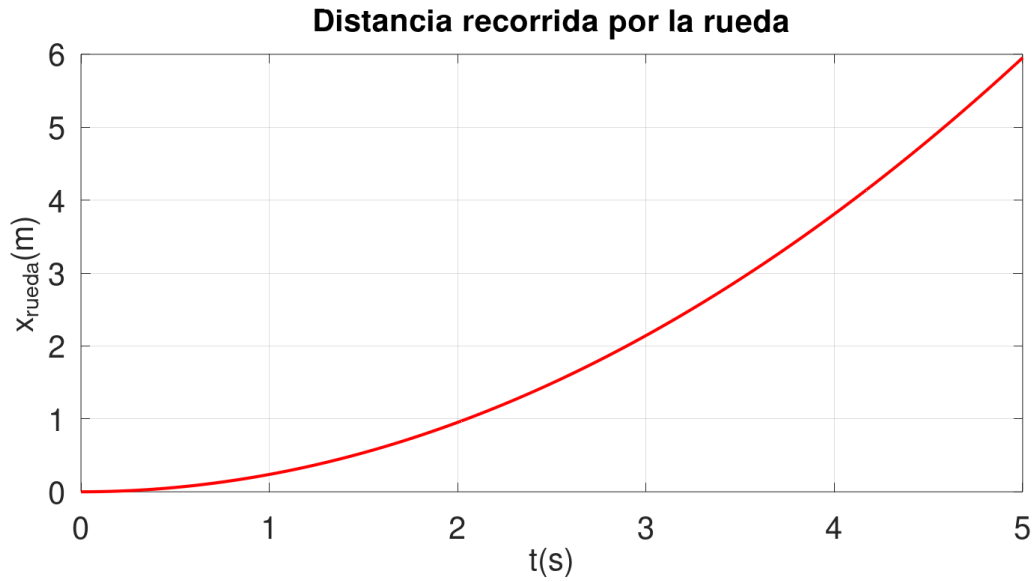


Figura 1.5. Distancia recorrida por la rueda.

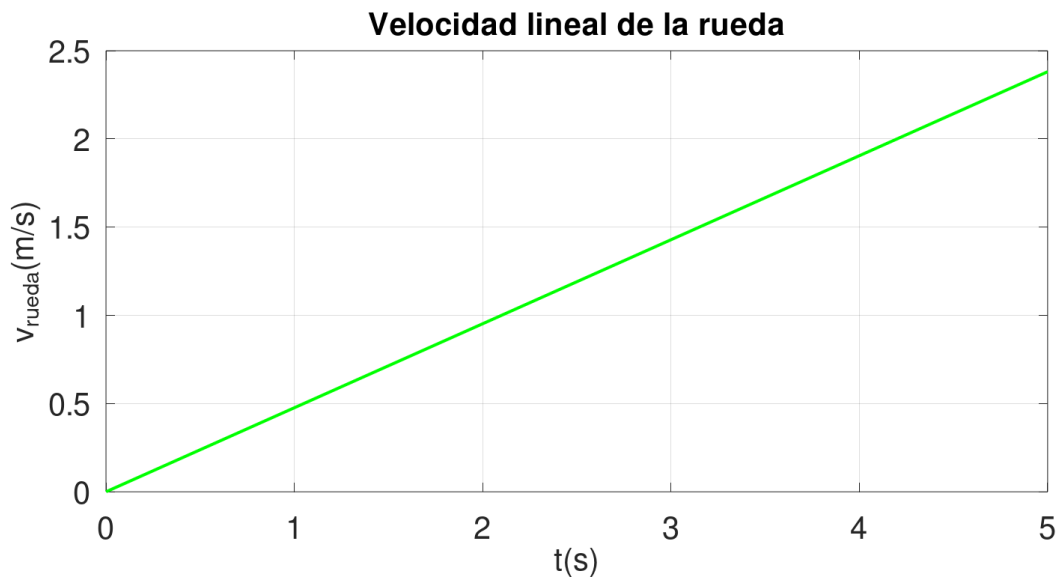


Figura 1.6. Velocidad de la rueda.

1.2.3. Modelo del robot de autobalance

En este apartado se unen los dos subsistemas anteriores para poder obtener el modelo que corresponde a la parte de autobalance del robot. Para esto se usa el esquema que se muestra en la figura (1.7).

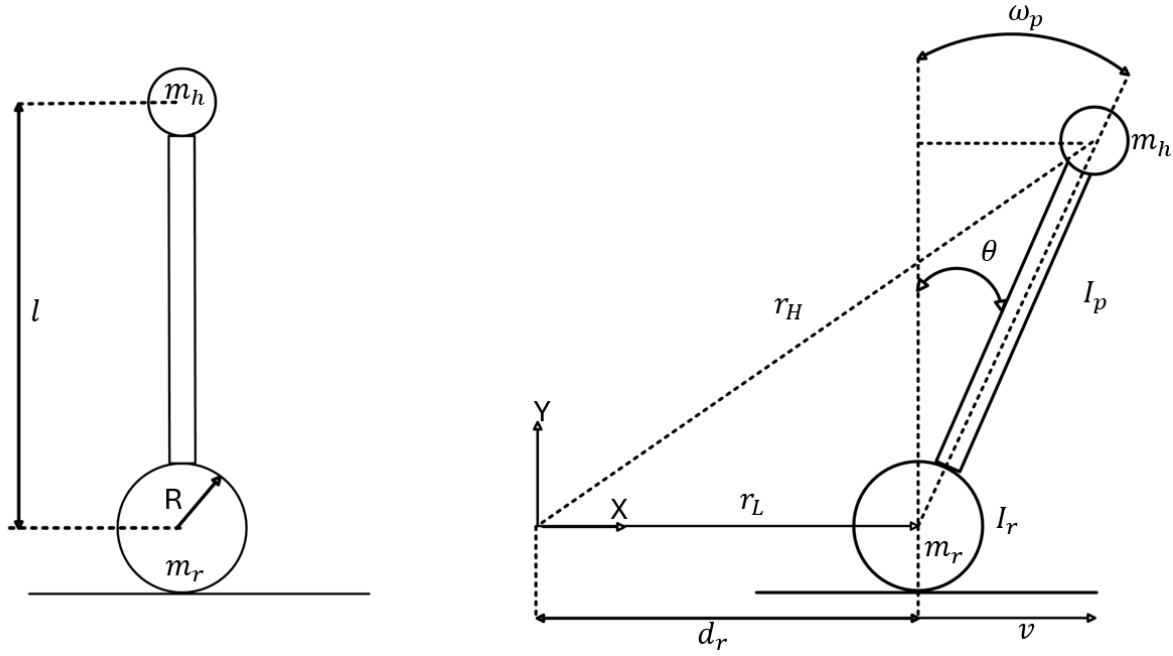


Figura 1.7. Diagrama del robot de autobalance.

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.
- $\mathbf{r}_H$ es la distancia del centro de masa del robot al marco de referencia.

- $\mathbf{r}_L$ 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.
- K_r es la energía cinética del conjunto de ruedas del robot.
- K_p es la energía cinética del cuerpo del robot.
- V_r es la energía potencial del conjunto de ruedas del robot.
- V_p es la energía potencial del cuerpo del robot.
- L_p es el Lagrangiano del robot.
- $\mathcal{R}_p$ corresponde a la función de disipación de Rayleigh, que indica la energía disipada por la fricción.
- c_v es el coeficiente de fricción de las ruedas respecto al suelo en el que se desplaza el robot.
- c_ω es el coeficiente de fricción del movimiento de rotación del péndulo respecto a la unión con las ruedas.
- τ_1 corresponde a la entrada de la ecuación de la sumatoria de fuerzas, siendo la fuerza $\frac{\tau}{R}$.
- τ es el par de los dos motores del robot.
- τ_2 corresponde a la entrada de la ecuación de la sumatoria de pares, siendo $\tau_2 = 0$.
- q_1 es la primera coordenada generalizada del sistema, siendo la distancia recorrida d_r .
- q_2 es la segunda coordenada generalizada del sistema, siendo el ángulo de inclinación del robot θ .
- $\dot{q}_1$ es la primera derivada de la primera coordenada generalizada, siendo la velocidad de las ruedas del robot v .

- $\dot{q}_2$ es la primera derivada de la segunda coordenada generalizada, siendo la velocidad angular del robot ω_p .
- x_1 corresponde al estado uno del sistema, siendo la distancia recorrida por las ruedas del robot d_r .
- x_2 corresponde al estado dos del sistema, siendo la velocidad de las ruedas del robot v .
- x_3 corresponde al estado tres del sistema, siendo el ángulo de inclinación del robot θ .
- x_4 corresponde al estado cuatro del sistema, siendo la velocidad angular del robot.
- $\dot{x}_1$ es la primera derivada del estado uno del sistema.
- $\dot{x}_2$ es la primera derivada del estado dos del sistema.
- $\dot{x}_3$ es la primera derivada del estado tres del sistema.
- $\dot{x}_4$ es la primera derivada del estado cuatro del sistema.

Para el modelado se usa de referencia [25], solo considerando el movimiento para mantener el robot en equilibrio. El trabajo realizado en el artículo citado se basa en el enfoque de Euler-Lagrange de cada parte del robot de autobalance. De este modo se obtiene la energía cinética y potencial del sistema para obtener el modelo alternativo al presentado en el artículo, aunque con metodología similar. Debido a que el robot se compone de dos subsistemas, un péndulo invertido y una rueda, cada uno de estos subsistemas tienen sus propias energías cinéticas y potenciales. En la ecuación (1.19) se muestra la energía cinética de la rueda.

$$K_r = \frac{1}{2}m_r v^2 + \frac{1}{2}I_r \omega_r^2 \quad (1.19)$$

Para obtener la energía cinética del péndulo hay que considerar el vector r_H ya que este considera el movimiento del péndulo respecto al marco de referencia de la rueda, ya que el péndulo invertido depende ahora del movimiento de la rueda. El vector r_H se muestra en la ecuación (1.20) y su derivada se observa en (1.21).

$$\mathbf{r}_H = \begin{bmatrix} (d_r + l \sin \theta) & (l \cos \theta) & 0 \end{bmatrix} \quad (1.20)$$

$$\begin{aligned} \dot{\mathbf{r}}_H &= \begin{bmatrix} (\dot{d}_r + l \dot{\theta} \cos \theta) & (-l \dot{\theta} \sin \theta) & 0 \end{bmatrix} \\ &= \begin{bmatrix} (\dot{d}_r + l \omega_p \cos \theta) & (-l \omega_p \sin \theta) & 0 \end{bmatrix} \end{aligned} \quad (1.21)$$

Con estas ecuaciones se puede describir la energía cinética del péndulo invertido mostrada en (1.22).

$$\begin{aligned} K_p &= \frac{1}{2} m_h \dot{\mathbf{r}}_h^T \dot{\mathbf{r}}_h + \frac{1}{2} I_p \omega_p^2 \\ &= \frac{1}{2} m_h (v^2 + 2 l \omega_p v \cos \theta + l^2 \omega_p^2) + \frac{1}{2} I_p \omega_p^2 \end{aligned} \quad (1.22)$$

Además, se procede a calcular la energía potencial del péndulo invertido mostrada en (1.23) y la energía potencial de la rueda, V_r , es cero ya que el marco de referencia se encuentra en el centro de masa de la rueda.

$$V_p = m_h \begin{bmatrix} 0 & g & 0 \end{bmatrix} \begin{bmatrix} d_r + l \cos \theta \\ l \cos \theta \\ 0 \end{bmatrix} = m_h g l \cos \theta \quad (1.23)$$

De esta manera se puede describir el Lagrangiano del sistema que se muestra en la ecuación (1.24).

$$\begin{aligned} L &= K_p + K_r - V_p - V_r \\ &= \frac{1}{2} m_h (v^2 + 2 l \omega_p v \cos \theta + l^2 \omega_p^2) + \frac{1}{2} I_p \omega_p^2 + \frac{1}{2} m_r v^2 + \frac{1}{2} I_r \frac{v^2}{R^2} - m_h g l \cos \theta \end{aligned} \quad (1.24)$$

Además del Lagrangiano, se establece una función de disipación de Rayleigh. Hay dos casos en dónde afecta la fricción, en la rotación del péndulo respecto a la base y también de la rueda con el suelo. Esta función se muestra en la ecuación (1.25).

$$\mathcal{R}_p = \frac{1}{2} \left(c_v v^2 + c_\omega \left(\frac{v}{R} - \omega_p \right)^2 \right) \quad (1.25)$$

Se procede a obtener las ecuaciones diferenciales del sistema siguiendo los procedimientos de las expresiones (1.26) y (1.27).

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_1} \right) - \frac{\partial L}{\partial q_1} + \frac{\partial \mathcal{R}_p}{\partial \dot{q}_1} &= \tau_1 = \\ \frac{d}{dt} \left(\frac{\partial L}{\partial v} \right) - \frac{\partial L}{\partial d_r} + \frac{\partial \mathcal{R}_p}{\partial v} &= \tau_1 \end{aligned} \quad (1.26)$$

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_2} \right) - \frac{\partial L}{\partial q_2} + \frac{\partial \mathcal{R}_p}{\partial \dot{q}_2} &= \tau_2 = \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \omega_p} \right) - \frac{\partial L}{\partial \theta} + \frac{\partial \mathcal{R}_p}{\partial \omega_p} &= \tau_2 \end{aligned} \quad (1.27)$$

Las derivadas de (1.26) se muestran en las ecuaciones (1.28) a (1.30)

$$\frac{d}{dt} \left(\frac{\partial L}{\partial v} \right) = \left(m_h + m_r + \frac{I_r}{R^2} \right) \dot{v} + l m_h (\dot{\omega}_p \cos \theta - \omega_p^2 \sin \theta) \quad (1.28)$$

$$\frac{\partial L}{\partial d_r} = 0 \quad (1.29)$$

$$\frac{\partial \mathcal{R}_p}{\partial v} = c_v v + \frac{c_\omega v}{R^2} - \frac{c_\omega \omega_p}{R} \quad (1.30)$$

Con estas derivadas se puede obtener la primera ecuación diferencial del sistema reemplazando los términos obtenidos de (1.28) a (1.30) en la ecuación (1.26), dando como resultado la expresión (1.31).

$$\left(m_h + m_r + \frac{I_r}{R^2} \right) \dot{v} + l m_h (\dot{\omega}_p \cos \theta - \omega_p^2 \sin \theta) + c_v v + \frac{c_\omega v}{R^2} - \frac{c_\omega \omega_p}{R} = \tau_1 \quad (1.31)$$

Las derivadas de (1.27) se muestran en las ecuaciones (1.32) a (1.34).

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \omega_p} \right) = I_p \dot{\omega}_p + l m_h (l \dot{\omega}_p - v \omega_p \sin \theta + \dot{v} \cos \theta) \quad (1.32)$$

$$\frac{\partial L}{\partial \theta} = l m_h (g - \omega_p v) \sin \theta \quad (1.33)$$

$$\frac{\partial \mathcal{R}_p}{\partial \omega_p} = c_\omega \omega_p - c_\omega \frac{v}{R} \quad (1.34)$$

Reemplazando los resultados de (1.32) a (1.34) en (1.27) se obtiene la segunda ecuación diferencial del sistema mostrada en (1.35).

$$(I_p + l^2 m_h) \dot{\omega}_p + l m_h (\dot{v} \cos \theta - g \sin \theta) + c_\omega \omega_p - c_\omega \frac{v}{R} = \tau_2 \quad (1.35)$$

Se observa que las ecuaciones diferenciales (1.31) y (1.35) son simultáneas y se debe realizar un proceso algebraico para despejar los términos de $\dot{v}$ y $\dot{\omega}_p$. Para realizar esto se comienza despejando el término $\dot{v}$ de (1.31) dando como resultado la expresión (1.36).

$$\dot{v} = \frac{\tau_1 - \left(c_v + \frac{c_\omega}{R^2} \right) v + \frac{c_\omega \omega_p}{R} + l m_h (\omega_p^2 \sin \theta - \dot{\omega}_p \cos \theta)}{\left(m_h + m_r + \frac{I_r}{R^2} \right)} \quad (1.36)$$

Posteriormente se sustituye (1.36) en la ecuación (1.35) obteniendo (1.37).

$$\begin{aligned}
 & (I_p + l^2 m_h) \dot{\omega}_p \\
 & + l m_h \left(\frac{\tau_1 - \left(c_v + \frac{c_\omega}{R^2}\right) v + \frac{c_\omega \omega_p}{R} + l m_h (\omega_p^2 \sin \theta - \dot{\omega}_p \cos \theta)}{(m_h + m_r + \frac{I_r}{R^2})} \right) \cos \theta \\
 & - l m_h g \sin \theta + c_\omega \omega_p - c_\omega \frac{v}{R} = \tau_2
 \end{aligned} \tag{1.37}$$

Se desarrolla y se agrupan los términos con ω_p para obtener la ecuación (1.38).

$$\begin{aligned}
 & (I_p + l^2 m_h) \dot{\omega}_p - \frac{l^2 m_h^2 \cos^2 \theta \dot{\omega}_p}{(m_h + m_r + \frac{I_r}{R^2})} \\
 & = l m_h g \sin \theta - c_\omega \omega_p + c_\omega \frac{v}{R} + \tau_2 \\
 & - \frac{l m_h \cos \theta \tau_1 + l m_h \cos \theta \left(c_v + \frac{c_\omega}{R^2}\right) v - l m_h \cos \theta \frac{c_\omega \omega_p}{R} - l^2 m_h^2 \omega_p^2 \sin \theta \cos \theta}{(m_h + m_r + \frac{I_r}{R^2})}
 \end{aligned} \tag{1.38}$$

Posteriormente se puede despejar ω_p para obtener las ecuaciones (1.39) y (1.40).

$$\dot{\omega}_p = \frac{\tau_2 - c_\omega \omega_p + l m_h (g \sin(\theta) - \beta) + \frac{c_\omega v}{R}}{I_p + l^2 m_h - \frac{l^2 m_h^2 \cos^2(\theta)}{m_h + m_r + \frac{I_r}{R^2}}} \tag{1.39}$$

donde

$$\begin{aligned}
 \beta &= \frac{\cos(\theta) \left(\tau_1 - v \left(c_v + \frac{c_\omega}{R^2}\right)\right)}{m_h + m_r + \frac{I_r}{R^2}} \\
 &+ \frac{\cos(\theta) \left(\frac{c_\omega \omega_p}{R} + l m_h \omega_p^2 \sin(\theta)\right)}{m_h + m_r + \frac{I_r}{R^2}}.
 \end{aligned} \tag{1.40}$$

Se realiza un proceso similar para obtener la ecuación de $\dot{v}$, primero se despeja ω_p de la expresión (1.35) obteniendo (1.41).

$$\dot{\omega}_p = \frac{\tau_2 + c_\omega \frac{v}{R} - c_\omega \omega_p - l m_h (\dot{v} \cos \theta - g \sin \theta)}{(I_p + l^2 m_h)} \tag{1.41}$$

Se procede a sustituir (1.41) en (1.31), de esta manera se obtiene la ecuación (1.42).

$$\begin{aligned}
 & \left(m_h + m_r + \frac{I_r}{R^2}\right) \dot{v} \\
 & + l m_h \left(\left(\frac{\tau_2 + c_\omega \frac{v}{R} - c_\omega \omega_p - l m_h (\dot{v} \cos \theta - g \sin \theta)}{(I_p + l^2 m_h)} \right) \cos \theta - \omega_p^2 \sin \theta \right) \\
 & + c_v v + \frac{c_\omega v}{R^2} - \frac{c_\omega \omega_p}{R} = \tau_1
 \end{aligned} \tag{1.42}$$

Se desarrolla y agrupan los términos con $\dot{v}$ y se obtiene la expresión (1.43).

$$\begin{aligned}
 & \left(m_h + m_r + \frac{I_r}{R^2} \right) \dot{v} - \frac{l^2 m_h^2 \cos^2 \theta \dot{v}}{I_p + l^2 m_h} \\
 & = \tau_1 - \left(c_v + \frac{c_\omega}{R^2} \right) v + \frac{c_\omega}{R} \omega_p + l m_h \omega_p^2 \sin \theta \\
 & - \frac{l m_h \cos \theta \tau_2 - l m_h \cos \theta \left(\frac{c_\omega}{R} \right) v + l m_h \cos \theta c_\omega \omega_p - l^2 m_h^2 g \sin \theta \cos \theta}{I_p + l^2 m_h}
 \end{aligned} \tag{1.43}$$

Para continuar se despeja $\dot{v}$ para obtener las ecuaciones (1.44) y (1.45).

$$\begin{aligned}
 \dot{v} & = \frac{\tau_1 - v \left(c_v + \frac{c_\omega}{R^2} \right)}{m_h + m_r + \frac{I_r}{R^2} - \frac{l^2 m_h^2 \cos(\theta)^2}{m_h l^2 + I_p}} \\
 & + \frac{l m_h \gamma}{m_h + m_r + \frac{I_r}{R^2} - \frac{l^2 m_h^2 \cos(\theta)^2}{m_h l^2 + I_p}} \\
 & + \frac{\frac{c_\omega \omega_p}{R}}{m_h + m_r + \frac{I_r}{R^2} - \frac{l^2 m_h^2 \cos(\theta)^2}{m_h l^2 + I_p}}
 \end{aligned} \tag{1.44}$$

$$\gamma = \omega_p^2 \sin \theta - \frac{\cos(\theta) \left(\tau_2 - c_\omega \omega_p + \frac{c_\omega v}{R} + l g m_h \sin \theta \right)}{m_h l^2 + I_p} \tag{1.45}$$

Con las ecuaciones del sistema ya despejadas se procede a obtener la representación en espacio de estados del sistema. Como primer paso se establecen los estados en la expresión (1.46).

$$\begin{aligned}
 x_1 & = d_r \\
 x_2 & = \dot{x}_1 = v \\
 x_3 & = \theta \\
 x_4 & = \dot{x}_3 = \omega_p
 \end{aligned} \tag{1.46}$$

Asimismo, es necesario recalcar que las entradas al sistema se toman de la forma mostrada en (1.47).

$$\begin{aligned}
 \tau_1 & = \frac{\tau}{R} \\
 \tau_2 & = 0
 \end{aligned} \tag{1.47}$$

τ es el par generado por los motores del robot y $\tau_2 = 0$ ya que no hay un motor directamente acoplado al péndulo invertido.

Con la entrada y los estados definidos se puede obtener las cuatro ecuaciones correspondientes a la representación en espacio de estados del robot de autobalance. Esta forma del sistema se muestra en las ecuaciones (1.48) a (1.51).

$$\boxed{\dot{x}_1 = x_2} \quad (1.48)$$

$$\boxed{\begin{aligned} \dot{x}_2 = & \frac{\frac{\tau}{R} - x_2 \left(c_v + \frac{c_\omega}{R^2} \right)}{m_h + m_r + \frac{I_r}{R^2} - \frac{l^2 m_h^2 \cos(x_3)^2}{m_h l^2 + I_p}} \\ & + \frac{l m_h \gamma}{m_h + m_r + \frac{I_r}{R^2} - \frac{l^2 m_h^2 \cos(x_3)^2}{m_h l^2 + I_p}} \\ & + \frac{\frac{c_\omega x_4}{R}}{m_h + m_r + \frac{I_r}{R^2} - \frac{l^2 m_h^2 \cos(x_3)^2}{m_h l^2 + I_p}} \end{aligned}} \quad (1.49)$$

$$\boxed{\dot{x}_3 = x_4} \quad (1.50)$$

$$\boxed{\dot{x}_4 = \frac{-c_\omega x_4 + l m_h (g \sin(x_3) - \beta) + \frac{c_\omega x_2}{R}}{I_p + l^2 m_h - \frac{l^2 m_h^2 \cos(x_3)^2}{m_h + m_r + \frac{I_r}{R^2}}}} \quad (1.51)$$

De esta forma los términos γ y β , que aparecen en las ecuaciones, son definidas por las ecuaciones (1.52) y (1.53) respectivamente.

$$\gamma = x_4^2 \sin(x_3) - \frac{\cos(x_3) \left(-c_\omega x_4 + \frac{c_\omega x_2}{R} + l g m_h \sin(x_3) \right)}{m_h l^2 + I_p} \quad (1.52)$$

$$\beta = \frac{\cos(x_3) \left(\frac{\tau}{R} - x_2 \left(c_v + \frac{c_\omega}{R^2} \right) \right)}{m_h + m_r + \frac{I_r}{R^2}} + \frac{\cos(x_3) \left(\frac{c_\omega x_4}{R} + l m_h x_4^2 \sin(x_3) \right)}{m_h + m_r + \frac{I_r}{R^2}} \quad (1.53)$$

1.3. Modelo mediante el principio de trabajo virtual

Para esta sección, se usa la siguiente nomenclatura. Ya que el objetivo es obtener el mismo modelo de la subsección 1.2.3, se reutilizan algunas variables.

- f_i indica una fuerza aplicada en un desplazamiento virtual.
- δr_i indica un desplazamiento virtual, infinitesimal y reversible.
- $\boldsymbol{\tau}_i$ es el vector de fuerzas o pares generalizados asociados a las coordenadas generalizadas del robot. Representa los esfuerzos equivalentes en el espacio articular que producen el mismo trabajo virtual que las fuerzas aplicadas en el espacio cartesiano.
- $\boldsymbol{F}_i$ es el wrench (vector de fuerzas generalizadas en el espacio cartesiano) aplicado al cuerpo i .
- $\mathbf{J}_i$ es el jacobiano geométrico del cuerpo i , que relaciona las velocidades generalizadas del sistema con la velocidad lineal y angular del cuerpo.
- m_i es la masa del cuerpo i .
- $\mathbf{r}_{gi}$ es el vector de la posición del centro de masa del cuerpo i .
- $\ddot{\mathbf{r}}_{gi}$ es el vector de aceleración lineal del centro de masa del cuerpo i .
- $\mathbf{I}_i$ es el tensor de inercia del cuerpo i .
- $\boldsymbol{\omega}_i$ es el vector de velocidad angular del cuerpo i .
- $\dot{\boldsymbol{\omega}}_i$ es el vector de aceleración angular del cuerpo i .
- $\mathbf{J}_v^T$ es el jacobiano lineal del cuerpo i .
- $\mathbf{J}_\omega^T$ es el jacobiano angular del cuerpo i .
- $\boldsymbol{\tau}_{\text{rot}}$ es el momento de inercia del cuerpo i .
- $\mathbf{q}$ es el vector de coordenadas generalizadas.
- $\dot{\mathbf{q}}$ es el vector de la primera derivada de las coordenadas generalizadas.
- $\ddot{\mathbf{q}}$ es el vector de la segunda derivada de las coordenadas generalizadas.
- g es la aceleración de la gravedad.
- d_r es la distancia del centro de las ruedas respecto al marco de referencia.
- θ es el ángulo de inclinación del robot respecto al eje vertical Y .
- v es la velocidad de las ruedas del robot.

- $\dot{v}$ es la aceleración de las ruedas del robot.
- $\dot{\omega}_p$ es la aceleración angular de la inclinación del robot.
- 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 .
- $\mathbf{r}_H$ es la distancia del centro de masa del robot al marco de referencia.
- $\mathbf{r}_L$ 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_p es la velocidad angular del cuerpo del robot.
- c_v es el coeficiente de fricción de las ruedas respecto al suelo en el que se desplaza el robot.
- c_ω es el coeficiente de fricción del movimiento de rotación del péndulo respecto a la unión con las ruedas.
- $\mathbf{J}_{vL}$ es el jacobiano lineal que corresponde a las ruedas del robot.
- $\mathbf{J}_{vH}$ es el jacobiano lineal que corresponde al cuerpo del robot.
- $\mathbf{a}_L$ es el vector de las aceleraciones de las ruedas del robot.
- $\mathbf{a}_H$ es el vector de las aceleraciones del cuerpo del robot.
- ω_L es la velocidad angular de las ruedas.
- ω_H es la velocidad angular del cuerpo del robot.
- α_L es la aceleración angular de las ruedas del robot.
- α_H es la aceleración angular del cuerpo del robot.
- $\mathbf{f}_L$ es el vector de fuerzas inerciales en las ruedas del robot.

- $\mathbf{f}_H$ es el vector de fuerzas inerciales en el cuerpo del robot.
- τ_L es el par inercial en las ruedas del robot.
- τ_H es el par inercial del cuerpo del robot.
- $\mathbf{J}_{\omega L}$ es el jacobiano angular de las ruedas del robot.
- $\mathbf{J}_{\omega H}$ es el jacobiano angular del cuerpo del robot.
- $\mathbf{J}_L$ es el jacobiano, considerando la parte lineal y angular, de las ruedas del robot.
- $\mathbf{J}_H$ es el jacobiano, considerando la parte lineal y angular, del cuerpo del robot.
- $\mathbf{W}_L$ es el "wrench" de las ruedas del robot.
- $\mathbf{W}_H$ es el "wrench" del cuerpo del robot.
- $\boldsymbol{\tau}_{inercial}$ es el vector de fuerzas o pares inerciales del robot.
- f_{gH} es la fuerza gravitacional que actúa en el cuerpo del robot.
- f_v es la fuerza debida a la fricción presente en el robot.
- f_ω es el par debido a la fricción presente en el robot.
- $\boldsymbol{\tau}_{disipativa}$ es el vector de fuerzas y pares debido a la fricción en el robot.
- τ_1 corresponde a la fuerza aplicada al robot, debido al par de los motores, siendo igual a $\frac{\tau}{R}$.
- τ es el par generado por los dos motores del robot.
- τ_2 corresponde a la entrada de la ecuación de la sumatoria de pares, siendo $\tau_2 = 0$.

Una de las técnicas actuales para el modelado de robots es mediante la técnica de trabajo virtual. Como se puede observar en [24] el teorema del trabajo virtual se define por la ecuación (1.54), indicando que el trabajo realizado por una fuerza aplicada en un desplazamiento virtual, infinitesimal y reversible es cero.

$$\sum_{i=1}^N f_i \delta r_i = 0 \quad (1.54)$$

Este principio se puede aplicar a robótica mediante la ecuación (1.55) usando de referencia [26] y [27].

$$\boldsymbol{\tau}_i = \sum_{i=1}^N \mathbf{J}_i^T \mathbf{F}_i \quad (1.55)$$

donde

$$\mathbf{F}_i = \begin{bmatrix} m_i \ddot{\mathbf{r}}_{gi} \\ \mathbf{I}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times (\mathbf{I}_i \boldsymbol{\omega}_i) \end{bmatrix} \quad (1.56)$$

De manera más general el principio de trabajo virtual se puede expresar como se muestra en la ecuación (1.57).

$$\boldsymbol{\tau}_i = \sum_{i=1}^N \begin{bmatrix} \mathbf{J}_v^T & \mathbf{J}_\omega^T \end{bmatrix} \begin{bmatrix} m_i \ddot{\mathbf{r}}_{gi} \\ \boldsymbol{\tau}_{\text{rot}} \end{bmatrix} \quad (1.57)$$

Para abordar este enfoque es necesario analizar el diagrama de cuerpo libre que se muestra en la figura (1.8).

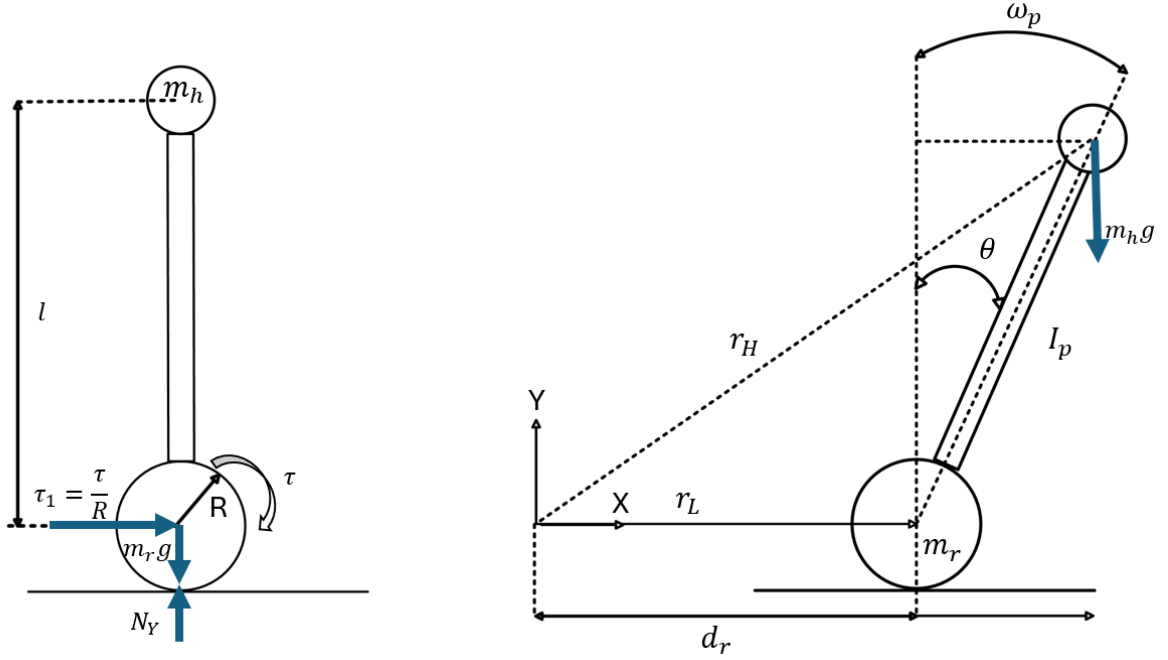


Figura 1.8. Esquema de fuerzas del péndulo invertido.

donde:

- $m_r g$ es el peso de las ruedas.
- $m_h g$ es el peso del cuerpo del robot.

- N_Y es la fuerza normal de las ruedas del robot respecto al suelo. No aparece en la formulación final al no contribuir en el movimiento dinámico del robot.

En el diagrama de cuerpo libre clásico únicamente se consideran fuerzas y pares externos reales. Las fuerzas inerciales y los pares inerciales no se representan explícitamente, ya que se incorporan posteriormente en la formulación dinámica mediante el principio de trabajo virtual.

De esta manera el procedimiento para obtener el modelo dinámico mediante trabajo virtual consiste en los productos de las matrices jacobianas del robot con respecto a la sumatoria de fuerzas y pares presentes en el sistema. Este procedimiento se aplica al robot de autobalance definiendo las coordenadas generalizadas mostradas en las ecuaciones (1.58), (1.59) y (1.60). Es importante remarcar que el uso de esta técnica de modelado en este tipo de robots no es común tal y como se muestra en el estudio de modelos realizado en el trabajo [1], por esto mismo es una actividad de interés en el desarrollo de esta tesis.

$$\mathbf{q} = \begin{bmatrix} d_r \\ \theta \end{bmatrix} \quad (1.58)$$

$$\dot{\mathbf{q}} = \begin{bmatrix} v \\ \omega_p \end{bmatrix} \quad (1.59)$$

$$\ddot{\mathbf{q}} = \begin{bmatrix} \dot{v} \\ \dot{\omega}_p \end{bmatrix} \quad (1.60)$$

Posteriormente se definen los vectores de posición para cada cuerpo del sistema, mostrados en las ecuaciones (1.61) y (1.62).

$$\mathbf{r}_L = \begin{bmatrix} d_r \\ 0 \end{bmatrix} \quad (1.61)$$

$$\mathbf{r}_H = \begin{bmatrix} d_r + l \sin(\theta) \\ l \cos(\theta) \end{bmatrix} \quad (1.62)$$

Posteriormente se obtiene las matrices jacobianas de los términos de las velocidades lineales, mostradas en las ecuaciones (1.63) y (1.64).

$$\mathbf{J}_{vL} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad (1.63)$$

$$\mathbf{J}_{vH} = \begin{bmatrix} 1 & l \cos(\theta) \\ 0 & -l \sin(\theta) \end{bmatrix} \quad (1.64)$$

Para los siguientes cálculos son necesarias las aceleraciones de cada cuerpo del sistema, éstas se presentan en las ecuaciones (1.65) y (1.66).

$$\mathbf{a}_L = \begin{bmatrix} \dot{v} \\ 0 \end{bmatrix} \quad (1.65)$$

$$\mathbf{a}_H = \begin{bmatrix} -l \sin(\theta) \omega_p^2 + \dot{v} + \dot{\omega}_p l \cos(\theta) \\ -l (\cos(\theta) \omega_p^2 + \dot{\omega}_p \sin(\theta)) \end{bmatrix} \quad (1.66)$$

Se procede a establecer las velocidades y aceleraciones angulares a ocupar, como se muestra en las ecuaciones (1.67) a (1.70).

$$\omega_L = v/R \quad (1.67)$$

$$\omega_H = \omega_p \quad (1.68)$$

$$\alpha_L = \frac{\dot{v}}{R} \quad (1.69)$$

$$\alpha_H = \dot{\omega}_p \quad (1.70)$$

Se procede a obtener las fuerzas y pares inerciales con los cálculos mostrados en las ecuaciones (1.71) a (1.74).

$$\mathbf{f}_L = m_r \mathbf{a}_L \quad (1.71)$$

$$\mathbf{f}_H = m_h \mathbf{a}_H \quad (1.72)$$

$$\tau_L = I_r \alpha_L \quad (1.73)$$

$$\tau_H = I_p \alpha_H \quad (1.74)$$

Posteriormente se hacen los cálculos correspondientes a las matrices jacobianas angulares mostrados en las ecuaciones (1.75) y (1.76) para poder calcular las matrices jacobianas totales como se muestra en las ecuaciones (1.77) y (1.78).

$$\mathbf{J}_{\omega L} = \begin{bmatrix} \frac{1}{R} & 0 \end{bmatrix} \quad (1.75)$$

$$\mathbf{J}_{\omega H} = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad (1.76)$$

$$\mathbf{J}_L = \begin{bmatrix} \mathbf{J}_{vL} \\ \mathbf{J}_{\omega L} \end{bmatrix} \quad (1.77)$$

$$\mathbf{J}_H = \begin{bmatrix} \mathbf{J}_{vH} \\ \mathbf{J}_{\omega H} \end{bmatrix} \quad (1.78)$$

Asimismo, es necesario calcular las llaves que corresponden a las fuerzas y pares en cada cuerpo del sistema. Esto se muestra en las ecuaciones (1.79) y (1.80).

$$\mathbf{W}_L = \begin{bmatrix} \mathbf{f}_L \\ \tau_L \end{bmatrix} \quad (1.79)$$

$$\mathbf{W}_H = \begin{bmatrix} \mathbf{f}_H \\ \tau_H \end{bmatrix} \quad (1.80)$$

Con las ecuaciones ya descritas se puede calcular el par inercial de todo el sistema que se muestra en la ecuación (1.81).

$$\begin{aligned} \tau_{inercial} &= \mathbf{J}_L^T \mathbf{W}_L + \mathbf{J}_H^T \mathbf{W}_H \\ &= \begin{bmatrix} m_h (-l \sin(\theta) \omega_p^2 + \dot{v} + \dot{\omega}_p l \cos(\theta)) + \dot{v} m_r + \frac{I_r \dot{v}}{R^2} \\ \dot{\omega}_p m_h l^2 + \dot{v} m_h \cos(\theta) l + I_p \dot{\omega}_p \end{bmatrix} \end{aligned} \quad (1.81)$$

La siguiente parte del modelo corresponde a las fuerzas provocadas por la gravedad, en este caso solo el cuerpo del péndulo invertido se ve afectado. De este modo la ecuación de la fuerza gravitacional se muestra en (1.82). Y la contribución de la fuerza de gravedad al modelo se expresa en la ecuación (1.83).

$$\mathbf{f}_{gH} = m_h g \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (1.82)$$

$$\tau_{gravitacional} = \mathbf{J}_{vH}^T \mathbf{f}_{gH} = \begin{bmatrix} 0 \\ -g l m_h \sin(\theta) \end{bmatrix} \quad (1.83)$$

Finalmente, la última parte del modelo corresponde a los términos disipativos de energía que previamente se calcularon con una función de disipación de Rayleigh. Estos términos se muestran en las ecuaciones (1.84) y (1.85). Por lo tanto, la contribución de este tipo de fuerzas al modelo del robot se expresa en la ecuación (1.86).

$$f_v = c_v v - \frac{c_\omega \left(\omega_p - \frac{v}{R} \right)}{R} \quad (1.84)$$

$$f_\omega = c_\omega \left(\omega_p - \frac{v}{R} \right) \quad (1.85)$$

$$\boldsymbol{\tau}_{disipativa} = \begin{bmatrix} f_v \\ f_\omega \end{bmatrix} \quad (1.86)$$

De esta manera el modelo del robot de autobalance se obtiene realizando la ecuación (1.87).

$$\boldsymbol{\tau}_i = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \left[\boldsymbol{\tau}_{inercial} + \boldsymbol{\tau}_{gravitacional} + \boldsymbol{\tau}_{disipativa} \right] \quad (1.87)$$

Al realizar estas operaciones se obtienen las ecuaciones (1.88) y (1.89).

$$\begin{aligned} \tau_1 = m_h \left(-l \sin(\theta) \omega_p^2 + \dot{v} + \dot{\omega}_p l \cos(\theta) \right) + c_v v + \dot{v} m_r \\ - \frac{c_\omega \left(\omega_p - \frac{v}{R} \right)}{R} + \frac{I_r \dot{v}}{R^2} \end{aligned} \quad (1.88)$$

$$\begin{aligned} \tau_2 = I_p \dot{\omega}_p + c_\omega \left(\omega_p - \frac{v}{R} \right) + \dot{\omega}_p l^2 m_h + \dot{v} l m_h \cos(\theta) \\ - g l m_h \sin(\theta) \end{aligned} \quad (1.89)$$

Se debe tener en cuenta que $\tau_1 = \frac{\tau}{R}$ siendo la fuerza que aportan los motores del robot y $\tau_2 = 0$ ya que el péndulo invertido no tiene un motor directamente acoplado que permita controlar el ángulo de este. Aplicando esto a las ecuaciones (1.88) y (1.89) y acomodando los términos se tiene las ecuaciones (1.90) y (1.91). Se puede observar que son exactamente las mismas ecuaciones obtenidas mediante el enfoque de Euler-Lagrange corroborando que el procedimiento fue correcto. Y por ende se obtendrá el mismo modelo en espacio de estados descritos en las ecuaciones (1.48) a (1.51).

$$\begin{aligned} \frac{\tau}{R} = \left(m_h + m_r + \frac{I_r}{R^2} \right) \dot{v} + l m_h (\dot{\omega}_p \cos \theta - \omega_p^2 \sin \theta) \\ + c_v v + \frac{c_\omega v}{R^2} - \frac{c_\omega \omega_p}{R} = \tau_1 \end{aligned} \quad (1.90)$$

$$0 = (I_p + l^2 m_h) \dot{\omega}_p + l m_h (\dot{v} \cos \theta - g \sin \theta) + c_\omega \omega_p - c_\omega \frac{v}{R} \quad (1.91)$$

1.4. Comparación de técnicas de modelado

Aunque el principio de trabajo virtual y el enfoque de Euler-Lagrange son teóricamente equivalentes (ambos derivan de las leyes de Newton y del principio d'Alembert), hay diferencias prácticas importantes en su uso, sobre todo en robótica. En la tabla (1.1) se analiza la naturaleza conceptual de cada enfoque.

Aspecto	Euler-Lagrange	Trabajo virtual
Fundamento	Se basa en energía (cinética y potencial).	Se basa en fuerzas y movimientos virtuales.
Interpretación física	Usa derivadas del lagrangiano para obtener ecuaciones diferenciales.	Usa el equilibrio entre fuerzas reales e inerciales proyectadas sobre movimientos virtuales.
Magnitudes usadas	Energías escalares.	Fuerzas, pares, y velocidades lineales - angulares (vectores).

Tabla 1.1. Naturaleza de los enfoques de Euler-Lagrange y Trabajo Virtual.

En otras palabras: Euler-Lagrange trabaja con energías y el Trabajo Virtual directamente con fuerzas y pares.

El método de Trabajo Virtual tiene una serie de ventajas:

1. Más directo para sistemas con restricciones. En robots paralelos o mecanismos con restricciones cinemáticas, el trabajo virtual permite trabajar directamente en el espacio cartesiano, sin necesidad de resolver las restricciones antes de derivar las ecuaciones. Esto evita la complejidad de introducir multiplicadores de Lagrange. Ejemplo: en un robot Delta o "Stewart Platform", el trabajo virtual permite escribir las fuerzas en los actuadores directamente desde el Jacobiano.
2. Evita la formulación energética completa. No se necesita calcular la energía cinética y potencial de todos los cuerpos; basta con conocer: Las fuerzas inerciales (masa $\times$ aceleración), las fuerzas externas (gravedad, actuadores, fricción) y los Jacobianos que relacionan fuerzas con coordenadas generalizadas. Esto hace el método más algebraico y modular.
3. Claridad en la relación fuerzas - pares. El trabajo virtual da una conexión clara entre las fuerzas en el espacio cartesiano (por ejemplo, fuerzas de gravedad o de contacto) y los pares en las articulaciones. Esto es natural para la robótica, donde el control suele actuar sobre pares articulares.

4. Extensible a análisis estáticos o cuasi-estáticos. El mismo principio se usa para obtener modelos puramente estáticos, sin derivar el Lagrangiano, lo cual es útil en diseño mecánico o control de robots manipuladores cuando las velocidades son bajas o despreciables.

El enfoque de trabajo virtual tiene algunas desventajas que se analizan a continuación:

1. Menos intuitivo para principiantes. El concepto de “movimiento virtual” no es tan claro como energía.
2. Requiere conocimiento del Jacobiano completo. Se debe conocer tanto el Jacobiano lineal como el angular de cada cuerpo.
3. Puede ser tedioso para sistemas sin simetría. Si el sistema no tiene un marco bien definido o hay fricción compleja, el trabajo virtual puede ser algebraicamente más pesado.

El principio del trabajo virtual es más general, más geométrico y adecuado para robots con estructuras cerradas o móviles complejas. El enfoque de Euler–Lagrange es más sistemático y fácil de automatizar cuando el sistema es abierto y simple (como manipuladores seriales o péndulos invertidos).

1.5. Verificación mediante simulación del modelo obtenido

En el anexo (A) se muestra el procedimiento para obtener los parámetros del robot Elegoo Tumbler. Estos se muestran en la ecuación (1.92) y son utilizados para las simulaciones y los posteriores diseños de los controladores.

$$\begin{aligned}
 l &= 0.0413m \\
 m_h &= 0.674kg \\
 I_p &= 240.307 \times 10^{-5} kg m^2 \\
 m_r &= 0.062Kg \\
 I_r &= 3.3759 \times 10^{-5} kg m^2 \\
 R &= 0.033m \\
 c_\omega &= 2.6 \times 10^{-3} \frac{Nms}{rad} \\
 c_v &= 2 \frac{Ns}{m}
 \end{aligned} \tag{1.92}$$

Con estos parámetros se obtiene la respuesta del sistema en lazo abierto del robot Elegoo. Además, el robot comienza en una posición en vertical. Este resultado se obtiene en la figura (1.9).

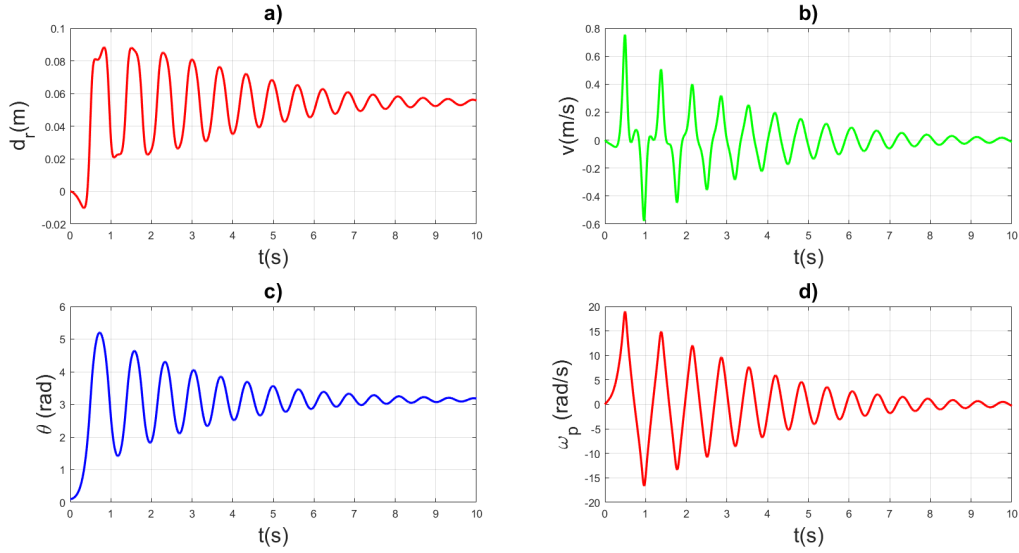


Figura 1.9. Resultados de la simulación del robot Elegoo Tumbler: a) Distancia recorrida. b) Velocidad lineal. c) Ángulo de inclinación. d) Velocidad angular.

1.6. Cinemática de un robot móvil de arquitectura diferencial

En el anexo (B) se muestra la teoría de sistemas no holonómicos como lo son los robots móviles del tipo diferencial. Esta teoría indica las condiciones intrínsecas que existen en los robots diferenciales, y es necesario comprender para desarrollar el modelado y control de seguimiento de trayectoria. Para poder analizar la cinemática y restricciones de un robot no holonómico, que en este caso es el robot móvil de arquitectura diferencial que se muestra en la figura (1.10). Asimismo, se usa de referencia [28] y [29].

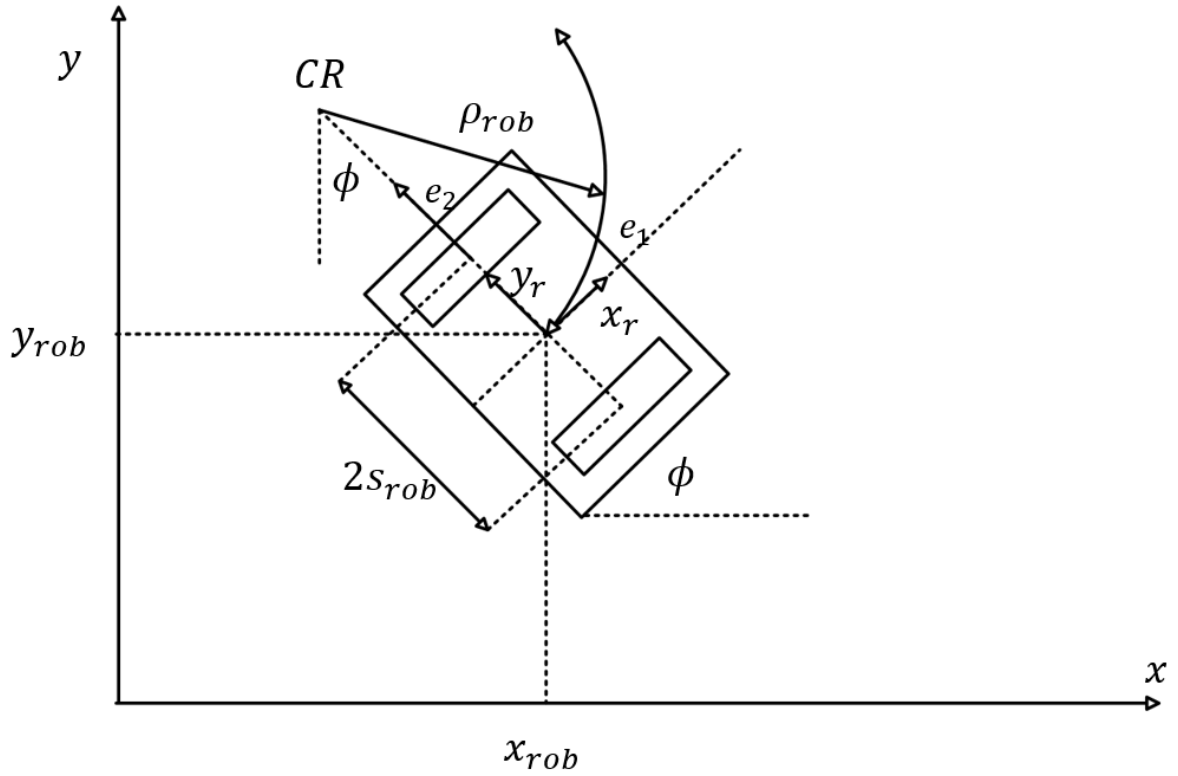


Figura 1.10. Robot móvil de arquitectura diferencial.

Para esta sección se usa la siguiente nomenclatura:

- ρ_{rob} es el radio de rotación del centro de rotación CR al centro de masa del robot.
- s_{rob} es la distancia del centro de masa del robot al centro de la rueda.
- $\rho_{i_{rob}}$ es el radio de rotación del centro de rotación CR a la rueda izquierda.
- $\rho_{d_{rob}}$ es el radio de rotación del centro de rotación CR a la rueda derecha.
- $v_{i_{rob}}$ es la velocidad lineal de la rueda izquierda.
- $v_{d_{rob}}$ es la velocidad lineal de la rueda derecha.
- ω_{rob} es la velocidad angular del robot, considerando la rotación sobre su propio eje.
- v_{rob} es la velocidad lineal del robot.
- $\omega_{i_{rob}}$ es la velocidad angular de la rueda izquierda.
- $\omega_{d_{rob}}$ es la velocidad angular de la rueda derecha.
- R es el radio de las ruedas.

- x_{rob} es la coordenada en el eje x del centro de masa del robot.
- y_{rob} es la coordenada en el eje y del centro de masa del robot.
- $\dot{x}_{mrob}$ es la velocidad en el eje x_r respecto al marco de referencia móvil.
- $\dot{y}_{mrob}$ es la velocidad en el eje y_r respecto al marco de referencia móvil.
- $\dot{x}_{rob}$ es la velocidad en el eje x respecto al marco de referencia fijo.
- $\dot{y}_{rob}$ es la velocidad en el eje y respecto al marco de referencia fijo.

Se considera al robot girando alrededor de un centro de rotación instantáneo (CR). Ambas ruedas tienen el mismo centro de rotación, sin embargo, las distancias de cada rueda al centro de rotación serán distintas, ρ_{irob} para la rueda izquierda y ρ_{drob} para la rueda derecha. Estos radios se muestran en las ecuaciones (1.93) y (1.94).

$$\rho_{irob} = \rho_{rob} - s_{rob} \quad (1.93)$$

$$\rho_{drob} = \rho_{rob} + s_{rob} \quad (1.94)$$

Con las distancias de cada rueda al centro de rotación se puede obtener las velocidades de cada una de las ruedas. La velocidad de la rueda izquierda se denota por v_{irob} y se muestra en la ecuación (1.95). La velocidad de la rueda derecha se denota por v_{drob} se muestra en la ecuación (1.96).

$$v_{irob} = \omega_{rob}(\rho_{rob} - s_{rob}) \quad (1.95)$$

$$v_{drob} = \omega_{rob}(\rho_{rob} + s_{rob}) \quad (1.96)$$

Para encontrar la velocidad angular del robot respecto al centro de rotación ω_{rob} se obtiene restando las ecuaciones (1.95) y (1.96). La velocidad angular se muestra en (1.97).

$$\omega_{rob} = \frac{v_{irob} - v_{drob}}{2s_{rob}} \quad (1.97)$$

Para obtener el radio de giro ρ_{rob} del robot se divide la suma de las velocidades de ambas ruedas entre la resta de estas. Esto se muestra en la ecuación (1.98).

$$\omega_{rob} = \frac{v_{irob} + v_{drob}}{v_{irob} - v_{drob}} s_{rob} \quad (1.98)$$

Asimismo, se obtiene la velocidad instantánea del centro del vehículo como se muestra en la ecuación (1.99).

$$v_{rob} = \omega_{rob} \rho_{rob} = \frac{1}{2}(v_{i rob} + v_{d rob}) \quad (1.99)$$

Con esta información se puede obtener las velocidades del robot diferencial respecto al marco móvil. Este marco móvil tiene ejes x_r e y_r paralelos a los vectores unitarios e_1 y e_2 respectivamente. Estas velocidades se muestran de forma matricial en la ecuación (1.100).

$$\begin{bmatrix} \dot{x}_{m rob} \\ \dot{y}_{m rob} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ 0 & 0 \\ \frac{R}{2s_{rob}} & \frac{-R}{2s_{rob}} \end{bmatrix} \begin{bmatrix} \omega_{i rob} \\ \omega_{d rob} \end{bmatrix} \quad (1.100)$$

En la ecuación (1.100) el término R es el radio de cada una de las ruedas motoras. Finalmente se obtiene el modelo cinemático respecto al marco mundial $x - y$, esto se muestra en la ecuación (1.101).

$$\boxed{\begin{bmatrix} \dot{x}_{rob} \\ \dot{y}_{rob} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \phi & 0 \\ \sin \phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{rob} \\ \omega_{rob} \end{bmatrix}} \quad (1.101)$$

Para verificar el modelo cinemático del robot móvil diferencial se realiza una simulación de seguimiento de trayectoria. Esta simulación se muestra en las figuras (1.11) y (1.12). Se puede observar que los resultados obtenidos del modelo cinemático son coherentes.

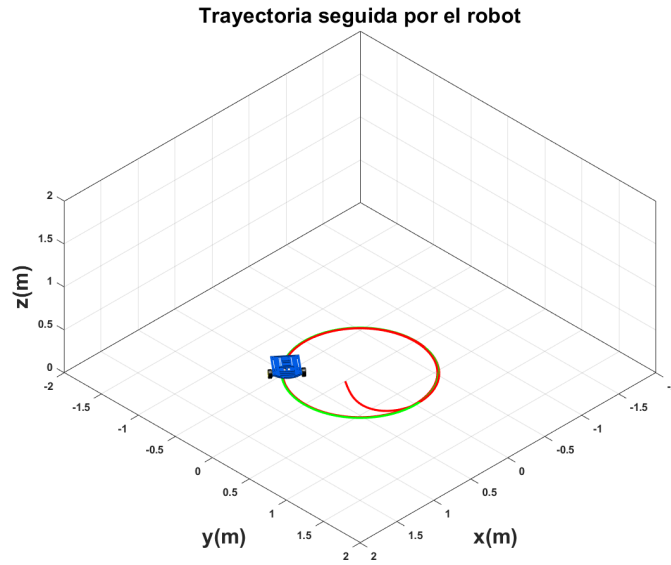


Figura 1.11. Seguimiento de trayectoria del robot móvil de arquitectura diferencial.

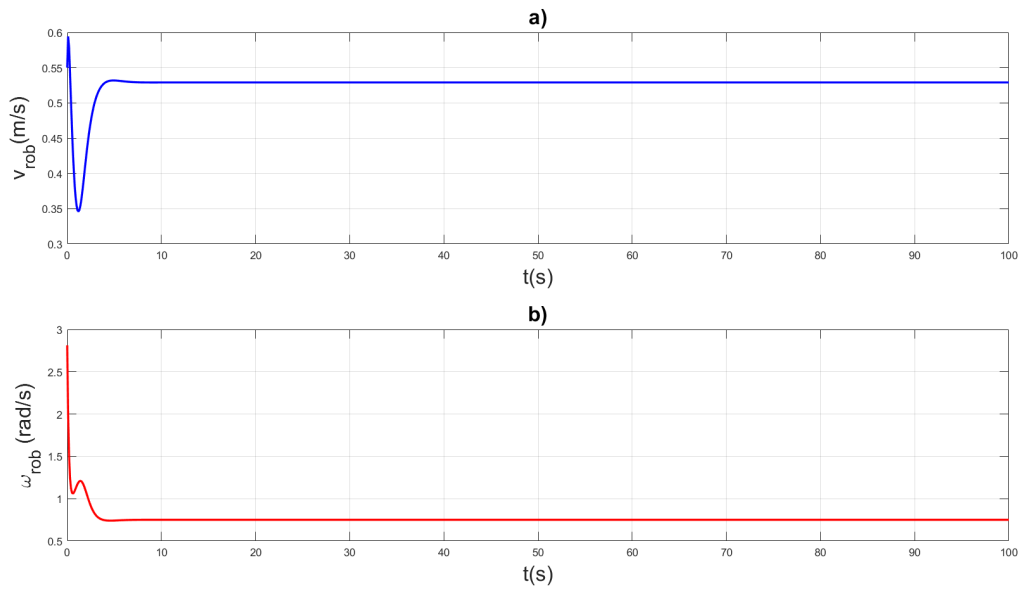


Figura 1.12. Velocidades del robot móvil de arquitectura diferencial: a) Velocidad lineal. b) velocidad angular.

DESARROLLO DE ALGORITMOS DE CONTROL PARA EL ROBOT MÓVIL DE DOS RUEDAS CON AUTOBALANCE

2.1. Introducción

Este capítulo tiene como principal objetivo mostrar el desarrollo de los algoritmos de control para el robot de dos ruedas con autobalance, teniendo dos controladores diferentes, uno para el equilibrio y otro para el seguimiento de trayectoria.

- En la sección 2.2, se establece el proceso para obtener un modelo linealizado del modelo no lineal obtenido en la subsección 1.2.3 del capítulo 1. Este apartado tiene las siguientes subsecciones:
 - En la subsección 2.2.1, se muestra el procedimiento para obtener las derivadas de la ecuación de estado 2, necesarias para obtener el modelo linealizado.
 - En la subsección 2.2.2, se muestra el procedimiento para obtener las derivadas de la ecuación de estado 4, necesarias para obtener el modelo linealizado.
 - En la subsección 2.2.3, se muestra el procedimiento para obtener las derivadas de las ecuaciones respecto a la entrada del sistema.
- En la sección 2.3, se expone el procedimiento para diseñar el control de equilibrio del robot. Este apartado tiene las siguientes subsecciones:
 - En la subsección 2.3.1, se desarrolla el controlador de equilibrio en tiempo continuo para el robot.

- En la subsección 2.3.2, se desarrolla el controlador de equilibrio en tiempo discreto para el robot. Este controlador es el que se implementa en el microcontrolador del robot.
- En la sección 2.4, se muestra la implementación del controlador de equilibrio mostrado en 2.3.2 en el microcontrolador del robot.
- En la sección 2.5, se desarrolla el controlador para el seguimiento de trayectoria. Se divide en las siguientes subsecciones:
 - En la subsección 2.5.1, se establece el desarrollo del controlador basado en técnicas de Lyapunov para el seguimiento de trayectoria. Para este procedimiento se usa el modelo obtenido en la sección 1.6 del capítulo 1.
 - En la subsección 2.5.2, se muestra el desarrollo de la odometría para poder obtener las coordenadas y orientación del robot.

En la sección 2.6, se expone el procedimiento para poder implementar los controladores de equilibrio y de seguimiento de trayectoria. Asimismo, se compone de las siguientes subsecciones:

- En la subsección 2.6.1, se muestra el procedimiento para implementar la odometría en el microcontrolador del robot.
- En la subsección 2.6.2, se muestra el procedimiento para implementar la combinación del control de equilibrio con el de seguimiento de trayectoria.

2.2. Linealización del modelo del robot móvil

Una vez que se obtienen los parámetros del robot, se puede continuar con la linealización del modelo. Se usa de referencia las ecuaciones (2.1) y (2.2).

$$A_{lin} = \left[\frac{\partial f(x, y)}{\partial x} \right]_{\bar{x}, \bar{u}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} \end{bmatrix}_{\bar{x}, \bar{u}} \quad (2.1)$$

$$B_{lin} = \left[\frac{\partial f(x, y)}{\partial u} \right]_{\bar{x}, \bar{u}} = \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \\ \frac{\partial f_3}{\partial u} \\ \frac{\partial f_4}{\partial u} \end{bmatrix}_{\bar{x}, \bar{u}} \quad (2.2)$$

donde:

- A_{lin} es la matriz Jacobiana del sistema evaluada en el punto de equilibrio.
- B_{lin} es la matriz Jacobiana para la entrada del sistema evaluada en el punto de equilibrio.

Asimismo las funciones a derivar f_1 , f_2 , f_3 y f_4 son las siguientes:

$$f_1 = \dot{x}_1 = x_2 \quad (2.3)$$

$$f_2 = \dot{x}_2 = \frac{\frac{\tau}{R} - x_2 \left(c_v + \frac{c_\omega}{R^2} \right) + l m_h \gamma + \frac{c_\omega x_4}{R}}{m_h + m_r + \frac{I_r}{R^2} - \frac{l^2 m_h^2 \cos(x_3)^2}{m_h l^2 + I_p}} \quad (2.4)$$

$$f_3 = \dot{x}_3 = x_4 \quad (2.5)$$

$$f_4 = \dot{x}_4 = \frac{-c_\omega x_4 + l m_h (g \sin(x_3) - \beta) + \frac{c_\omega x_2}{R}}{I_p + l^2 m_h - \frac{l^2 m_h^2 \cos(x_3)^2}{m_h + m_r + \frac{I_r}{R^2}}} \quad (2.6)$$

donde:

$$\gamma = x_4^2 \sin(x_3) - \frac{\cos(x_3) \left(-c_\omega x_4 + \frac{c_\omega x_2}{R} + l g m_h \sin(x_3) \right)}{m_h l^2 + I_p} \quad (2.7)$$

$$\beta = \frac{\cos(x_3) \left(\frac{\tau}{R} - x_2 \left(c_v + \frac{c_\omega}{R^2} \right) \right)}{m_h + m_r + \frac{I_r}{R^2}} + \frac{\cos(x_3) \left(\frac{c_\omega x_4}{R} + l m_h x_4^2 \sin(x_3) \right)}{m_h + m_r + \frac{I_r}{R^2}} \quad (2.8)$$

Al aplicar las ecuaciones anteriores se obtiene la forma general del modelo linealizado mostrado en las ecuaciones (2.9) y (2.10). Es necesario remarcar que muchas derivadas al realizarlas tienen el valor de 0 o 1, las derivadas que no tienen un valor numérico exacto se realizan a continuación.

$$A_{lin} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} \end{bmatrix}_{x=[0,0,0,0], \tau=0} \quad (2.9)$$

$$B_{lin} = \begin{bmatrix} 0 \\ \frac{\partial f_2}{\partial \tau} \\ 0 \\ \frac{\partial f_4}{\partial \tau} \end{bmatrix}_{x=[0,0,0,0], \tau=0} \quad (2.10)$$

2.2.1. Derivadas de la ecuación del estado 2

Las derivadas de la ecuación de estado 2 se muestran en las ecuaciones (2.11), (2.12) y (2.27). Para este procedimiento se utilizan algunas variables auxiliares que se definen durante el procedimiento. Estas variables auxiliares se utilizan en esta subsección, pero en la subsección (2.2.2) se reutilizan con otros valores.

$$\begin{aligned}\frac{\partial f_2}{\partial x_2} &= \frac{-\left(c_v + \frac{c_\omega}{R^2}\right) - \frac{l m_h \cos(\theta) \frac{c_\omega}{R}}{I_p + l^2 m_h}}{m_h + m_r + \frac{I_r}{R^2} - \frac{l^2 m_h^2 \cos(\theta)^2}{m_h l^2 + I_p}} \\ &= -\frac{c_v + \frac{c_\omega}{R^2} + \frac{l c_\omega m_h \cos(\theta)}{R(m_h l^2 + I_p)}}{m_h + m_r + \frac{I_r}{R^2} - \frac{l^2 m_h^2 \cos(\theta)^2}{m_h l^2 + I_p}}\end{aligned}\quad (2.11)$$

$$\frac{\partial f_2}{\partial x_4} = \frac{\frac{c_\omega}{R} + l m_h \left(2 \omega_p \sin(\theta) + \frac{c_\omega \cos(\theta)}{m_h l^2 + I_p}\right)}{m_h + m_r + \frac{I_r}{R^2} - \frac{l^2 m_h^2 \cos(\theta)^2}{m_h l^2 + I_p}}\quad (2.12)$$

Para la derivada respecto al estado x_3 se realiza el siguiente procedimiento. Se establecen una serie de variables auxiliares definidas en (2.13).

$$\begin{aligned}x_2 &= v \\ x_3 &= \theta \\ x_4 &= \omega_p \\ M &= m_h l^2 + I_p \\ \Delta(\theta) &= m_h + m_r + \frac{I_r}{R^2} - \frac{l^2 m_h^2 \cos^2 \theta}{M}\end{aligned}\quad (2.13)$$

Se redefine la operación a realizar como se muestran en (2.14).

$$\begin{aligned}f_2 &= \dot{x}_2 = \frac{N}{\Delta} \\ N &= \frac{\tau}{R} - v \left(c_v + \frac{c_\omega}{R^2}\right) + l m_h \gamma + \frac{c_\omega \omega_p}{R}\end{aligned}\quad (2.14)$$

Se debe tener en cuenta la variable auxiliar γ corresponde a (2.15).

$$\begin{aligned}\gamma &= \omega_p^2 \sin \theta - \frac{\cos \theta E}{M}, \\ E &= -c_\omega \omega_p + \frac{c_\omega v}{R} + l g m_h \sin \theta\end{aligned}\quad (2.15)$$

Se vuelven a definir una serie de variables auxiliares (2.16) que permiten trabajar las derivadas de manera más modular.

$$\begin{aligned}
 A &= \frac{\sin \theta \left(\frac{c_\omega v}{R} - c_\omega \omega_p + l g m_h \sin \theta \right)}{M} \\
 B &= \frac{l g m_h \cos^2 \theta}{M} \\
 C &= \frac{\tau}{R} - v \left(c_v + \frac{c_\omega}{R^2} \right) \\
 D &= \frac{\cos \theta \left(\frac{c_\omega v}{R} - c_\omega \omega_p + l g m_h \sin \theta \right)}{M}
 \end{aligned} \tag{2.16}$$

Se aplica la regla del cociente.

$$\frac{\partial f_2}{\partial x_3} = \frac{\Delta \frac{\partial N}{\partial x_3} - N \frac{\partial \Delta}{\partial x_3}}{\Delta^2} \tag{2.17}$$

$$\frac{\partial N}{\partial x_3} = l m_h \frac{\partial \gamma}{\partial x_3} \tag{2.18}$$

Ahora,

$$\begin{aligned}
 \frac{\partial \gamma}{\partial x_3} &= \frac{d}{d\theta} \left(\omega_p^2 \sin \theta \right) - \frac{d}{d\theta} \left(\frac{\cos \theta E}{M} \right) \\
 &= \omega_p^2 \cos \theta - \frac{1}{M} \left(-\sin \theta E + \cos \theta \frac{dE}{d\theta} \right).
 \end{aligned} \tag{2.19}$$

Como $\frac{dE}{d\theta} = l g m_h \cos \theta$, resulta

$$\frac{\partial \gamma}{\partial x_3} = \omega_p^2 \cos \theta + \frac{\sin \theta E}{M} - \frac{l g m_h \cos^2 \theta}{M}. \tag{2.20}$$

Usando las variables auxiliares A y B (definidas arriba) se tiene la forma compacta

$$\frac{\partial N}{\partial x_3} = l m_h \left(\omega_p^2 \cos \theta + A - B \right) \tag{2.21}$$

$$\begin{aligned}
 \Delta &= -\frac{l^2 m_h^2 \cos^2 \theta}{M}, \\
 \frac{\partial \Delta}{\partial x_3} &= -\frac{l^2 m_h^2}{M} \frac{d}{d\theta} (\cos^2 \theta) = -\frac{l^2 m_h^2}{M} \cdot (-2 \cos \theta \sin \theta) \\
 &= \frac{2 l^2 m_h^2 \cos \theta \sin \theta}{M}
 \end{aligned} \tag{2.22}$$

Ahora,

$$\begin{aligned}\frac{\partial \gamma}{\partial x_3} &= \frac{d}{d\theta} \left(\omega_p^2 \sin \theta \right) - \frac{d}{d\theta} \left(\frac{\cos \theta E}{M} \right) \\ &= \omega_p^2 \cos \theta - \frac{1}{M} \left(-\sin \theta E + \cos \theta \frac{dE}{d\theta} \right).\end{aligned}\tag{2.23}$$

$$\begin{aligned}\frac{\partial f_2}{\partial x_3} &= \frac{\Delta \cdot (l m_h (\omega_p^2 \cos \theta + A - B)) - N \cdot \left(\frac{2l^2 m_h^2 \cos \theta \sin \theta}{M} \right)}{\Delta^2} \\ &= \frac{l m_h (\omega_p^2 \cos \theta + A - B)}{\Delta} - \frac{2l^2 m_h^2 \cos \theta \sin \theta}{M} \cdot \frac{N}{\Delta^2}\end{aligned}\tag{2.24}$$

$$N = C + l m_h \gamma + \frac{c_w \omega_p}{R} = C + l m_h (\omega_p^2 \sin \theta - D) + \frac{c_w \omega_p}{R}\tag{2.25}$$

La forma final se puede expresar de la siguiente forma.

$$\frac{\partial f_2}{\partial x_3} = \frac{l m_h (\omega_p^2 \cos \theta + A - B)}{\Delta} - \frac{2l^2 m_h^2 \cos \theta \sin \theta (C + l m_h (\omega_p^2 \sin \theta - D) + \frac{c_w \omega_p}{R})}{M \Delta^2}\tag{2.26}$$

Finalmente, la expresión final de esta derivada se muestra en (2.27).

$$\begin{aligned}\frac{\partial f_2}{\partial x_3} &= \frac{l m_h (\omega_p^2 \cos(\theta) + A - B)}{m_h + m_r + \frac{I_r}{R^2} - \frac{l^2 m_h^2 \cos(\theta)^2}{m_h l^2 + I_p}} \\ &\quad - \frac{2l^2 m_h^2 \cos(\theta) \sin(\theta) (C + l m_h (\omega_p^2 \sin(\theta) - D) + \frac{c_w \omega_p}{R})}{(m_h l^2 + I_p) \left(m_h + m_r + \frac{I_r}{R^2} - \frac{l^2 m_h^2 \cos(\theta)^2}{m_h l^2 + I_p} \right)^2}\end{aligned}\tag{2.27}$$

donde

$$A = \frac{\sin(\theta) \left(\frac{c_w v}{R} - c_w \omega_p + l g m_h \sin(\theta) \right)}{m_h l^2 + I_p}.\tag{2.28}$$

$$B = \frac{l g m_h \cos(\theta)^2}{m_h l^2 + I_p}\tag{2.29}$$

$$C = \frac{\tau}{R} - v \left(c_v + \frac{c_w}{R^2} \right)\tag{2.30}$$

$$D = \frac{\cos(\theta) \left(\frac{c_w v}{R} - c_w \omega_p + l g m_h \sin(\theta) \right)}{m_h l^2 + I_p}\tag{2.31}$$

2.2.2. Derivadas de la ecuación del estado 4

Las derivadas de la ecuación de estado 4 se muestran en las expresiones (2.32), (2.33) y (2.48).

$$\begin{aligned}
 \frac{\partial f_4}{\partial x_2} &= \frac{\frac{c_\omega}{R} + \frac{l m_h \cos(\theta) \left(c_v + \frac{c_\omega}{R^2} \right)}{m_h + m_r + \frac{I_r}{R^2}}}{I_p + l^2 m_h - \frac{l^2 m_h^2 \cos^2(\theta)}{m_h + m_r + \frac{I_r}{R^2}}} \\
 &= \frac{\frac{c_\omega}{R} + \frac{l m_h \cos(\theta) \left(\frac{c_v R^2 + c_\omega}{R^2} \right)}{\frac{m_h R^2 + m_r R^2 + I_r}{R^2}}}{I_p + l^2 m_h - \frac{l^2 m_h^2 \cos^2(\theta)}{\frac{m_h R^2 + m_r R^2 + I_r}{R^2}}} \\
 &= \frac{\frac{c_\omega}{R} + \frac{l m_h \cos(\theta) (c_v R^2 + c_\omega)}{I_r + R^2 m_h + R^2 m_r}}{I_p + l^2 m_h - \frac{l^2 R^2 m_h^2 \cos^2(\theta)^2}{I_r + R^2 m_h + R^2 m_r}}
 \end{aligned} \tag{2.32}$$

$$\begin{aligned}
 \frac{\partial f_4}{\partial x_4} &= \frac{- \left(c_\omega + l m_h \cos(\theta) \left(\frac{\frac{c_\omega}{R} + 2 l m_h \omega_p \sin(\theta)}{m_h + m_r + \frac{I_r}{R^2}} \right) \right)}{I_p + l^2 m_h - \frac{l^2 m_h^2 \cos^2(\theta)}{m_h + m_r + \frac{I_r}{R^2}}} \\
 &= \frac{- \left(c_\omega + l m_h \cos(\theta) \left(\frac{c_\omega + 2 l m_h R \omega_p \sin(\theta)}{R (m_h + m_r + \frac{I_r}{R^2})} \right) \right)}{I_p + l^2 m_h - \frac{l^2 m_h^2 \cos^2(\theta)}{\left(\frac{m_h R^2 + m_r R^2 + I_r}{R^2} \right)}} \\
 &= \frac{- \left(c_\omega + l m_h R^2 \cos(\theta) \left(\frac{c_\omega + 2 l m_h R \omega_p \sin(\theta)}{R (m_h R^2 + m_r R^2 + I_r)} \right) \right)}{I_p + l^2 m_h - \left(\frac{l^2 m_h^2 R^2 \cos^2(\theta)}{m_h R^2 + m_r R^2 + I_r} \right)} \\
 &= - \frac{c_\omega + \frac{l R m_h \cos(\theta) (c_\omega + 2 l R m_h \omega_p \sin(\theta))}{I_r + R^2 m_h + R^2 m_r}}{I_p + l^2 m_h - \frac{l^2 R^2 m_h^2 \cos^2(\theta)^2}{I_r + R^2 m_h + R^2 m_r}}
 \end{aligned} \tag{2.33}$$

Para obtener la derivada respecto al estado x_3 se realiza el siguiente procedimiento.

Se hace uso de un conjunto de variables auxiliares, definidas en (2.34) a (2.37), para llevar el proceso de la derivada de mejor manera.

$$\begin{aligned}
 x_2 &= v \\
 x_3 &= \theta \\
 x_4 &= \omega_p \\
 M_2 &= m_h + m_r + \frac{I_r}{R^2} \\
 \Delta(\theta) &= I_p + l^2 m_h - \frac{l^2 m_h^2 \cos^2 \theta}{M_2}
 \end{aligned} \tag{2.34}$$

$$\begin{aligned}
 f_4 = \dot{x}_4 &= \frac{N_4}{\Delta_4} \\
 N_4 &= -c_\omega \omega_p + l m_h (g \sin \theta - \beta) + \frac{c_\omega v}{R}
 \end{aligned} \tag{2.35}$$

$$\begin{aligned}
 \beta &= \frac{\cos \theta}{M_2} \left(C + \frac{c_\omega \omega_p}{R} + l m_h \omega_p^2 \sin \theta \right) \\
 C &= \frac{\tau}{R} - v \left(c_v + \frac{c_\omega}{R^2} \right)
 \end{aligned} \tag{2.36}$$

$$F = C + \frac{c_\omega \omega_p}{R} + l m_h \omega_p^2 \sin \theta \tag{2.37}$$

de modo que

$$\beta = \frac{\cos \theta F}{M_2}. \tag{2.38}$$

Se procede a aplicar la regla del cociente con las variables.

$$\frac{\partial f_4}{\partial x_3} = \frac{\Delta_4 \frac{\partial N_4}{\partial x_3} - N_4 \frac{\partial \Delta_4}{\partial x_3}}{\Delta_4^2}. \tag{2.39}$$

Se calculan las derivadas parciales donde solo dependen de θ en N_4 los términos $l m_h (g \sin \theta - \beta)$. Por tanto

$$\frac{\partial N_4}{\partial x_3} = l m_h \left(g \cos \theta - \frac{\partial \beta}{\partial \theta} \right). \tag{2.40}$$

Se calcula $\frac{\partial \beta}{\partial \theta}$. Como $\beta = \frac{\cos \theta F}{M_2}$ y F depende de θ solo por $l m_h \omega_p^2 \sin \theta$, se tiene.

$$\begin{aligned}
 \frac{\partial \beta}{\partial \theta} &= \frac{1}{M_2} \left(-\sin \theta F + \cos \theta \frac{dF}{d\theta} \right) \\
 &= -\frac{\sin \theta F}{M_2} + \frac{\cos \theta l m_h \omega_p^2 \cos \theta}{M_2} \\
 &= -\frac{\sin \theta F}{M_2} + \frac{l m_h \omega_p^2 \cos^2 \theta}{M_2}
 \end{aligned} \tag{2.41}$$

Lo anterior se sustituye en $\partial N_4 / \partial x_3$.

$$\frac{\partial N_4}{\partial x_3} = l m_h \left(g \cos \theta + \frac{\sin \theta F}{M_2} - \frac{l m_h \omega_p^2 \cos^2 \theta}{M_2} \right) \tag{2.42}$$

Se definen un par de variables auxiliares

$$\begin{aligned} A &= \frac{\sin \theta F}{M_2} \\ B &= \frac{lm_h \omega_p^2 \cos^2 \theta}{M_2} \end{aligned} \quad (2.43)$$

de modo que la expresión compacta es:

$$\frac{\partial N_4}{\partial x_3} = lm_h (g \cos \theta + A - B). \quad (2.44)$$

Se calculan el resto de las derivadas parciales.

$$\begin{aligned} \Delta_4 &= -\frac{l^2 m_h^2 \cos^2 \theta}{M_2}, \\ \frac{\partial \Delta_4}{\partial x_3} &= -\frac{l^2 m_h^2}{M_2} \cdot \frac{d}{d\theta}(\cos^2 \theta) = -\frac{l^2 m_h^2}{M_2} \cdot (-2 \cos \theta \sin \theta) \\ &= \frac{2l^2 m_h^2 \cos \theta \sin \theta}{M_2} \end{aligned} \quad (2.45)$$

Sustituyendo $\partial N_4 / \partial x_3$ y $\partial \Delta_4 / \partial x_3$ en (2.39) se obtiene

$$\begin{aligned} \frac{\partial f_4}{\partial x_3} &= \frac{\Delta_4 \cdot (lm_h (g \cos \theta + A - B)) - N_4 \cdot \left(\frac{2l^2 m_h^2 \cos \theta \sin \theta}{M_2} \right)}{\Delta_4^2} \\ &= \frac{lm_h (g \cos \theta + A - B)}{\Delta_4} - \frac{2l^2 m_h^2 \cos \theta \sin \theta}{M_2} \cdot \frac{N_4}{\Delta_4^2}. \end{aligned} \quad (2.46)$$

Recordando

$$N_4 = -c_\omega \omega_p + lm_h (g \sin \theta - \beta) + \frac{c_\omega v}{R},$$

y usando $\beta = \frac{\cos \theta F}{M_2}$, también puede expresarse como

$$N_4 = lm_h \left(g \sin \theta - \frac{\cos \theta F}{M_2} \right) - c_\omega \omega_p + \frac{c_\omega v}{R}.$$

Se sustituye lo anterior en el segundo término para obtener la forma general de la derivada. Obteniendo la ecuación (2.47).

$$\frac{\partial f_4}{\partial x_3} = \frac{lm_h (g \cos \theta + A - B)}{\Delta_4} - \frac{2l^2 m_h^2 \cos \theta \sin \theta}{M_2} \frac{N_4}{\Delta_4^2} \quad (2.47)$$

$$\begin{aligned} \frac{\partial f_4}{\partial x_3} = & \frac{l m_h (g \cos(\theta) + A - B)}{I_p + l^2 m_h - \frac{l^2 R^2 m_h^2 \cos(\theta)^2}{I_r + R^2 m_h + R^2 m_r}} \\ & - \frac{2l^2 R^2 m_h^2 \cos(\theta) \sin(\theta) (l m_h (g \sin(\theta) - C) - c_\omega \omega_p + \frac{c_\omega v}{R})}{(I_r + R^2 m_h + R^2 m_r) \left(I_p + l^2 m_h - \frac{l^2 R^2 m_h^2 \cos(\theta)^2}{I_r + R^2 m_h + R^2 m_r} \right)^2} \end{aligned} \quad (2.48)$$

donde

$$A = \frac{\sin(\theta) (l m_h \sin(\theta) R^2 \omega_p^2 - c_v v R^2 + c_\omega R \omega_p + \tau R - c_\omega v)}{I_r + R^2 m_h + R^2 m_r} \quad (2.49)$$

$$B = \frac{l R^2 m_h \omega_p^2 \cos(\theta)^2}{I_r + R^2 m_h + R^2 m_r} \quad (2.50)$$

$$C = \frac{\cos(\theta) (l m_h \sin(\theta) R^2 \omega_p^2 - c_v v R^2 + c_\omega R \omega_p + \tau R - c_\omega v)}{I_r + R^2 m_h + R^2 m_r} \quad (2.51)$$

2.2.3. Derivadas de las ecuaciones de estado respecto a la entrada

Las derivadas de las ecuaciones de estado respecto a la entrada del sistema τ se muestran en las ecuaciones (2.52) y (2.53).

$$\frac{\partial f_2}{\partial \tau} = \frac{1}{R \left(m_h + m_r + \frac{I_r}{R^2} - \frac{l^2 m_h^2 \cos(\theta)^2}{m_h l^2 + I_p} \right)} \quad (2.52)$$

$$\frac{\partial f_4}{\partial \tau} = - \frac{l R m_h \cos(\theta)}{(I_r + R^2 m_h + R^2 m_r) \left(I_p + l^2 m_h - \frac{L^2 R^2 m_h^2 \cos(\theta)^2}{I_r + R^2 m_h + R^2 m_r} \right)} \quad (2.53)$$

2.3. Diseño del controlador por retroalimentación de estados

2.3.1. Diseño del controlador en tiempo continuo

Se procede a diseñar el controlador por retroalimentación de estados. Se obtienen las matrices $\mathbf{A}_{\text{cont}}$ y $\mathbf{B}_{\text{cont}}$ al evaluarlas en el punto de equilibrio $\mathbf{x}_{\text{pe}} = [0, 0, 0, 0]$, esto se muestra en las ecuaciones (2.54) y (2.55). Este proceso se basa en lo descrito en [30]. Asimismo, es necesario describir la nomenclatura utilizada.

- $\mathbf{A}_{\text{cont}}$ es la matriz jacobiana evaluada en el punto de equilibrio y con los valores obtenidos en 1.92.

- $\mathbf{B}_{\text{cont}}$ es la matriz jacobiana correspondiente a la entrada evaluada en el punto de equilibrio y con los valores obtenidos en 1.92.
- $\mathbf{x}_{\text{pe}}$ es el vector del punto de equilibrio utilizado en el proceso de linealización.
- $\mathbf{MC}_{\text{cont}}$ es la matriz de controlabilidad para el sistema en tiempo continuo.
- $\bar{\mathbf{C}}_{\text{cont}}$, $\bar{\mathbf{C}}_{\text{cont}}^{-1}$, $\mathbf{P}_{\text{cont}}^{-1}$, $\mathbf{P}_{\text{cont}}$ y $\bar{\mathbf{k}}_{\text{cont}}$ son matrices o vectores auxiliares utilizados durante el procedimiento de Ackermann.
- $\mathbf{k}_{\text{cont}}$ es el vector de retroalimentación en tiempo continuo.
- τ_{cont} es la entrada de control diseñada para el tiempo continuo, siendo el par generado por los motores del robot.
- $\mathbf{f}_{\text{cont}}$ es la fuerza que permite al robot mantenerse en pie.

$$\mathbf{A}_{\text{cont}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -9.1180 & -3.8980 & 0.1807 \\ 0 & 0 & 0 & 1 \\ 0 & 93.6180 & 107.4047 & -2.1473 \end{bmatrix} \quad (2.54)$$

$$\mathbf{B}_{\text{cont}} = \begin{bmatrix} 0 \\ 55.2071 \\ 0 \\ -432.5587 \end{bmatrix} \quad (2.55)$$

Los valores propios de la matriz $\mathbf{A}_{\text{cont}}$ se muestran en la ecuación (2.56).

$$\begin{aligned} s_1 &= 0 \\ s_2 &= -15.4686 \\ s_3 &= -4.5418 \\ s_4 &= 8.7451 \end{aligned} \quad (2.56)$$

Y la matriz de controlabilidad del sistema se muestra en la ecuación (2.57).

$$\begin{aligned} \mathbf{MC}_{\text{cont}} &= [\mathbf{B}_{\text{cont}}, \mathbf{A}_{\text{cont}}\mathbf{B}_{\text{cont}}, \mathbf{A}_{\text{cont}}^2\mathbf{B}_{\text{cont}}, \mathbf{A}_{\text{cont}}^3\mathbf{B}_{\text{cont}}] \\ &= (1 \times 10^5) \begin{bmatrix} 0 & 0.001 & -0.006 & 0.081 \\ 0.001 & -0.006 & 0.081 & -1.181 \\ 0 & -0.004 & 0.061 & -1.140 \\ -0.004 & 0.061 & -1.140 & 16.570 \end{bmatrix} \end{aligned} \quad (2.57)$$

Se evalúa el rango de la matriz de controlabilidad, siendo de valor pleno. Y se obtiene el polinomio característico de la matriz $\mathbf{A}_{\text{cont}}$, esto se muestra en la ecuación (2.58).

$$s^4 + 11.2652s^3 - 104.7382s^2 - 614.3931s = s^4 + \alpha_1s^3 + \alpha_2s^2 + \alpha_3s + \alpha_4 = 0 \quad (2.58)$$

Para realizar el procedimiento de Ackerman es necesario calcular algunas matrices, como lo es la matriz $\bar{\mathbf{C}}_{\text{cont}}^{-1}$.

$$\begin{aligned} \bar{\mathbf{C}}_{\text{cont}}^{-1} &= \begin{bmatrix} 1 & \alpha_1 & \alpha_2 & \alpha_3 \\ 0 & 1 & \alpha_1 & \alpha_2 \\ 0 & 0 & 1 & \alpha_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 11.2652 & -104.7382 & -614.3931 \\ 0 & 1 & 11.2652 & -104.7382 \\ 0 & 0 & 1 & 11.2652 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (2.59)$$

Se proceden a obtener las matrices $\mathbf{P}_{\text{cont}}^{-1}$ y $\mathbf{P}_{\text{cont}}$ que se muestran en las ecuaciones (2.60) y (2.61) respectivamente.

$$\mathbf{P}_{\text{cont}}^{-1} = (\mathbf{M}\mathbf{C}_{\text{cont}})(\bar{\mathbf{C}}_{\text{cont}}^{-1}) = (1 \times 10^3) \begin{bmatrix} 0 & 0.055 & 0.040 & -4.243 \\ 0.055 & 0.040 & -4.243 & 0 \\ 0 & -0.432 & 1.224 & 0 \\ -0.432 & 1.224 & 0 & 0 \end{bmatrix} \quad (2.60)$$

$$\mathbf{P}_{\text{cont}} = (\mathbf{P}_{\text{cont}}^{-1})^{-1} = \begin{bmatrix} 0 & -0.0022 & -0.0075 & -0.0026 \\ 0 & -0.0008 & -0.0027 & -0.0001 \\ 0 & -0.0003 & -0.0001 & 0 \\ -0.0002 & 0 & 0 & 0 \end{bmatrix} \quad (2.61)$$

Para el diseño del controlador por retroalimentación de estados se seleccionan cuatro valores propios deseados (2.62), dos complejos conjugados y dos reales. La elección de los valores complejos conjugados obedece a la naturaleza oscilatoria de la dinámica del robot de autobalance, equivalente a un péndulo invertido, permitiendo imponer una respuesta de segundo orden subamortiguada en lazo cerrado. Dichos valores propios se ubican cercanos al eje imaginario para que tengan una mayor influencia en la respuesta transitoria. Los

polos reales restantes se sitúan más alejados del eje imaginario para garantizar que su dinámica sea más rápida y no influya de manera significativa en la respuesta transitoria del sistema. Esta separación de escalas temporales permite obtener un comportamiento estable y suave, validado mediante las simulaciones y pruebas experimentales, mostradas en los siguientes capítulos.

$$\begin{aligned} s_1 &= -5 + j \\ s_2 &= -5 - j \\ s_3 &= -15 \\ s_4 &= -25 \end{aligned} \quad (2.62)$$

Con los valores propios deseados se establece el polinomio característico deseado, este se muestra en la ecuación (2.63).

$$s^4 + 50s^3 + 801s^2 + 4790s + 9750 = s^4 + \bar{\alpha}_1 s^3 + \bar{\alpha}_2 s^2 + \bar{\alpha}_3 s + \bar{\alpha}_4 = 0 \quad (2.63)$$

Las constantes de $\bar{\mathbf{k}}_{\text{cont}}$ se obtienen mediante la ecuación (2.64).

$$\bar{\mathbf{k}}_{\text{cont}} = \begin{bmatrix} \bar{\alpha}_1 - \alpha_1 \\ \bar{\alpha}_2 - \alpha_2 \\ \bar{\alpha}_3 - \alpha_3 \\ \bar{\alpha}_4 - \alpha_4 \end{bmatrix}^T = (1 \times 10^3) \begin{bmatrix} 0.0387 \\ 0.9057 \\ 5.4044 \\ 9.7500 \end{bmatrix}^T \quad (2.64)$$

Y se procede a obtener las constantes finales de retroalimentación mostrada en la ecuación (2.65).

$$\mathbf{k}_{\text{cont}} = (\bar{\mathbf{k}}_{\text{cont}})(\mathbf{P}_{\text{cont}}) = \begin{bmatrix} -2.2977 & -2.3680 & -3.7172 & -0.3918 \end{bmatrix} \quad (2.65)$$

Con las constantes de retroalimentación se establece la ley de control de entrada al sistema, esto se muestra en la ecuación (2.66).

$$\mathbf{f}_{\text{cont}} = 0 - \frac{\tau_{\text{cont}}}{R} = -\frac{1}{R} [-2.2977 x_1 + -2.3680 x_2 + -3.7172 x_3 + -0.3918 x_4]$$

(2.66)

El diseño en tiempo continuo se utiliza como etapa de análisis y validación teórica. Los resultados de la simulación en tiempo continuo se muestran en el capítulo (3). Posteriormente el diseño en tiempo discreto se emplea para garantizar una implementación correcta en el microcontrolador del robot.

2.3.2. Diseño del controlador en tiempo discreto

Se procede a discretizar el sistema linealizado usando un retenedor de orden cero con un tiempo de muestreo de 15 ms. El sistema linealizado y discretizado se muestra en (2.67) y (2.68).

- $\mathbf{A}_{\text{disc}}$ es la matriz jacobiana evaluada en el punto de equilibrio y discretizada mediante un retenedor de orden cero.
- $\mathbf{B}_{\text{disc}}$ es la matriz jacobiana correspondiente a la entrada evaluada en el punto de equilibrio y discretizada mediante un retenedor de orden cero.
- $\mathbf{MC}_{\text{disc}}$ es la matriz de controlabilidad para el sistema en tiempo discreto.
- $\bar{\mathbf{C}}_{\text{disc}}$, $\bar{\mathbf{C}}_{\text{disc}}^{-1}$, $\mathbf{P}_{\text{disc}}^{-1}$, $\mathbf{P}_{\text{disc}}$ y $\bar{\mathbf{k}}_{\text{disc}}$ son matrices o vectores auxiliares utilizados durante el procedimiento de Ackermann para el sistema en tiempo discreto.
- $\mathbf{k}_{\text{disc}}$ es el vector de retroalimentación en tiempo discreto.
- τ_{disc} es la entrada de control diseñada para el tiempo discreto, siendo el par generado por los motores del robot.
- $\mathbf{f}_{\text{disc}}$ es la fuerza que permite al robot mantenerse en pie y es la ley de control implementada en el microcontrolador del robot.

$$\mathbf{A}_{\text{disc}} = \begin{bmatrix} 1 & 0.01403 & -0.0004097 & 1.716 \times 10^{-05} \\ 0 & 0.8737 & -0.05284 & 0.002088 \\ 0 & 0.009983 & 1.012 & 0.01483 \\ 0 & 1.297 & 1.554 & 0.9817 \end{bmatrix} \quad (2.67)$$

$$\mathbf{B}_{\text{disc}} = \begin{bmatrix} 0.0059 \\ 0.767 \\ -0.04546 \\ -5.863 \end{bmatrix} \quad (2.68)$$

Los valores propios del sistema discreto mapeados al círculo unitario se muestran en la ecuación (2.69). Se observa que un polo al estar fuera del círculo unitario hace al sistema

en lazo abierto inestable.

$$\begin{aligned}
 z_1 &= e^{(0.015)(0)} = 1 \\
 z_2 &= e^{(0.015)(-15.4686)} = 0.7929 \\
 z_3 &= e^{(0.015)(8.7451)} = 1.1402 \\
 z_4 &= e^{(0.015)(-4.5418)} = 0.9341
 \end{aligned} \tag{2.69}$$

La matriz de controlabilidad se muestra en la ecuación (2.70). Al obtener el rango de esta matriz se obtiene que es de rango pleno y por ende el sistema discretizado también es controlable.

$$\mathbf{MC}_{\text{disc}} = \begin{bmatrix} 0.0059 & 0.0166 & 0.0258 & 0.0339 \\ 0.7670 & 0.6603 & 0.5734 & 0.5026 \\ -0.0455 & -0.1253 & -0.1918 & -0.2489 \\ -5.8632 & -4.8319 & -4.0818 & -3.5615 \end{bmatrix} \tag{2.70}$$

El polinomio característico de la matriz $\mathbf{A}_{\text{disc}}$ esta dado en la ecuación (2.71).

$$z^4 - 3.8672z^3 + 5.5771z^2 - 3.5544z + 0.8445 = z^4 + \alpha_1z^3 + \alpha_2z^2 + \alpha_3z + \alpha_4 = 0 \tag{2.71}$$

Se obtiene la matriz $\bar{\mathbf{C}}_{\text{disc}}^{-1}$ como se muestra en la ecuación (2.72).

$$\begin{aligned}
 \bar{\mathbf{C}}_{\text{disc}}^{-1} &= \begin{bmatrix} 1 & \alpha_1 & \alpha_2 & \alpha_3 \\ 0 & 1 & \alpha_1 & \alpha_2 \\ 0 & 0 & 1 & \alpha_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & -3.8672 & 5.5771 & -3.5544 \\ 0 & 1 & -3.8672 & 5.5771 \\ 0 & 0 & 1 & -3.8672 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{2.72}$$

Las matrices $\mathbf{P}_{\text{disc}}^{-1}$ y $\mathbf{P}_{\text{disc}}$ se muestran en las ecuaciones (2.73) y (2.74).

$$\mathbf{P}_{\text{disc}}^{-1} = (\mathbf{MC}_{\text{disc}})(\bar{\mathbf{C}}^{-1}) = \begin{bmatrix} 0.0059 & -0.0062 & -0.0054 & 0.0055 \\ 0.7670 & -2.3059 & 2.2975 & -0.7587 \\ -0.0455 & 0.0505 & 0.0391 & -0.0442 \\ -5.8632 & 17.8426 & -18.0955 & 6.1162 \end{bmatrix} \tag{2.73}$$

$$\mathbf{P}_{\text{disc}} = (\mathbf{P}_{\text{disc}}^{-1})^{-1} = (1.0 \times 10^3) \begin{bmatrix} -5.0495 & -0.4022 & -0.8374 & -0.0514 \\ -5.0495 & -0.3114 & -0.7850 & -0.0397 \\ -5.0495 & -0.2243 & -0.7456 & -0.0286 \\ -5.0495 & -0.1408 & -0.7188 & -0.0179 \end{bmatrix} \quad (2.74)$$

Los valores deseados para el sistema en lazo cerrado se muestran en la ecuación (2.75). Dos de esos valores propios son complejos conjugados cercanos al eje imaginario para asegurar un comportamiento similar a un sistema de segundo orden sub amortiguado.

$$\begin{aligned} z_1 &= e^{(0.015)(-5+j)} = 0.9276 + 0.0139j \\ z_2 &= e^{(0.015)(-5-j)} = 0.9276 - 0.0139j \\ z_3 &= e^{(0.015)(-15)} = 0.7985 \\ z_4 &= e^{(0.015)(-25)} = 0.6873 \end{aligned} \quad (2.75)$$

El polinomio característico para los valores propios deseados se muestra en la ecuación (2.76).

$$z^4 - 3.3411z^3 + 4.1661z^2 - 2.2970z + 0.4724 = z^4 + \bar{\alpha}_1 z^3 + \bar{\alpha}_2 z^2 + \bar{\alpha}_3 z + \bar{\alpha}_4 = 0 \quad (2.76)$$

Los cálculos para el vector de las ganancias de retroalimentación del controlador se muestran en las ecuaciones (2.77) y (2.78).

$$\bar{\mathbf{k}}_{\text{disc}} = \begin{bmatrix} \bar{\alpha}_1 - \alpha_1 \\ \bar{\alpha}_2 - \alpha_2 \\ \bar{\alpha}_3 - \alpha_3 \\ \bar{\alpha}_4 - \alpha_4 \end{bmatrix}^T = \begin{bmatrix} 0.5262 \\ -1.4110 \\ 1.2573 \\ -0.3722 \end{bmatrix}^T \quad (2.77)$$

$$\mathbf{k}_{\text{disc}} = (\bar{\mathbf{k}}_{\text{disc}})(\mathbf{P}_{\text{disc}}) = \begin{bmatrix} -1.7275 & -1.8906 & -3.0031 & -0.3155 \end{bmatrix} \quad (2.78)$$

Con las constantes de retroalimentación se establece la ley de control de entrada al sistema; esto se muestra en la ecuación (2.79). Esta es la ley de control que se implementa en el robot, teniendo una ley de control en tiempo discreto para un dispositivo digital. Asimismo, la simulación de esta ley de control se muestra en la subsección 3.2.2.

$$\mathbf{f}_{\text{disc}} = 0 - \frac{\tau_{\text{disc}}}{R} = -\frac{1}{R} [-1.7275 x_1 - 1.8906 x_2 - 3.0031 x_3 - 0.3155 x_4] \quad (2.79)$$

Se puede observar que (2.79) tiene la misma estructura de la ley de control en tiempo continuo, pero con las constantes de sintonización obtenidas para el tiempo discreto.

2.4. Implementación del controlador por retroalimentación de estados

Una vez que se tiene el controlador digital se procede a implementarlo en el microcontrolador del robot. Debido a que el robot previamente tenía un conjunto de controladores PI-PD, las variables utilizadas por dichos controladores tienen unidades distintas a las que utiliza el controlador por retroalimentación de estados. Para solucionar esto se establecen algunas constantes de conversión. En (2.80) se muestra cómo deben realizarse estas conversiones.

$$\begin{aligned}x_1(\text{pulsos}) &\Rightarrow x_1(\text{metros}) \\x_2(\text{pulsos}) &\Rightarrow x_2(\text{m/s}) \\x_3(\text{grados}) &\Rightarrow x_3(\text{radianes}) \\x_4(\text{grados/seg}) &\Rightarrow x_4(\text{rad/seg})\end{aligned}\tag{2.80}$$

Las constantes de conversión se muestran en la ecuación (2.81).

$$\begin{aligned}x'_1 &= \frac{\pi*2*R}{(\text{pulsos})*(\text{relaciondetransformacion})}x_1 \\x'_2 &= \frac{\pi*2*R}{(\text{pulsos})*(\text{relaciondetransformacion})}\frac{x_2}{dt} \\x'_3 &= \frac{\pi}{180}x_3 \\x'_4 &= \frac{\pi}{180}x_4\end{aligned}\tag{2.81}$$

El robot no cuenta con sensores de corriente ni con controladores internos de par, por lo que no es posible implementar un control directo de par. En la práctica, los motores son accionados mediante señales PWM cuyo ciclo de trabajo se encuentra en el rango de 0 a 255, lo que equivale a un control de voltaje promedio aplicado a los motores. En consecuencia, todos los controladores desarrollados en este trabajo generan como señal de control el ciclo de trabajo del PWM. Debido a que el controlador desarrollado toma un par como la señal de control, se establece una relación lineal entre el par y el ciclo de trabajo de la señal PWM. Para esta relación se usa un factor de escalamiento de 540.

Por pruebas experimentales se realiza un incremento en la constante del estado x_2 . El cambio se muestra en la ecuación (2.82) y la ley de control implementada se muestra en la ecuación (2.83). Es necesario recalcar que el factor de 0.5 en la ley de control se debe a que son dos motores los que aportan el par que necesita el robot para generar la fuerza necesaria para moverse y mantener el equilibrio.

$$\begin{aligned}
 k_1 &= -1.7275 \\
 k_2 &= -1.8906 \Rightarrow -2.8906 \Rightarrow -3.2 \\
 k_3 &= -3.0031 \\
 k_4 &= -0.3155
 \end{aligned} \tag{2.82}$$

$$\tau_{motor} = (0.5) (540) (-1.7275x'_1 - 1.8906x'_2 - 3.0031x'_3 - 0.3155x'_4) \tag{2.83}$$

2.5. Diseño del controlador no lineal para el seguimiento de trayectoria

2.5.1. Controlador basado en técnicas de Lyapunov

Se plantea un control no lineal para el seguimiento de trayectoria. Se hace uso del modelo de la figura (2.1). El modelo y control utilizado se basa en [29], [28], [31] y [32].

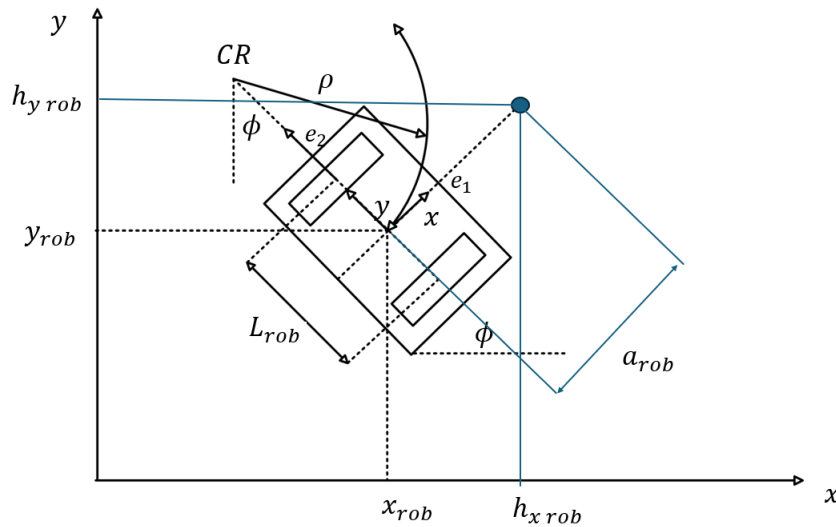


Figura 2.1. Modelo de un robot móvil tipo diferencial.

Para comprender mejor el procedimiento es necesario definir la nomenclatura usada.

- x_{rob} es la coordenada en el eje x del centro de masa del robot.
- y_{rob} es la coordenada en el eje y del centro de masa del robot.
- $\dot{x}_{rob}$ es la velocidad en el eje x del centro de masa del robot.
- $\dot{y}_{rob}$ es la velocidad en el eje y del centro de masa del robot.

- ϕ es el ángulo de rotación del robot sobre el eje perpendicular al plano $x - y$.
- v_{rob} es la velocidad lineal del robot. Es una de las señales del controlador junto a ω_{rob} .
- ω_{rob} es la velocidad angular del robot sobre el eje perpendicular al plano $x - y$. Es una de las señales del controlador junto a v_{rob} .
- a_{rob} es un punto que indica una distancia en adelante respecto al centro de masa del robot. Es usado para el diseño del controlador de seguimiento de trayectoria.
- $\mathbf{h}_{rob}$ es el vector de las coordenadas del punto a_{rob} , medidas mediante la odometría.
- h_{xrob} es la coordenada en el eje x del punto a_{rob} .
- h_{yrob} es la coordenada en el eje y del punto a_{rob} .
- $\dot{\mathbf{h}}_{rob}$ es el vector de las velocidades medidas del punto a_{rob} .
- $\dot{h}_{xrob}$ es la velocidad en el eje x del punto a_{rob} .
- $\dot{h}_{yrob}$ es la velocidad en el eje y del punto a_{rob} .
- $\mathbf{J}_{rob}$ es la matriz jacobiana del robot de arquitectura diferencial en función del ángulo de rotación ϕ del robot y de la distancia a_{rob} .
- $\mathbf{h}_{drob}$ es el vector de las coordenadas deseadas del punto a_{rob} , son las coordenadas de referencia de la trayectoria a seguir.
- h_{dxrob} es la coordenada en x del vector $\mathbf{h}_{drob}$.
- h_{dyrob} es la coordenada en y del vector $\mathbf{h}_{drob}$.
- t_{circ} es la variable de tiempo utilizado para generar los valores de la trayectoria de referencia $\mathbf{h}_{drob}$.
- $\dot{\mathbf{h}}_{drob}$ es el vector de las velocidades de referencia del punto a_{rob} , son las velocidades de la trayectoria que se debe seguir.
- $\dot{h}_{dxrob}$ es la velocidad en el eje x del vector $\dot{\mathbf{h}}_{drob}$.
- $\dot{h}_{dyrob}$ es la velocidad en el eje y del vector $\dot{\mathbf{h}}_{drob}$.
- $\mathbf{e}_h$ es el error entre las coordenadas seguidas por el robot $\mathbf{h}_{rob}$, respecto a las coordenadas de referencia $\mathbf{h}_{drob}$.

- V_{rob} es la función candidata de Lyapunov propuesta para desarrollar el controlador para el seguimiento de trayectoria.
- $\dot{V}_{rob}$ es la derivada de la función candidata de Lyapunov.
- $\mathbf{K}_{rob}$ es la matriz de las constantes del controlador de seguimiento de trayectoria. Se construye a partir del vector $\mathbf{k}_{rob}$ y la matriz $\mathbf{I}_{rob}$.
- $\mathbf{k}_{rob}$ es el vector de las constantes del controlador de seguimiento de trayectoria.
- $\mathbf{I}_{rob}$ indica la matriz identidad.

El modelo de un robot móvil tipo diferencial se muestra en la ecuación (2.84).

$$\begin{bmatrix} \dot{x}_{rob} \\ \dot{y}_{rob} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} v_{rob} \cos(\phi) \\ v_{rob} \sin(\phi) \\ \omega_{rob} \end{bmatrix} = \begin{bmatrix} \cos(\phi) & 0 \\ \sin(\phi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{rob} \\ \omega_{rob} \end{bmatrix} \quad (2.84)$$

Las coordenadas del punto de control en adelante se muestran a continuación.

$$\begin{bmatrix} h_{xrob} \\ h_{yrob} \end{bmatrix} = \begin{bmatrix} x_{rob} + a_{rob} \cos(\phi) \\ y_{rob} + a_{rob} \sin(\phi) \end{bmatrix} \quad (2.85)$$

Utilizando las definiciones mostradas en (2.84), se procede a obtener la primera derivada del punto en adelante mostrada en 2.86.

$$\begin{aligned} \begin{bmatrix} \dot{h}_{xrob} \\ \dot{h}_{yrob} \end{bmatrix} &= \begin{bmatrix} \dot{x}_{rob} - a_{rob} \sin(\phi) \dot{\phi} \\ \dot{y}_{rob} + a_{rob} \cos(\phi) \dot{\phi} \end{bmatrix} = \begin{bmatrix} v_{rob} \cos(\phi) - a_{rob} \sin(\phi) \omega_{rob} \\ v_{rob} \sin(\phi) + a_{rob} \cos(\phi) \omega_{rob} \end{bmatrix} \\ &= \begin{bmatrix} \cos(\phi) & -a_{rob} \sin(\phi) \\ \sin(\phi) & a_{rob} \cos(\phi) \end{bmatrix} \begin{bmatrix} v_{rob} \\ \omega_{rob} \end{bmatrix} \end{aligned} \quad (2.86)$$

La ecuación anterior tiene la forma mostrada en (2.87).

$$\begin{bmatrix} \dot{h}_{xrob} \\ \dot{h}_{yrob} \end{bmatrix} = \mathbf{J}_{rob} \begin{bmatrix} v_{rob} \\ \omega_{rob} \end{bmatrix} \quad (2.87)$$

En esta parte se diseña un controlador para seguimiento de trayectoria basado en técnicas de Lyapunov utilizando el modelo cinemático del robot de arquitectura diferencial, para tal fin se utilizará el error de seguimiento definido en la ecuación (2.88).

$$\mathbf{e}_h = \mathbf{h}_{drob} - \mathbf{h}_{rob} = \begin{bmatrix} h_{dxrob} \\ h_{dyrob} \end{bmatrix} - \begin{bmatrix} h_{xrob} \\ h_{yrob} \end{bmatrix} \quad (2.88)$$

Se propone una función candidata de Lyapunov mostrada en la ecuación (2.89).

$$V_{rob} = \frac{1}{2} \mathbf{e}_h^T \mathbf{e}_h \quad (2.89)$$

La derivada de la función candidata se muestra en la ecuación (2.90).

$$\dot{V}_{rob} = \mathbf{e}_h^T \dot{\mathbf{e}}_h = \mathbf{e}_h^T (\dot{\mathbf{h}}_{d\text{rob}} - \dot{\mathbf{h}}_{\text{rob}}) \quad (2.90)$$

Se busca que $\dot{V}_{rob} < 0$, para esto el error debe disminuir. Para esta tarea se propone la ecuación (2.91).

$$\dot{\mathbf{h}}_{\text{rob}} = \dot{\mathbf{h}}_{d\text{rob}} + \mathbf{K}_{\text{rob}} \mathbf{e}_h \quad (2.91)$$

donde $\mathbf{K}_{\text{rob}}$ es una matriz definida positiva $\mathbf{K}_{\text{rob}} = \mathbf{k}_{\text{rob}} \mathbf{I}_{\text{rob}}$. Se sustituye la ecuación (2.91) en (2.90).

$$\dot{V}_{rob} = \mathbf{e}_h^T (\dot{\mathbf{h}}_{d\text{rob}} - (\dot{\mathbf{h}}_{d\text{rob}} + \mathbf{K}_{\text{rob}} \mathbf{e}_h)) = -\mathbf{e}_h^T \mathbf{K}_{\text{rob}} \mathbf{e}_h < 0 \quad (2.92)$$

Del modelo anterior se obtiene las ecuaciones (2.93) y (2.94).

$$\dot{\mathbf{h}}_{\text{rob}} = \mathbf{J}_{\text{rob}} \begin{bmatrix} v_{rob} \\ \omega_{rob} \end{bmatrix} = \dot{\mathbf{h}}_{d\text{rob}} + \mathbf{K}_{\text{rob}} \mathbf{e}_h \quad (2.93)$$

$$\begin{bmatrix} v_{rob} \\ \omega_{rob} \end{bmatrix} = \mathbf{J}_{\text{rob}}^{-1} [\dot{\mathbf{h}}_{d\text{rob}} + \mathbf{K}_{\text{rob}} \mathbf{e}_h]$$

(2.94)

2.5.2. Odometría para el robot tipo diferencial

Para poder medir el ángulo ϕ y las distancias x e y , que se necesitan para el controlador no lineal, se usa la técnica de odometría para un robot diferencial. Para realizar el proceso es necesario definir la nomenclatura utilizada.

- D_L es la distancia recorrida por la rueda izquierda.
- D_R es la distancia recorrida por la rueda derecha.
- N_L y N_R son el número de las vueltas de las ruedas izquierda y derecha respectivamente.
- $\Delta pulsos$ es el incremento de los pulsos del encoder de cada motor.
- PPR es el número de pulsos que hay en una revolución de cada motor.
- R es el radio de cada rueda.

- L_{rob} es la distancia entre las ruedas del robot.
- $\Delta\phi$ es el incremento del ángulo de rotación del robot.
- ΔS_{rob} es el incremento de la distancia recorrida por el robot.
- ϕ_k es el valor actual del ángulo de rotación ϕ .
- ϕ_{k-1} es el valor anterior del ángulo de rotación ϕ .
- x_{rob} es la coordenada actual del centro de masa del robot en el eje x .
- $x_{rob\,k-1}$ es la coordenada anterior del centro de masa del robot en el eje x .
- y_{rob} es la coordenada actual del centro de masa del robot en el eje y .
- $y_{rob\,k-1}$ es la coordenada anterior del centro de masa del robot en el eje y .
- a_{rob} es un punto que indica una distancia en adelante respecto al centro de masa del robot. Es usado para el diseño del controlador de seguimiento de trayectoria.
- $h_{x\,rob}$ es la coordenada en el eje x del punto a_{rob} .
- $h_{y\,rob}$ es la coordenada en el eje y del punto a_{rob} .
- v_{rob} es la velocidad lineal del robot. Es una de las señales del controlador junto a ω_{rob} .
- ω_{rob} es la velocidad angular del robot sobre el eje perpendicular al plano $x - y$. Es una de las señales del controlador junto a v_{rob} .
- $\omega_{L\,rob}$ es la velocidad angular que debe tener la rueda izquierda. Se calcula a partir de las señales de control v_{rob} y ω_{rob} .
- $\omega_{R\,rob}$ es la velocidad angular que debe tener la rueda derecha. Se calcula a partir de las señales de control v_{rob} y ω_{rob} .

Se comienza por calcular las distancias recorridas por cada rueda del robot con las ecuaciones (2.95) y (2.96).

$$D_L = N_L 2\pi R = \left(\frac{\Delta pulsos}{PPR} \right) 2\pi R \quad (2.95)$$

$$D_R = N_R 2\pi R = \left(\frac{\Delta pulsos}{PPR} \right) 2\pi R \quad (2.96)$$

Se procede a obtener los incrementos del ángulo de orientación y de la distancia recorrida por el robot, esto se muestra en las ecuaciones (2.97) y (2.98) respectivamente.

$$\Delta\phi = \frac{D_R - D_L}{L_{rob}} \quad (2.97)$$

$$\Delta S_{rob} = \frac{D_R + D_L}{2} \quad (2.98)$$

Mediante el incremento mostrado en la ecuación (2.97) se puede calcular el ángulo de rotación que tiene el robot en cada instante del tiempo, esto se muestra en la ecuación (2.99).

$$\phi_k = \phi_{k-1} + \Delta\phi \quad (2.99)$$

Posteriormente se puede obtener los valores de las coordenadas del centro de masa del robot y de su punto en adelante a_{rob} . Las coordenadas del centro de masa se muestran en las ecuaciones (2.100) y (2.101). Y las coordenadas del punto en adelante se muestra en las ecuaciones (2.102) y (2.103).

$$x_{rob} = x_{rob\ k-1} + \Delta S_{rob} \cos \phi \quad (2.100)$$

$$y_{rob} = y_{rob\ k-1} + \Delta S_{rob} \sin \phi \quad (2.101)$$

$$h_{x\ rob} = x_{rob\ k} + (a_{rob} \cos \phi) \quad (2.102)$$

$$h_{y\ rob} = y_{rob\ k} + (a_{rob} \sin \phi) \quad (2.103)$$

Las ecuaciones anteriores se programan en el microcontrolador del robot, al obtener las coordenadas y orientación reales del robot mediante odometría se puede programar la ley de control de la ecuación (2.94). Con los valores de la velocidad lineal v_{rob} y la velocidad angular del robot ω_{rob} se puede obtener las velocidades angulares que debe tener cada rueda para seguir la trayectoria deseada. Las velocidades angulares de cada rueda se muestran en las ecuaciones (2.104) y (2.105).

$$\omega_{L\ rob} = \frac{(v_{rob} - \frac{L_{rob}}{2}\omega_{rob})}{R}$$

(2.104)

$$\omega_{R\ rob} = \frac{(v_{rob} + \frac{L_{rob}}{2}\omega_{rob})}{R}$$

(2.105)

2.6. Implementación del controlador para seguimiento de trayectoria

2.6.1. Implementación de la odometría

Una vez definido el control para el seguimiento de trayectoria es necesario realizar dos acciones: implementar la odometría en el robot y el controlador basado en técnicas de Lyapunov. Para poder implementar las ecuaciones (2.95) y (2.96) es necesario obtener el valor del *PPR*. Este valor se obtiene de forma experimental, se establecen las interrupciones activadas por cada pulso del encoder. Se realiza una modificación al código del robot para acumular el número de pulsos conforme se gira manualmente cada rueda. De esta manera se puede girar las ruedas exactamente una vuelta y mediante el monitor serial visualizar el número de *PPR* necesario. En (2.106) se establece el valor del *PPR* obtenido mediante estas pruebas.

$$PPR = 315 \text{ (pulsos por vuelta)} \quad (2.106)$$

De esta manera, se puede seguir haciendo cambios al código para poder acumular distancia recorrida por cada rueda. Realizando esta prueba se puede tener un control de las vueltas que da cada rueda y verificar en el monitor serial que dicha distancia sea la correcta. Otro aspecto importante que considerar es que originalmente el robot a pesar de tener motores con dos canales en los encoders, solo tiene un canal, canal A, conectado a la placa del microcontrolador. Como solución se desmonta el sensor ultrasónico, ya que no se requiere para el desarrollo de las pruebas, y se obtienen un conjunto de pines libres que se pueden utilizar para el canal B de cada motor. Posteriormente se realizan las uniones de el canal B de cada motor con los pines disponibles. Con esta modificación al hardware se procede a modificar el código de las interrupciones para que pueda detectar el sentido de giro de las ruedas, siendo distancias negativas cuando las ruedas se mueven hacia atrás. En la tabla (2.1) se muestran los resultados de las primeras pruebas para validar la implementación de la odometría. Se debe tener en cuenta que se usa de referencia la distancia que recorre cada rueda al girarlas manualmente, siendo 0.105 m y 0.21 m lo correspondiente a media circunferencia y a una circunferencia completa respectivamente. De esta manera se puede comparar los datos obtenidos mediante el monitor serial con los obtenidos mediante las ecuaciones de la odometría. Los resultados son casi idénticos a los calculados.

Se realizan pruebas adicionales considerando movimientos hacia atrás de ambas ruedas, resultando los valores mostrados en la tabla (2.2). Estos valores se comprobaron con los calculados y resultaron prácticamente idénticos.

D_L medido	0	0	0.105	0.21	0.105	0.21
D_L calculado	0	0	0.105	0.21	0.105	0.21
D_R medido	0.105	0.21	0	0	0.105	0.21
D_R calculado	0.105	0.21	0	0	0.105	0.21
S_{rob} medido	0.05	0.105	0.05	0.105	0.105	0.21
S_{rob} calculado	0.05	0.105	0.05	0.105	0.105	0.21
ϕ medido	0.74	1.48	-0.74	-1.48	0	0
ϕ calculado	0.742	1.484	-0.742	-1.484	0	0
x_{rob} medido	0.05	0.07	0.05	0.07	0.10	0.21
x_{rob} calculado	0.05	0.07	0.05	0.07	0.105	0.21
y_{rob} medido	0.02	0.07	-0.02	-0.07	0	0.01
y_{rob} calculado	0.02	0.07	-0.02	-0.07	0	0
$h_{x_{rob}}$ medido	0.2	0.09	0.19	0.09	0.30	0.41
$h_{x_{rob}}$ calculado	0.2	0.09	0.19	0.09	0.305	0.42
$h_{y_{rob}}$ medido	0.15	0.26	-0.16	-0.26	0.01	0.02
$h_{y_{rob}}$ calculado	0.15	0.26	-0.16	-0.26	0.01	0

Tabla 2.1. Resultados de las pruebas para validar la odometría implementada.

Con las pruebas anteriores se comprueba la correcta implementación de la odometría.

2.6.2. Implementación de los controles de equilibrio y de seguimiento de trayectoria

Una vez que se comprueba el funcionamiento de la odometría, se procede a implementar el controlador para el seguimiento de trayectoria en conjunto con el control de equilibrio. El control de equilibrio y el control de seguimiento de trayectoria representan dos objetivos con naturaleza significativamente distintas. Por ejemplo, la velocidad del robot en el control de equilibrio debe ser cero, pero para seguir una trayectoria es diferente de cero. Tal como se ha reportado en [33] en sistemas subactuados estos objetivos de control al entrar en conflicto impide que un controlador continuo único pueda simultáneamente estabilizar la plataforma y seguir una referencia general. Este problema fue formalizado matemáticamente por Brockett en [34] quien mostró que muchos sistemas no holonómicos no admiten estabilización asintótica mediante control continuo, abriendo la puerta al uso de control discontinuo, híbrido, o por conmutación. En este trabajo de investigación se utiliza la técnica de conmutar ambos controladores. El fundamento formal de la conmutación de los controladores que se utiliza en este trabajo se basa en [35], donde los autores muestran que es posible mantener la estabilidad global si se cumplen los siguientes puntos:

D_L medido	0	0	0	0	0.105	0.21	-0.105	-0.21	0.105	0.21
D_L calculado	0	0	0	0	0.105	0.21	-0.105	-0.21	0.105	0.21
D_R medido	0.105	0.21	-0.105	-0.21	0	0	0	0	0.105	0.21
D_R calculado	0.105	0.21	-0.105	-0.21	0	0	0	0	0.105	0.21
ϕ medido	0.74	1.48	-0.73	-1.48	-0.74	-1.48	0.73	1.47	0	0
ϕ calculado	0.742	1.484	-0.742	-1.484	-0.742	-1.484	0.742	1.484	0	0
x_{rob} medido	0.05	0.07	-0.05	-0.07	0.05	0.07	-0.05	-0.07	0.10	0.21
x_{rob} calculado	0.05	0.07	-0.05	-0.07	0.05	0.07	-0.05	-0.07	0.10	0.21
y_{rob} medido	0.02	0.07	0.02	0.07	-0.02	0.07	-0.02	-0.07	0	0
y_{rob} calculado	0.02	0.07	0.02	0.07	-0.02	0.07	-0.02	-0.07	0	0
$h_{x_{rob}}$ medido	0.2	0.09	0.10	-0.05	0.19	0.09	0.10	-0.05	0.30	0.41
$h_{x_{rob}}$ calculado	0.2	0.09	0.10	-0.05	0.19	0.09	0.10	-0.05	0.305	0.41
$h_{y_{rob}}$ medido	0.15	0.26	-0.11	-0.13	-0.16	-0.26	0.11	0.13	0	0
$h_{y_{rob}}$ calculado	0.15	0.26	-0.11	-0.13	-0.16	-0.26	0.11	0.13	0	0

Tabla 2.2. Resultados para validar el sentido de giro de las ruedas en la odometría.

- La conmutación es diseñada adecuadamente.
- Los controladores son individualmente estabilizantes.
- Existen funciones de Lyapunov múltiples o equivalentes.
- El cambio entre controladores evita incompatibilidades internas.

Para abordar esta problemática se propone una estrategia que consiste en conmutar ambos controladores. En robótica, ya hay registros de conmutación entre algoritmos de control como en [36] donde se tiene un robot llamado Acrobot, que es un sistema subactuado ya que solo tiene un motor acoplado en el hombro y debe mantener en equilibrio ambos eslabones en forma vertical. Por lo tanto, se conmuta entre un control de equilibrio y

otro de seguimiento de trayectoria, pero se debe establecer una jerarquía entre ambos controladores. El controlador que debe ser predominante es el de equilibrio, ya que si no hay equilibrio en el robot el sistema será inestable. Para asegurar esta jerarquía se establece que el control de equilibrio se ejecute más veces que el de seguimiento de trayectoria. En el capítulo (4) se establecen los tiempos de muestreo utilizados para cada caso de conjuntos de controladores.

Los conjuntos de controladores se dividen en dos:

- Controlador R1. PI-PD (control de equilibrio) y control basado en técnicas de Lyapunov (seguimiento de trayectoria). Es necesario recalcar que el controlador PI-PD que se usa en este caso tiene una sintonización distinta al control PI-PD implementado de fábrica. Esta nueva sintonización fue necesaria para permitir la implementación del controlador de seguimiento de trayectoria.
- Controlador R2. Retroalimentación de estados (control de equilibrio) y control basado en técnicas de Lyapunov (seguimiento de trayectoria).

RESULTADOS EN SIMULACIÓN

3.1. Introducción

El presente capítulo muestra los resultados en simulación obtenidos.

- En la sección 3.2, se muestran las simulaciones realizadas del modelo y el controlador por retroalimentación de estados. Solo se muestran las simulaciones del modelo obtenido en la subsección 1.2.3. El control de equilibrio se simula de la siguiente manera:
 - En la subsección 3.2.1, se muestran las simulaciones realizadas del modelo en lazo abierto y con el controlador por retroalimentación de estados en tiempo continuo. Corresponde al controlador obtenido en la subsección 2.3.1.
 - En la subsección 3.2.2, se muestran las simulaciones realizadas del modelo en lazo abierto y con el controlador por retroalimentación de estados en tiempo discreto. Corresponde al controlador obtenido en la subsección 2.3.2.
- En la sección 3.3, se muestran las simulaciones realizadas para el modelo y el controlador para el seguimiento de trayectoria. Corresponde al modelo y controlador desarrollado en la sección 2.5.

3.2. Resultados del controlador por retroalimentación de estados

3.2.1. Resultados de la simulación del controlador en tiempo continuo

Una vez que se ha implementado de manera correcta el control por retroalimentación de estados se procede a analizar los resultados obtenidos. Primero se analiza el desempeño del controlador en simulación. Esta se realiza en Simulink tanto en lazo abierto como en lazo cerrado. El esquema para simular el sistema en lazo abierto se muestra en la figura (3.1).

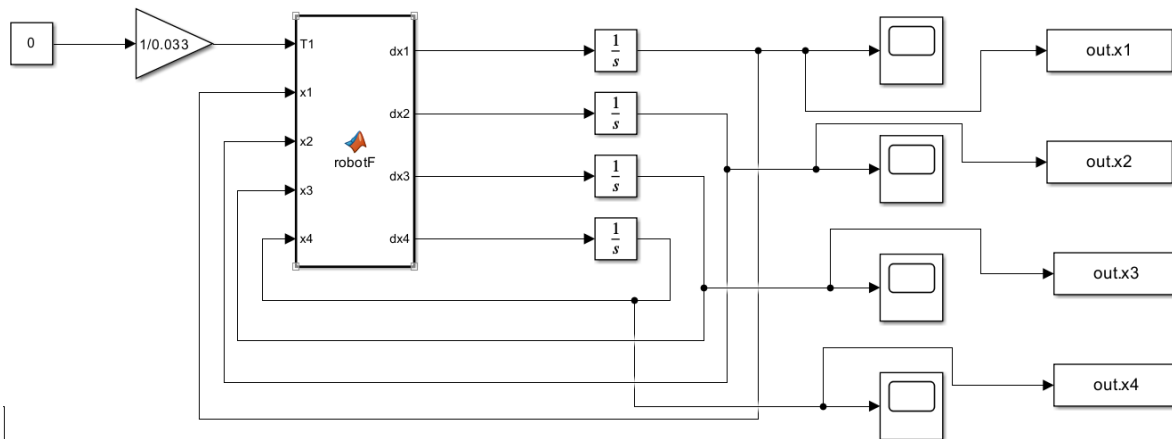


Figura 3.1. Esquema del sistema en lazo abierto en Simulink.

Los estados al realizar esta simulación muestran el comportamiento mostrado en la figura (3.2).

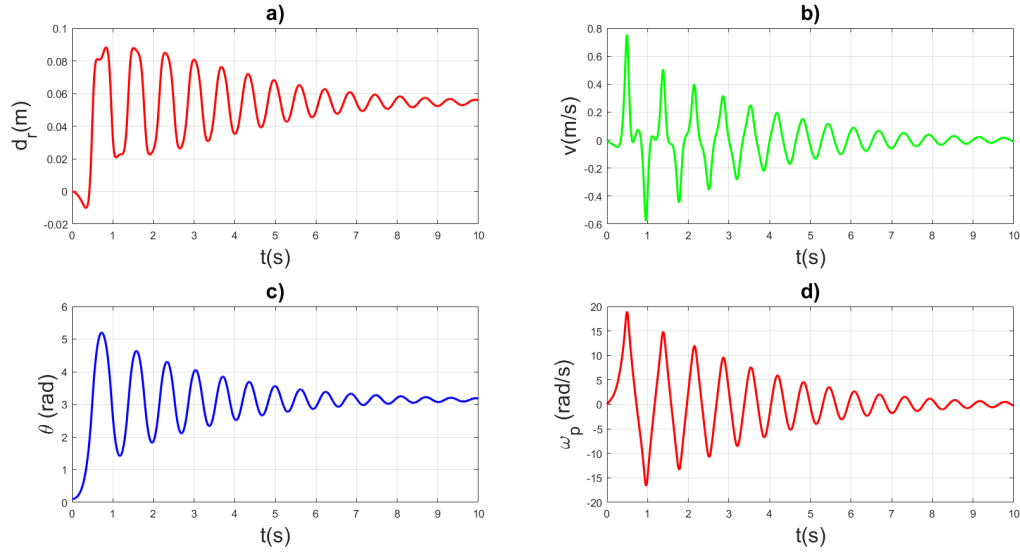


Figura 3.2. Respuestas del sistema en lazo abierto en tiempo continuo: a) Distancia recorrida. b) Velocidad lineal. c) Ángulo de inclinación. d) Velocidad angular.

El sistema en lazo cerrado se muestra en la figura (3.3). Se puede observar de manera explícita la retroalimentación que existe en los estados del sistema.

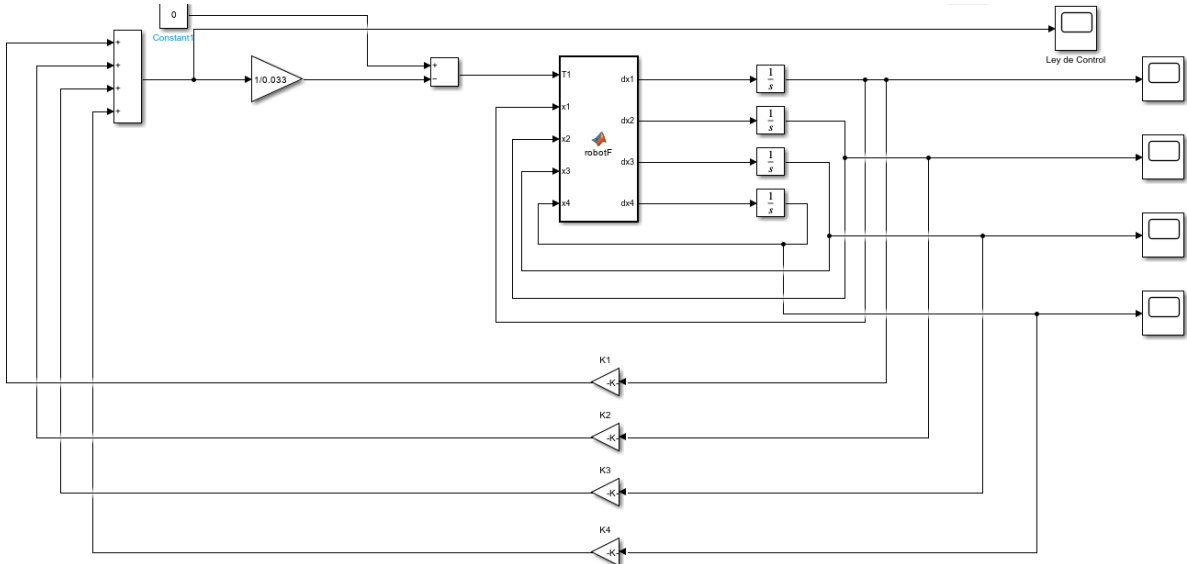


Figura 3.3. Esquema del sistema en lazo cerrado en Simulink

Al ejecutar la simulación en lazo cerrado se obtienen las gráficas mostradas en la figura (3.4). Las constantes usadas para este controlador corresponden a las obtenidas en la ecuación (2.65), cuándo se diseñó el controlador por retroalimentación de estados en tiempo continuo.

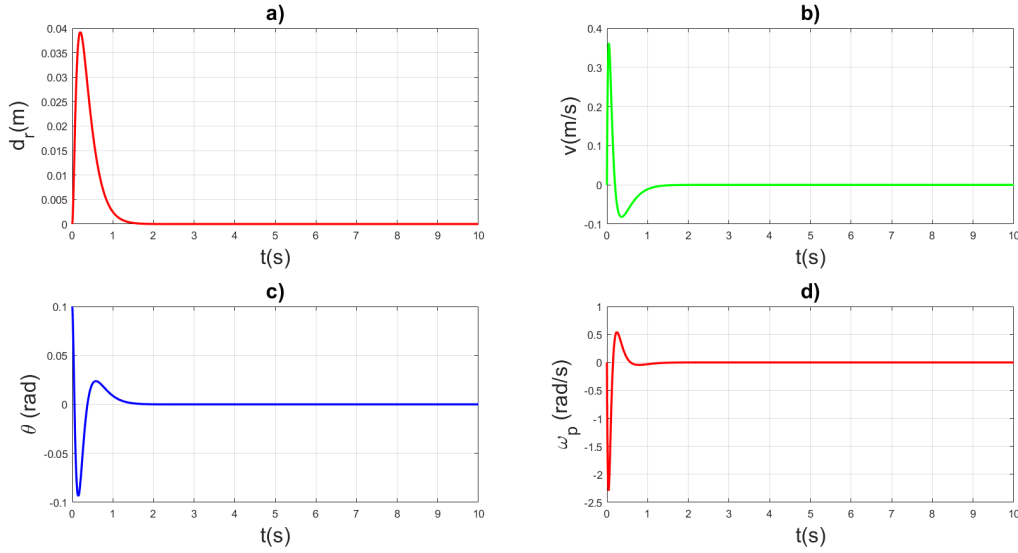


Figura 3.4. Respuestas del sistema en lazo cerrado en tiempo continuo: a) Distancia recorrida. b) Velocidad lineal. c) Ángulo de inclinación. d) Velocidad angular

Es importante observar que todos los estados convergen al valor de 0 que es el punto de equilibrio que se utilizó para linealizar el sistema. También se recalca que el ángulo de inclinación tiene condiciones iniciales de 0.1 radianes y las oscilaciones al estabilizarse son menores a este valor. El que las oscilaciones no superen el valor de la condición inicial garantiza que el comportamiento en las pruebas experimentales sea bueno. Asimismo, el tiempo de estabilización es de alrededor de 1 segundo , siendo un tiempo apropiado ya que el sistema es mecánico que requiere vencer la inercia y gravedad en sus movimientos.

Otro aspecto por analizar es la gráfica de la ley de control, esto permite entender el esfuerzo del controlador en hacer que el sistema se comporte de la manera deseada. Esta gráfica se observa en la figura (3.5).

La ley de control muestra un par máximo de aproximadamente 0.37 Nm el cuál lo deben aportar los dos motores y cada uno cuenta con un par máximo de 0.049 Nm pero contando la caja reductora el par que puede aplicar cada motor es de 1.47 Nm siendo capaz de aportar el par necesario para el control.

3.2.2. Resultados de la simulación del controlador en tiempo discreto

Se procede a mostrar los resultados de las simulaciones realizadas para el sistema y control en tiempo discreto. En la figura (3.6) se muestra el esquema en Simulink del sistema en

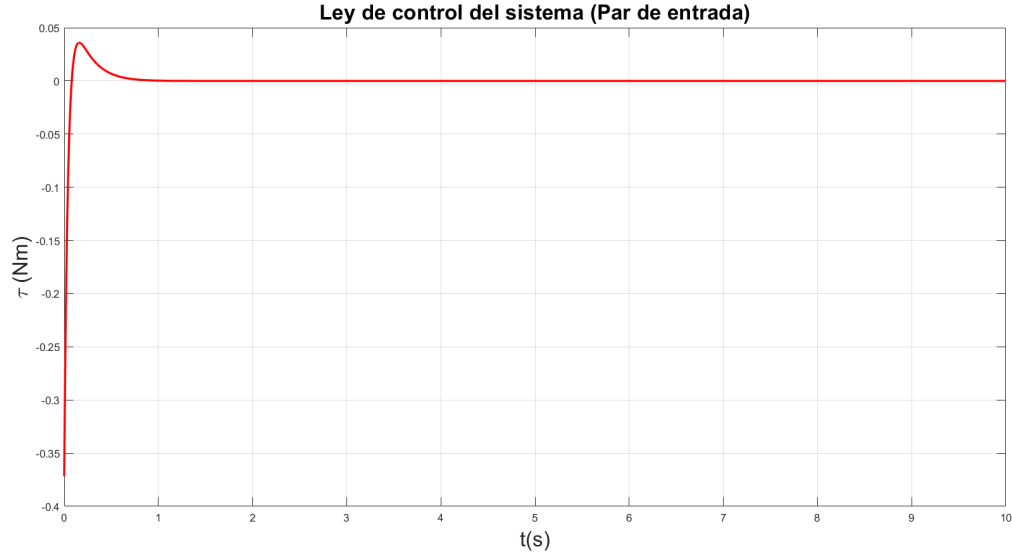


Figura 3.5. Ley de control del sistema en lazo cerrado en tiempo continuo.

lazo abierto en tiempo discreto, se realiza esta simulación para verificar que el sistema discretizado tiene un comportamiento adecuado.

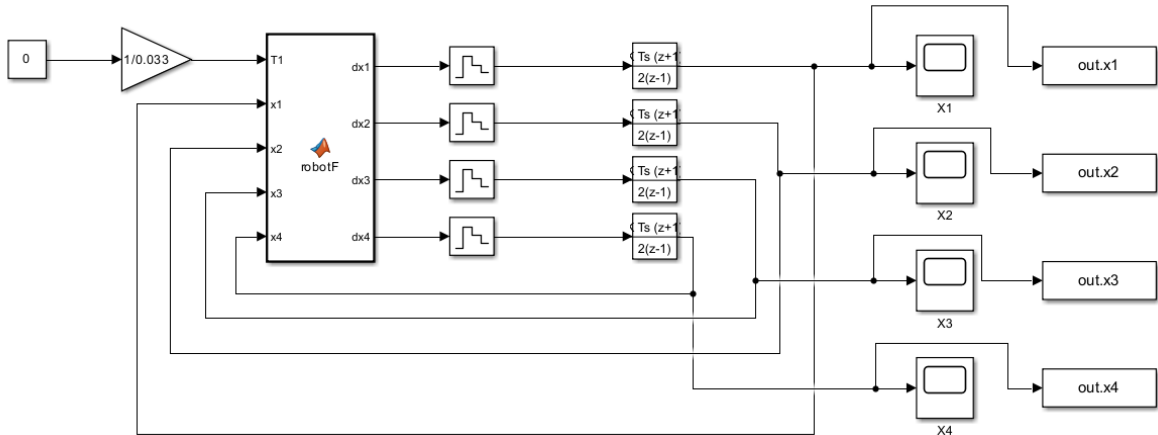


Figura 3.6. Esquema del sistema discretizado en lazo abierto en Simulink.

Al ejecutar la simulación en lazo abierto se obtienen los resultados de la figura (3.7). Debido a que el período de muestreo es del orden de los *ms* no se alcanzan a distinguir las muestras de los estados, más adelante se mostrará un acercamiento a las muestras.

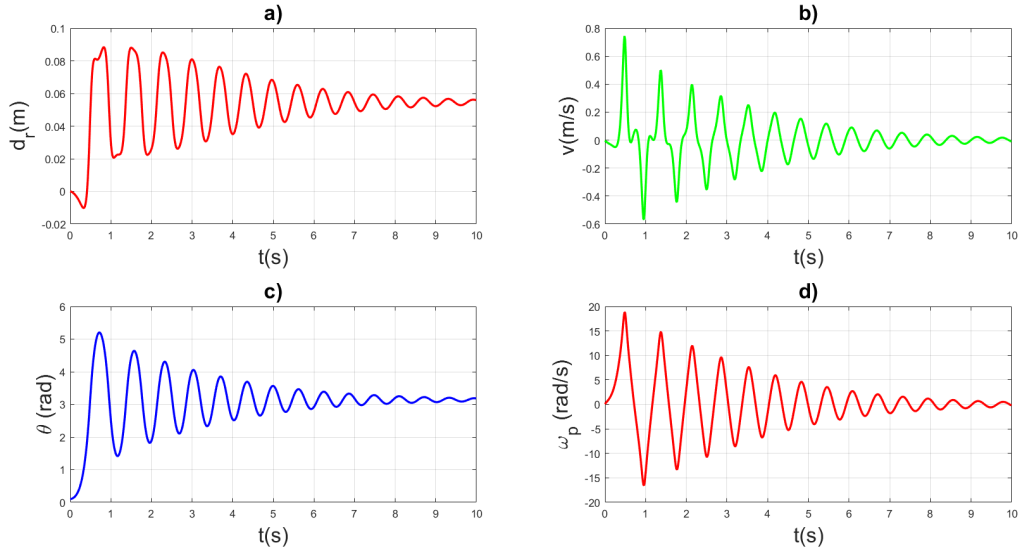


Figura 3.7. Respuestas del sistema en lazo abierto en tiempo discreto: a) Distancia recorrida. b) Velocidad lineal. c) Ángulo de inclinación. d) Velocidad angular.

En la figura (3.8) se muestra el diagrama utilizado en Simulink para simular el sistema en lazo cerrado considerando las constantes de retroalimentación obtenidas en la ecuación (2.78). Se puede observar que se usa el bloque del retenedor de orden cero para realizar la discretización usando un período de muestreo de 15 ms .

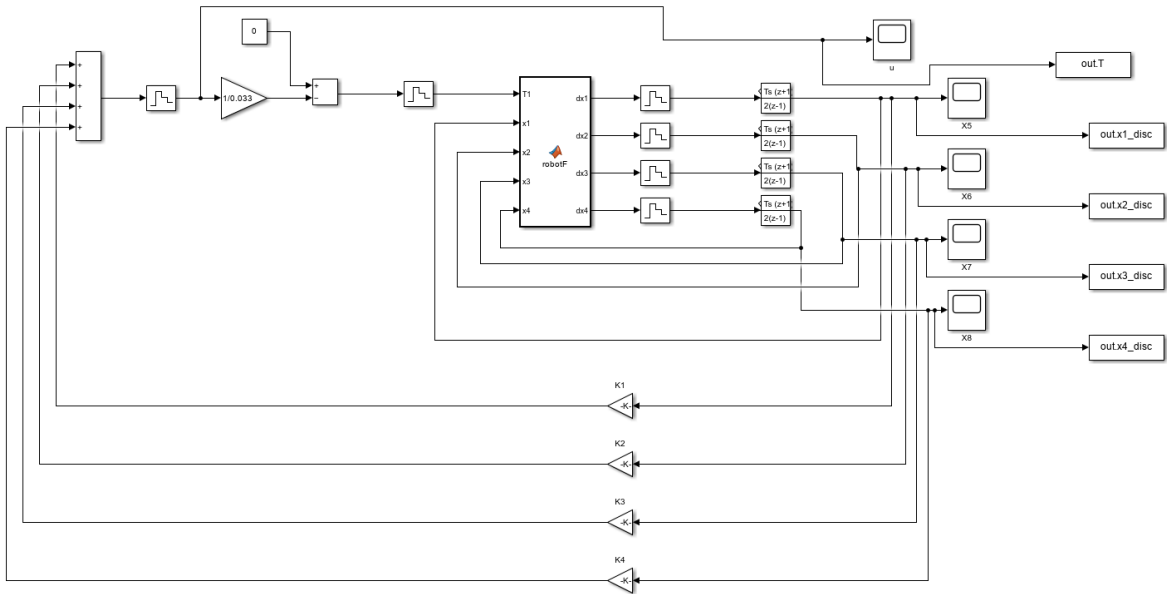


Figura 3.8. Esquema del sistema discretizado en lazo abierto en Simulink.

En la figura (3.9) se muestra los estados del sistema discreto una vez que se ha aplicado el control.

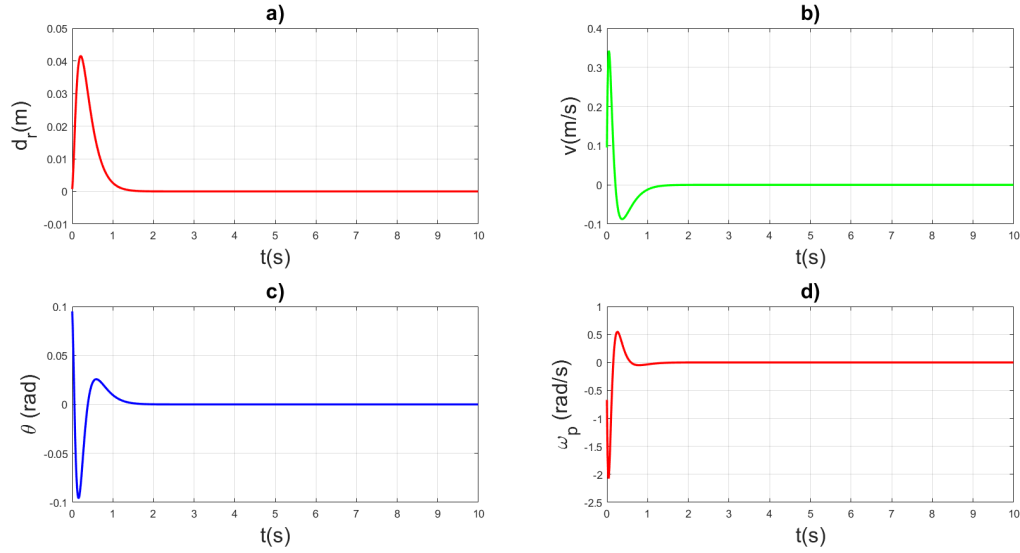


Figura 3.9. Respuestas del sistema en lazo cerrado en tiempo discreto: a) Distancia recorrida. b) Velocidad lineal. c) Ángulo de inclinación. d) Velocidad angular.

La ley de control para el caso discreto se muestra en la figura 3.10, mientras que en la figura (3.11) se presentan las muestras que conforman dicha ley de control.

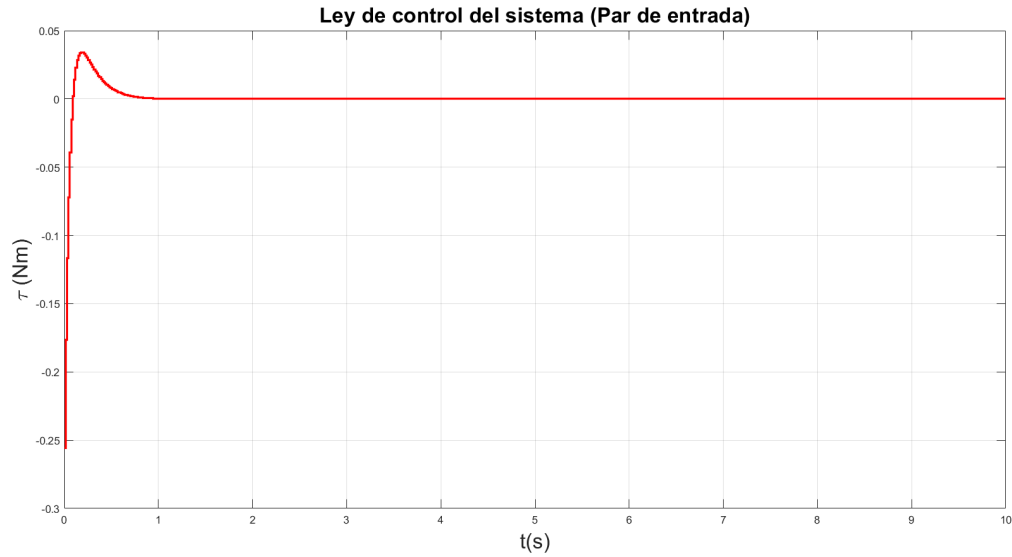


Figura 3.10. Ley de control del sistema discretizado en lazo cerrado.

En este caso el par máximo necesario para controlar el sistema es de 0.25 Nm que es menor al caso en tiempo continuo por lo tanto los motores siguen siendo capaces de aportar este par.

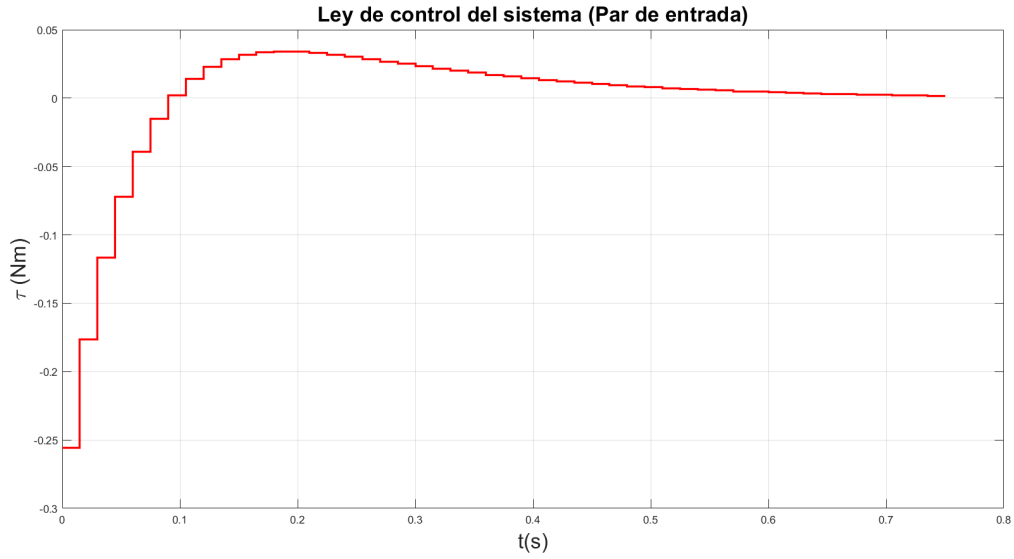


Figura 3.11. Muestras de la ley de control del sistema discretizado en lazo cerrado.

3.3. Resultados en simulación para el seguimiento de trayectoria

La simulación para probar el controlador para seguimiento de trayectoria se muestra en la figura (3.12) y las velocidades lineales y angulares se muestran en (3.13) la referencia que se decide usar es una trayectoria circular.

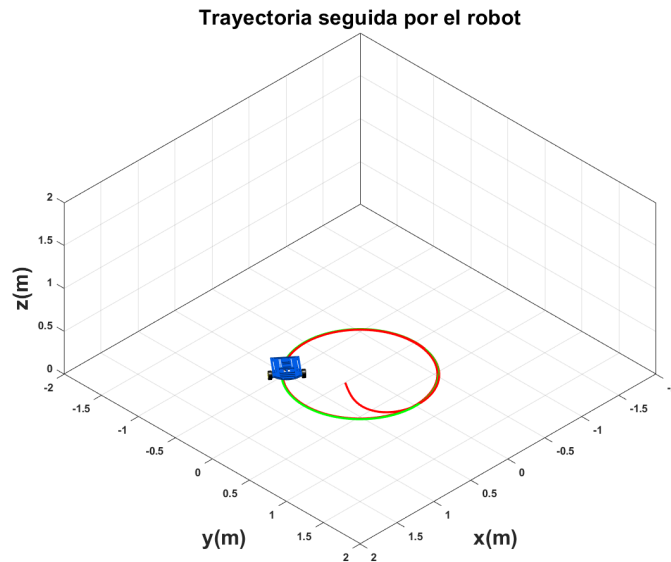


Figura 3.12. Simulación del seguimiento de trayectoria del robot.

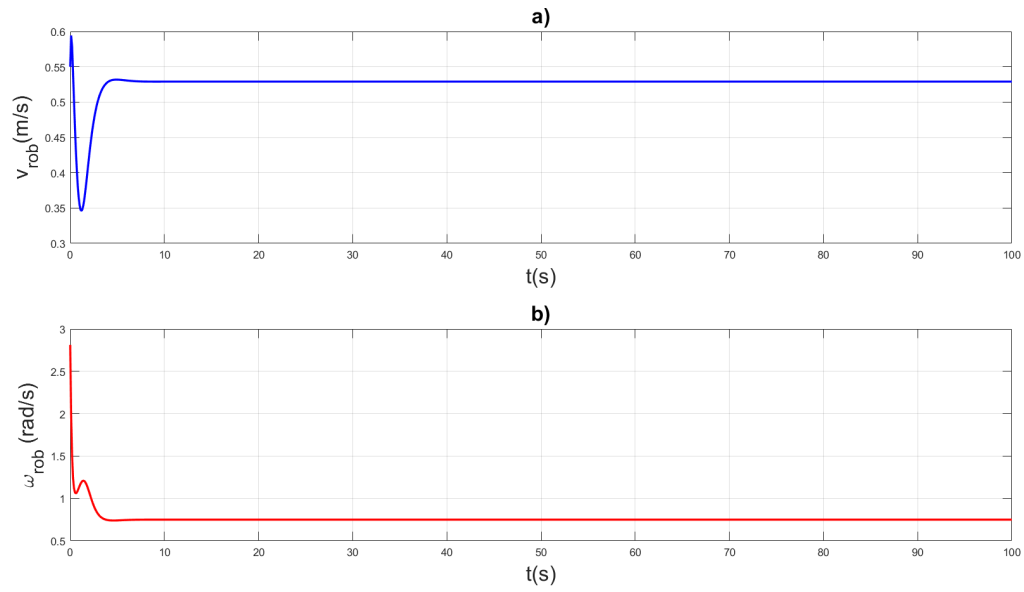


Figura 3.13. Velocidades en simulación del seguimiento de trayectoria del robot.

DESCRIPCIÓN DE LA PLATAFORMA Y PRUEBAS EXPERIMENTALES

4.1. Introducción

En el presente capítulo se muestra una descripción del robot que se utiliza así como los resultados experimentales obtenidos. Todos los controladores analizados son programados en lenguaje C++ en el microcontrolador implementado en el robot. Los algoritmos de control se implementan en la rutina de servicio de interrupción del timer. Además, es necesario clasificar los controladores que se comparan, siendo los siguientes:

- Los controladores que solo mantienen el equilibrio del robot son los siguientes:
 - Controlador E1, se refiere al controlador de fábrica del robot. Siendo un controlador PI-PD.
 - Controlador E2, se refiere al controlador por retroalimentación de estados desarrollado en la sección 2.3.2.
- Los conjuntos de controladores que mantienen el equilibrio del robot a la par que realiza el seguimiento de trayectoria son los siguientes:
 - Conjunto de controladores R1, se refiere al control de equilibrio PI-PD modificado junto con el controlador basado en técnicas de Lyapunov para el seguimiento de trayectoria.
 - Conjunto de controladores R2, se refiere al control de equilibrio por retroalimentación de estados junto con el controlador basado en técnicas de Lyapunov para el seguimiento de trayectoria.

El contenido se divide en las siguientes secciones:

- En la sección 4.2, se describe el robot que se usa para las pruebas experimentales.
 - En la subsección 4.2.1, se muestra una descripción del filtro de Kalman usado para filtrar las señales obtenidas mediante la unidad MPU.
- En la sección 4.3, se muestra el resultado de implementar el controlador por retroalimentación de estados para mantener el equilibrio.
- En la sección 4.4, se muestra la comparación de los controladores de equilibrio E1 y E2.
- En la sección 4.5, se muestran los resultados al implementar en conjunto el controlador de equilibrio junto al de seguimiento de trayectoria.
- En la sección 4.6, se realiza la comparación entre los conjuntos de los controladores de equilibrio y de seguimiento de trayectoria R1 y R2.

4.2. Descripción del robot a utilizar, Elegoo Tumbler

Este robot cuenta con una tarjeta basada en el microcontrolador ATmega328, que funciona como controlador. El robot consta de dos ruedas, cada una con un motor de corriente directa, de esta manera el robot puede moverse en línea recta o girar sobre su propio eje. También tiene una estructura mecánica de metal y acrílico donde se coloca la tarjeta que contienen los componentes para garantizar su funcionamiento. Debido a que el controlador no tiene una salida de corriente elevada, es necesario un driver que pueda enviar la cantidad adecuada de corriente a los motores. El driver para los motores es el TB6612FENG, el cual tiene un puente MOSFET-H de alta corriente y dos canales conducen los dos motores a la vez.

El uso de un driver para los motores también ofrece las siguientes ventajas.

- Alimenta con la suficiente corriente a los motores.
- Cumple la función de aislante para evitar que se dañe el controlador por alguna corriente.

El robot es un sistema subactuado, ya que los únicos actuadores disponibles son los dos motores. Estos se controlan exclusivamente mediante el ajuste de su velocidad y su sentido de giro, regulados a través de una señal PWM que actúa como voltaje de alimentación. Para obtener estas magnitudes se emplean dos sensores Hall (encoders), uno por motor,

que permiten medir la velocidad de rotación y su dirección. Una vista general del robot se muestra en la figura (4.1).



Figura 4.1. Robot Elegoo Tumbler

Las principales conexiones del driver y el controlador son las siguientes.

Motor A

- El PWMA está conectado a la terminal D5 de la tarjeta del controlador.
- El AIN1 está conectado a la terminal D7 de la tarjeta del controlador.

Motor B

- El PWMB está conectado a la terminal D6 de la tarjeta del controlador.
- El BIN1 está conectado a la terminal D12 de la tarjeta del controlador.

La tabla (4.1) muestra la configuración de las señales que debe tener las terminales para poder mover los motores de determinada manera. En la figura (4.4) se muestra un diagrama de las conexiones generales del microcontrolador con los sensores y motores del robot.

Tabla de verdad			
AIN1	0	1	0
AIN2	0	0	1
	Detener	Adelante/Rotar	Reversa

Tabla 4.1. Tabla de verdad para el driver de los motores.

Se puede observar que cuando el motor se mueve, AIN1 y AIN2 siempre son contrarios. Por esto, para salvar los recursos del IO, se agrega el inversor de dos canales SN74LVC2G14. La entrada esta conectada a AIN1 y la salida esta conectada al AIN2, de esta manera la entrada y la salida siempre tendrán los niveles invertidos. Lo mismo se hace para BIN1 Y BIN2. Otro aspecto importante de los motores es su relación de transformación, la cual es de 1:30, lo que significa que la velocidad de entrada es 30 y la salida del eje es 1 revolución.

El robot dispone de un conjunto de programas en los que se definen las asignaciones de pines de la tarjeta controladora, las operaciones matemáticas necesarias, un filtro de Kalman para reducir el ruido de las mediciones de la unidad inercial y un conjunto de controladores PD-PI que permiten al robot mantener el equilibrio de forma adecuada.

Para controlar el autobalance del robot es necesario medir el ángulo de inclinación y la velocidad angular del cuerpo del robot. Estos parámetros son medidos por la unidad inercial MPU6050, que cuenta con acelerómetro y giroscopio. Las terminales se conectan de la siguiente manera.

- El SCL del MP6050 está conectada a la terminal 9 (A5) del controlador.
- El SDA del MP6050 está conectada a la terminal 8 (A4) del controlador.

El sistema de control del robot se compone de un controlador PD para la tarea de balance, un controlador PI para regular la velocidad de los motores y un controlador PD adicional para el movimiento de giro sobre su propio eje. Se emplea un controlador PD en el lazo de balance porque este requiere una respuesta rápida y la estimación de la velocidad del cuerpo, de modo que el controlador PI de los motores pueda generar la velocidad necesaria para compensar dicha dinámica. Este proceso se ilustra en la figura (4.2). Cuando el robot se encuentra en la posición A, los motores permanecen inmóviles debido a que el sistema está en equilibrio. En la posición B, las ruedas deben girar hacia la izquierda con una velocidad superior a la velocidad de caída del cuerpo para recuperar el equilibrio. De manera similar, en la posición C los motores deben mover las ruedas hacia la derecha, también a mayor velocidad que el cuerpo, para restablecer la postura vertical.

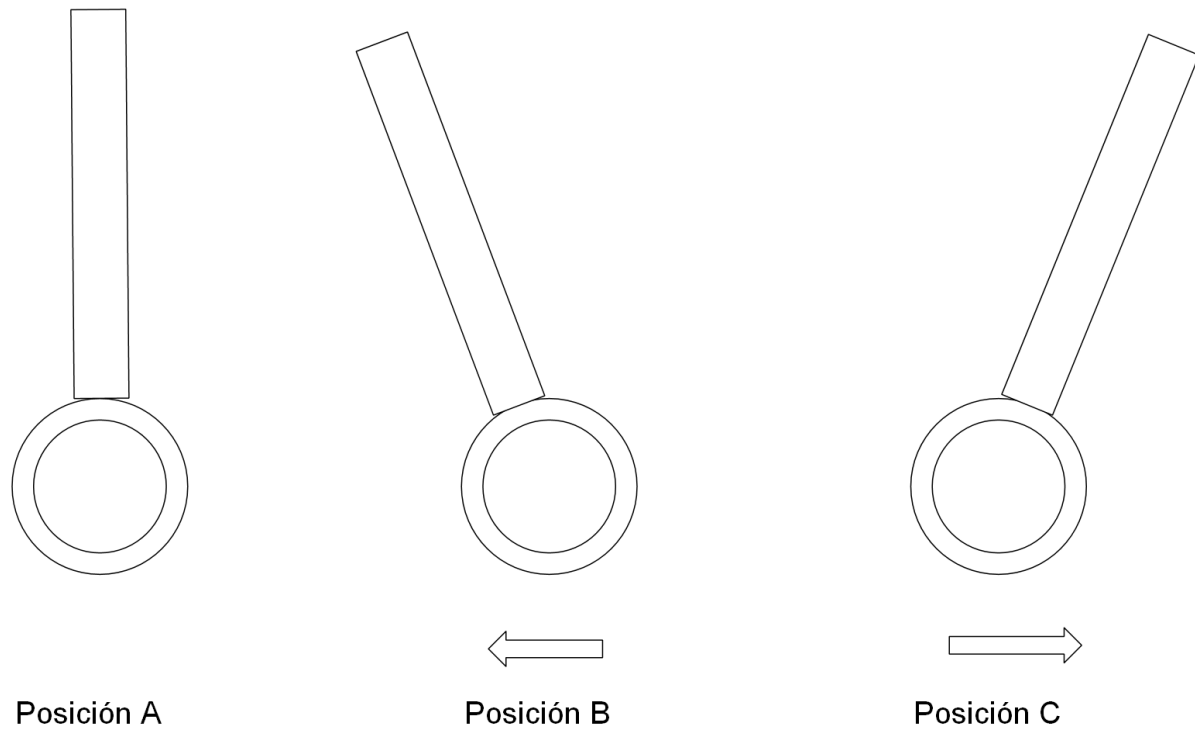


Figura 4.2. Posiciones del robot Elegoo Tumbler.

Las señales generadas por cada controlador se combinan para obtener una única ley de control, la cual determina el ciclo de trabajo de la señal PWM que alimenta a cada motor. La figura (4.3) muestra la estructura general de la programación de los controladores del robot.

El acelerómetro y el giroscopio del MPU6050 presentan ventajas y limitaciones complementarias. El acelerómetro no sufre error acumulado y permite obtener el ángulo de inclinación mediante el cálculo de la tangente; sin embargo, sus mediciones contienen un nivel elevado de ruido, ya que la aceleración varía con el movimiento del robot y las vibraciones generadas por los motores. Por ello, sus valores no pueden utilizarse directamente. El giroscopio, por su parte, es menos sensible a las vibraciones externas y ofrece alta precisión en la medición de la velocidad angular, pero el ángulo de inclinación obtenido por integración presenta error acumulado con el tiempo. Para obtener mediciones confiables tanto del ángulo de inclinación como de la velocidad angular del cuerpo, se emplea un filtro de Kalman, previamente implementado en el robot. Este filtro combina de manera óptima las lecturas del acelerómetro y del giroscopio, reduciendo el ruido y compensando el error acumulado, lo que permite estimar con mayor precisión la inclinación y la velocidad del robot.

Además, el robot cuenta con un sensor ultrasónico y dos sensores infrarrojos que le

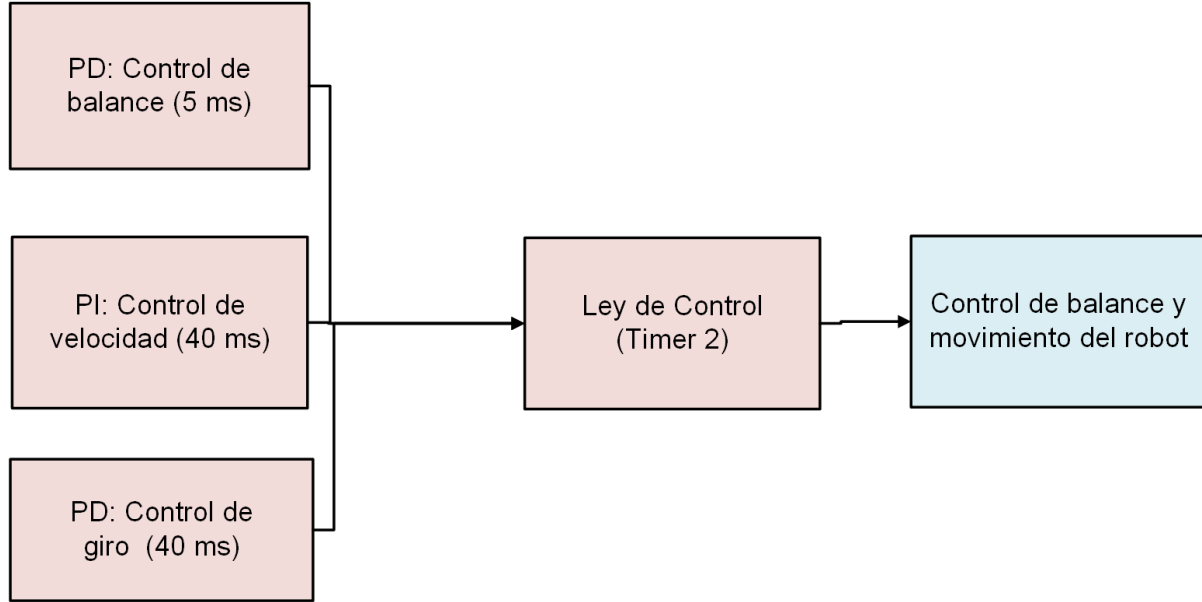


Figura 4.3. Controles del robot Elegoo Tumbler.

permiten localizar objetos que se encuentren cerca de él. De esta manera se puede programar al robot para que pueda seguir un objeto que este cercano a él o bien que se vaya alejando del objeto. También dispone de un modo donde el robot se mueve libremente evitando los obstáculos que tenga alrededor.

4.2.1. Descripción del filtro de Kalman utilizado

El robot contiene un filtro de Kalman programado en una librería implementada de fábrica. El filtro de Kalman recibe como entradas el ángulo estimado por el acelerómetro y la velocidad angular medida por el giroscopio de la unidad MPU6050. El filtro predice la evolución del ángulo y corrige dicha predicción utilizando la medición del acelerómetro, compensando el sesgo del giroscopio, obteniendo estimaciones suaves y libres de ruido del ángulo de inclinación y de la velocidad angular, las cuales se emplean directamente como estados del sistema para los controladores. En la figura (4.5) se muestra el procedimiento que realiza el filtro de Kalman.

donde:

- θ_{acc} es el ángulo de inclinación calculado con los valores sin filtrar de la MPU.
- ω_{gyro} es la velocidad angular calculada con los valores sin filtrar de la MPU.

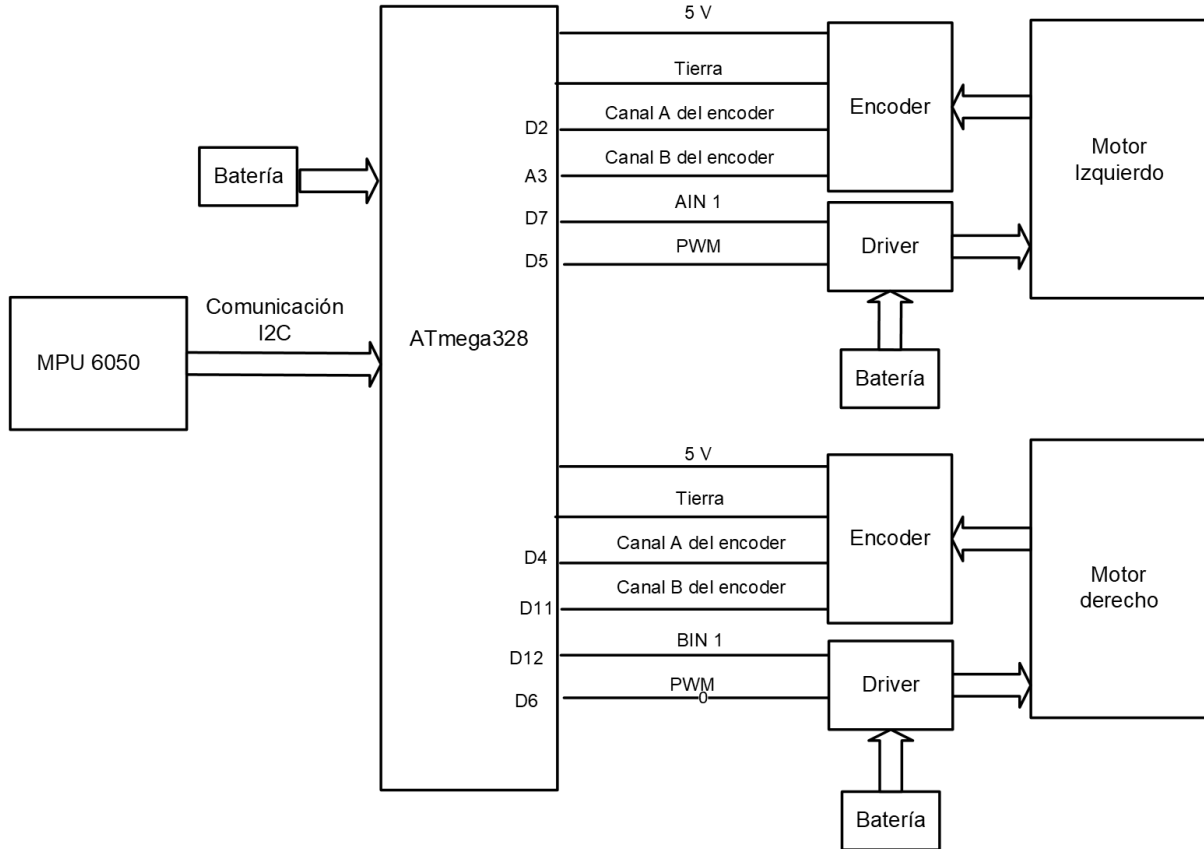


Figura 4.4. Conexiones generales del robot Elegoo Tumbler.

4.3. Resultados del control de equilibrio

Una vez que se evalúa el buen desempeño en simulación del controlador diseñado, se procede a implementarlo en el robot para controlar su equilibrio. Los resultados obtenidos fueron satisfactorios ya que el robot se mantuvo en equilibrio de manera adecuada. En la figura (4.6) se muestra el robot manteniéndose en equilibrio con el controlador por retroalimentación de estados, E2. El controlador diseñado E2, al igual que los demás controladores analizados, se implementa en la rutina de servicio de interrupción del timer.

4.4. Comparación entre controladores para el equilibrio

La comparación se lleva a cabo entre los controladores E1 y E2, mostrado en las figuras (4.7) y (4.8) respectivamente. El controlador E1 es el implementado de fábrica y el controlador E2 es el controlador por retroalimentación de estados.

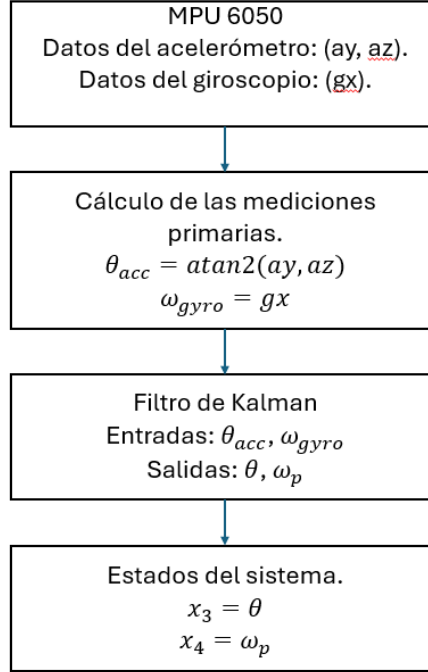


Figura 4.5. Filtro de Kalman implementado en el robot.

Para esta tarea se hace uso de la Integral del Error Cuadrático (IEC) ecuación (4.1), la Integral del Valor Absoluto de la Señal de Control (IVABSC) ecuación (4.2) y la Integral del Valor Absoluto de la Variación de la Señal de Control (IVAVSC) ecuación (4.3). Para esta tarea se usa de referencia [37] y [38]. 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.

$$IEC = \sum_{k=0}^N e[k]^2 \quad (4.1)$$

$$IVABSC = \sum_{k=0}^N |u[k]| \quad (4.2)$$

$$IVAVSC = \sum_{k=0}^N \left[\frac{u[k] - u[k-1]}{\Delta t} \right] \quad (4.3)$$

donde Δt es el período de muestreo utilizado.

Al calcular estos parámetros en los dos controladores se obtienen los resultados mostrados en la tabla (4.2). Se observa que en cuanto a los parámetros del IEC el controlador E1 tiene un desempeño mejor, por una diferencia de 43.6553 unidades. En cuanto al IVABSC ambos controladores son muy similares, teniendo una diferencia de 55.42 unidades para el

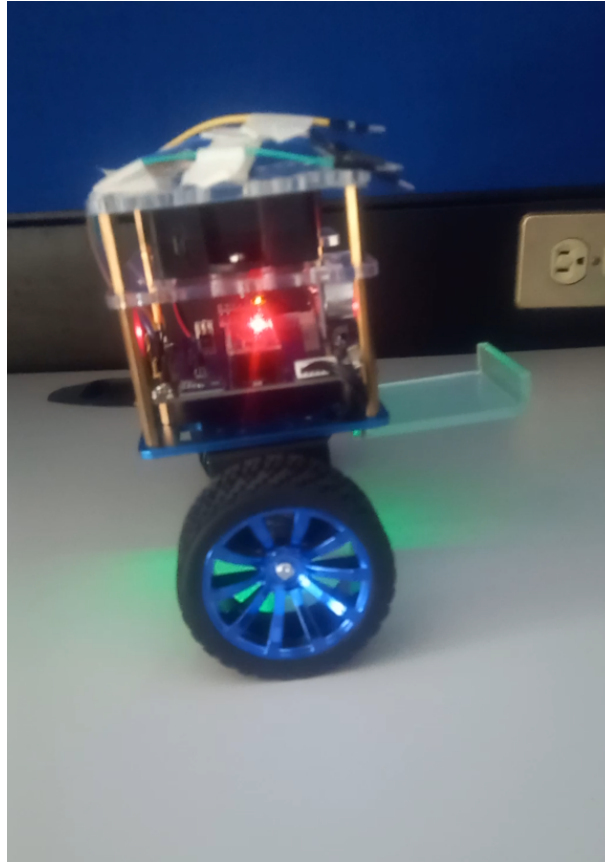


Figura 4.6. Robot manteniéndose en equilibrio.

motor derecho y 116.3 unidades para el motor izquierdo. Indicando que el tamaño de la ley de control en cada caso es similar ya que las diferencias es menor al 10 % de los valores del IVABSC. En el caso del IVAVSC el controlador por E2 tiene un desempeño mucho mejor ya que la variación de la ley de control no es tan brusca como en el controlador E1. Esto se indica ya que esta cantidad en el controlador modificado es menor del 50 % de los valores del controlador E1.

Se puede observar que los índices de desempeño no son muy diferentes a excepción del *IEC* y el *IVAVSC*. La diferencia en el *IEC* se debe a que, al ser una ecuación cuadrática del error, aunque los errores sean pequeños son penalizados. Para analizar a mayor profundidad esta diferencia entre los errores de los controladores es necesario revisar las figuras (4.9) y (4.10). Se puede observar que el ángulo de inclinación tiende a ser mayor en el caso del controlador E2, de ahí que el *IEC* sea más grande en este caso. Esto se debe a que los motores tienen un movimiento más suave y eso provoca que el ángulo tienda a ser más grande en algunos instantes.

Las gráficas de la ley de control del motor derecho se establecen en las figuras (4.11) y (4.12). Se puede observar que la ley de control del controlador E2, tiene cambios menos

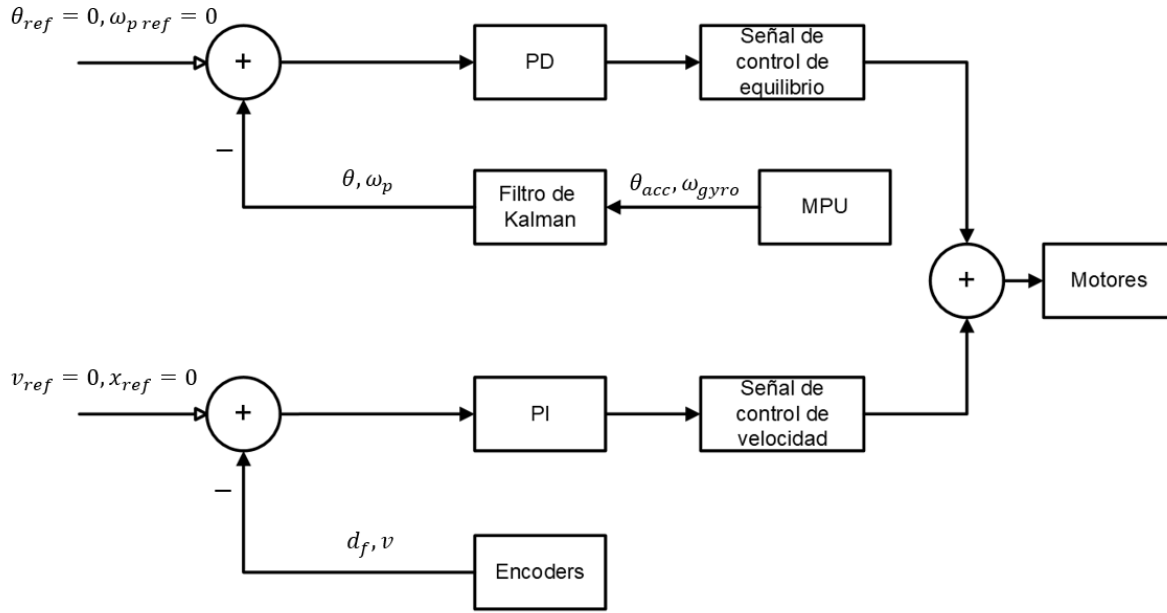


Figura 4.7. Control de equilibrio E1.

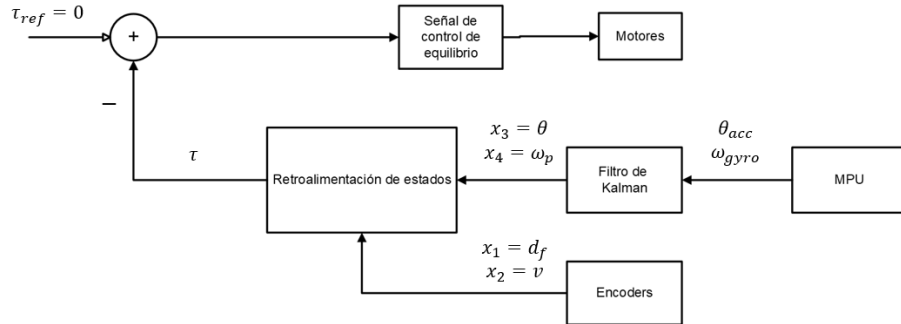


Figura 4.8. Control de equilibrio E2.

abruptos a comparación de la ley de control del controlador E1. Debido a esto es que el factor del *IVAVSC* es mucho menor en el controlador E2.

Las gráficas de la ley de control del motor izquierdo se establecen en las figuras (4.13) Y (4.14). Al igual que en el caso del motor derecho, aquí también se observa que la ley de control del controlador E2 provoca cambios menos abruptos en los cambios de sentido de giro del motor.

Asimismo, se debe comparar el tiempo de procesamiento en ambos casos. Esto permite saber si el microcontrolador tiene la capacidad suficiente para ejecutar todos los algoritmos de control. En las figuras (4.15) y (4.16) se muestran las medidas tomadas mediante un osciloscopio. La forma de medir el tiempo de procesamiento consiste en cambiar el estado

Controlador E1	
IEC	11.5540
IVABSC (PWM del motor derecho)	1340.9900
IVABSC (PWM del motor izquierdo)	1169.2700
IVAVSC (PWM del motor derecho)	8864.6000
IVAVSC (PWM del motor izquierdo)	7636.0000
Controlador E2	
IEC	55.2093
IVABSC (PWM del motor derecho)	1285.5700
IVABSC (PWM del motor izquierdo)	1285.5700
IVAVSC (PWM del motor derecho)	3182.2000
IVAVSC (PWM del motor izquierdo)	3182.2000

Tabla 4.2. Comparación de los resultados entre el controlador implementado de fábrica, E1, y el controlador por retroalimentación de estados E2.

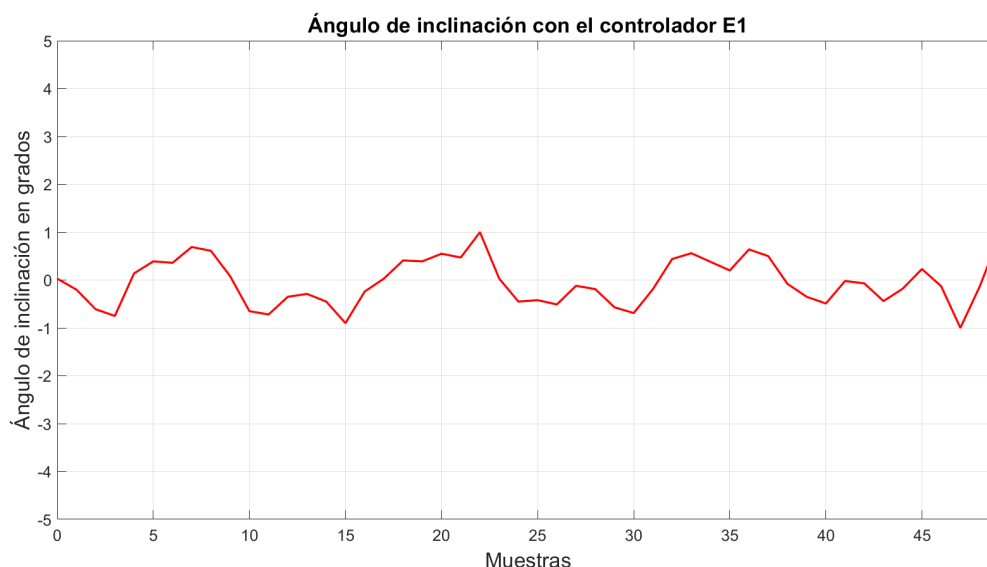


Figura 4.9. Medidas del ángulo de inclinación del robot con el controlador E1.

de un pin al inicio y final de la secuencia de interrupción del timer donde se ejecutan los algoritmos de control.

Se aprecia que los tiempos de procesamiento en ambos casos son muy similares, sin embargo, una diferencia importante es que el controlador E2 tiene un tiempo sobrante mayor debido a que el período de muestreo es mayor. Esto es una ventaja si se desea agregar más código para tareas secundarias. En la tabla (4.3), indicando que el controlador E2, tiene un mejor desempeño del tiempo de procesamiento ya que ocupa menos del 20 % del período de muestreo.



Figura 4.10. Medidas del ángulo de inclinación del robot con el controlador por E2.

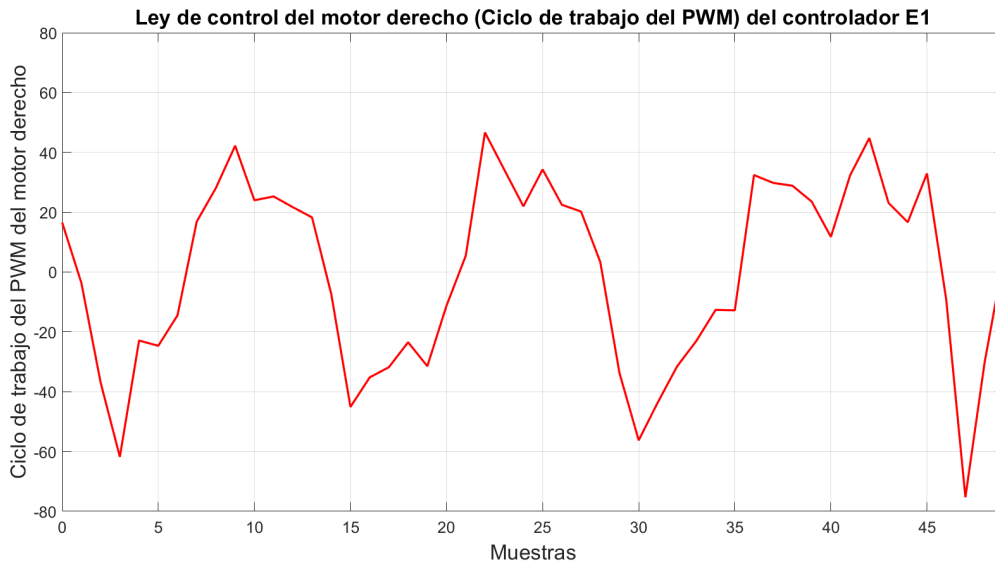


Figura 4.11. Medidas de la ley de control del motor derecho con el controlador E1.

4.5. Resultados experimentales para el seguimiento de trayectoria

Debido a que el robot necesita un controlador para mantenerse en equilibrio, es necesario combinar el control de equilibrio con el control para seguir una trayectoria. Como ya se ha mencionado, la estrategia a utilizar es conmutar entre los dos controladores. Al igual que en la simulación la trayectoria a seguir es un círculo de un radio de 0.75 m . Por lo tanto, habrá

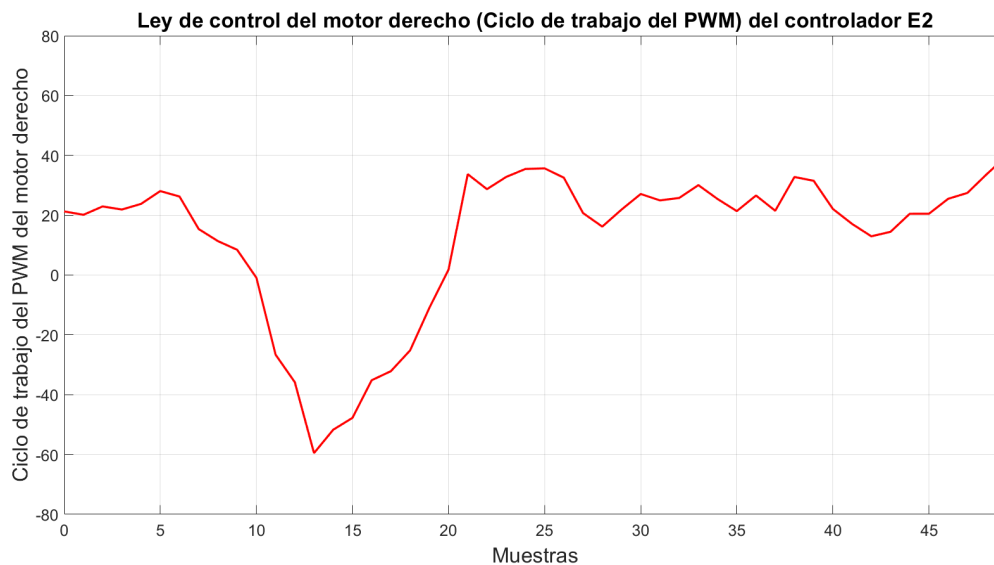


Figura 4.12. Medidas de la ley de control del motor derecho con el controlador E2.

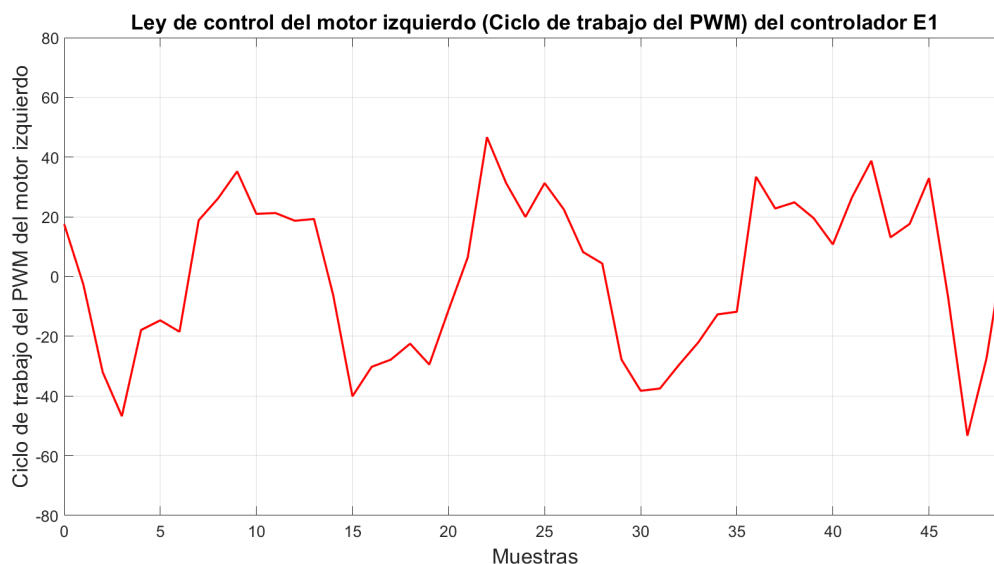


Figura 4.13. Medidas de la ley de control del motor izquierdo con el controlador E1.

dos casos a analizar y a comparar: en el controlador R1 se tiene un controlador PI-PD, con sintonización distinta al implementado de fábrica, con el controlador basado en técnicas de Lyapunov. El controlador R2 tiene el controlador por retroalimentación de estados y el controlador basado en técnicas de Lyapunov. En ambos casos se obtuvo resultados satisfactorios. En la figura (4.17) se muestra una imagen demostrativa del primer caso y en (4.18) se muestra la prueba realizada en el segundo caso.

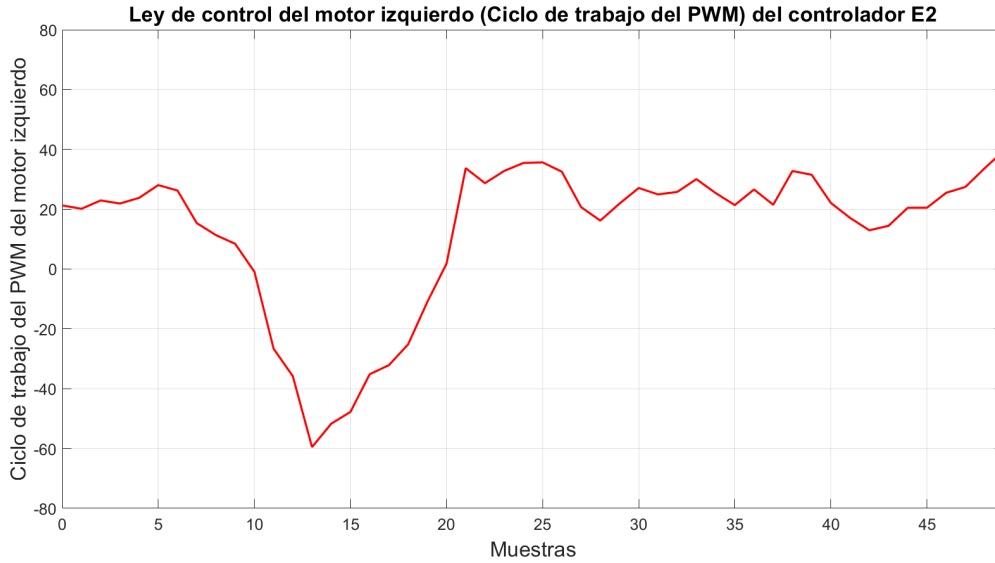


Figura 4.14. Medidas de la ley de control del motor izquierdo con el controlador E2.

Controladores	Tiempo de procesamiento medido	Período de muestreo	Porcentaje usado del período de muestreo
Controlador E1 (PI-PD)	2.5 ms	5 ms	50 %
Controlador E2 (Retroalimentación de estados)	2.6 ms	15 ms	17.33 %

Tabla 4.3. Comparación de los tiempos de procesamiento en los controladores E1 y E2 con una frecuencia del oscilador interno de 16 MHz.

4.6. Comparación de controladores para el equilibrio y seguimiento de trayectoria

Los dos controladores a comparar son los R1 y R2 mostrados en las figuras (4.19) y (4.20) respectivamente. Es necesario recalcar que, para el controlador R1, a pesar de usar un controlador PI-PD para el equilibrio, se emplea una sintonización distinta, así como un cambio en las interrupciones de los pulsos de los encoders.

Para realizar la comparación de los conjuntos de controladores se evaluarán de la misma manera que se usó en la tabla (4.2), pero se añadirán el error de seguimiento de trayectoria y el consumo de la batería. Para realizar las pruebas se busca que las condiciones en ambos casos sean los más parecidas posibles, esto se muestra en la tabla (4.4).

En la ecuación (4.4), se muestra la ecuación paramétrica de la trayectoria circular usada

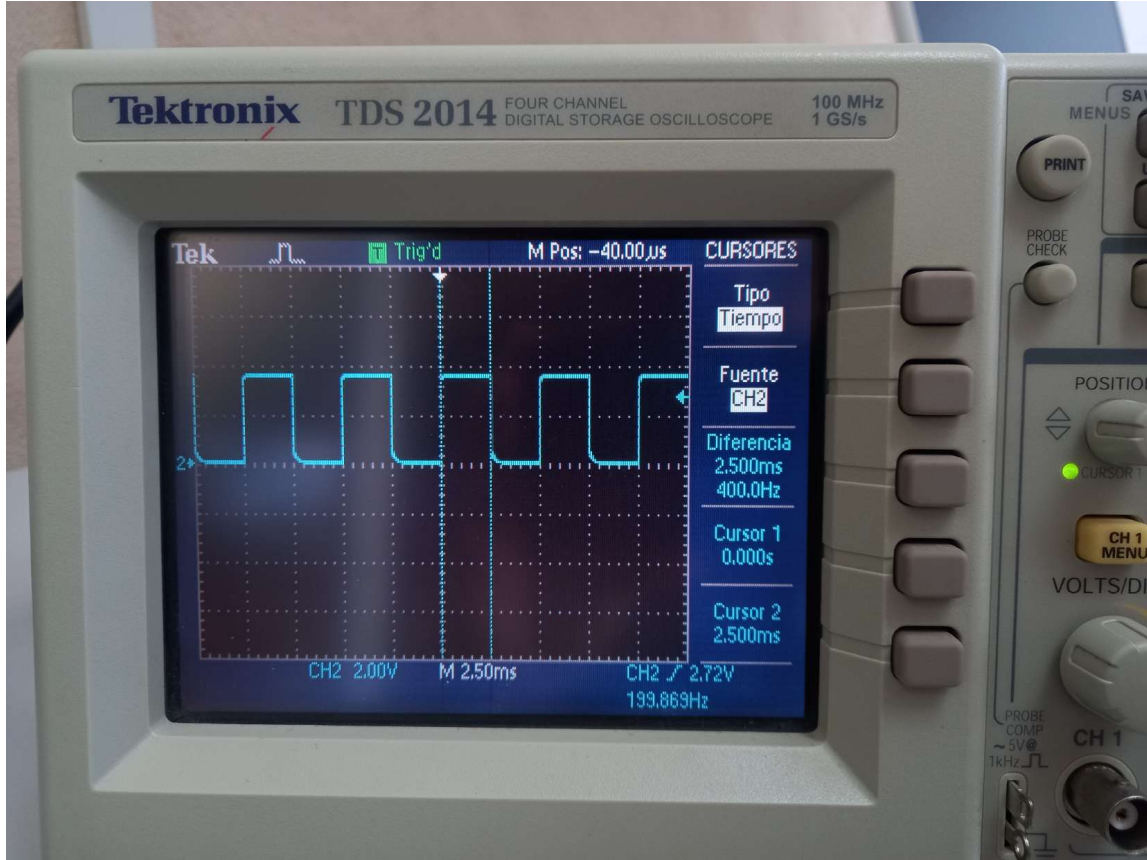


Figura 4.15. Tiempo de procesamiento del controlador E1.

Dato	Valor
Punto en adelanto, a_{rob}	0.20 m
Radio del círculo	0.75 m
Velocidad angular de la trayectoria	0.011 rad/seg

Tabla 4.4. Datos usados para las pruebas

de referencia.

$$\mathbf{h}_{d\text{rob}} = \begin{bmatrix} h_{dx\text{rob}} \\ h_{dy\text{rob}} \end{bmatrix} = \begin{bmatrix} 0.75 \cos(0.011 t_{\text{circ}}) \\ 0.75 \sin(0.011 t_{\text{circ}}) \end{bmatrix} \quad (4.4)$$

Y la derivada de la ecuación (4.4) corresponde a las velocidades de referencia que debe seguir el robot, definidas en la ecuación (4.5).

$$\dot{\mathbf{h}}_{d\text{rob}} = \begin{bmatrix} \dot{h}_{dx\text{rob}} \\ \dot{h}_{dy\text{rob}} \end{bmatrix} = \begin{bmatrix} (-0.75) (0.011) \sin(0.011 t_{\text{circ}}) \\ (0.75) (0.011) \cos(0.011 t_{\text{circ}}) \end{bmatrix} \quad (4.5)$$

En la figura (4.21) se muestra el ángulo de inclinación para el primer caso y en la figura (4.22) para el segundo caso.

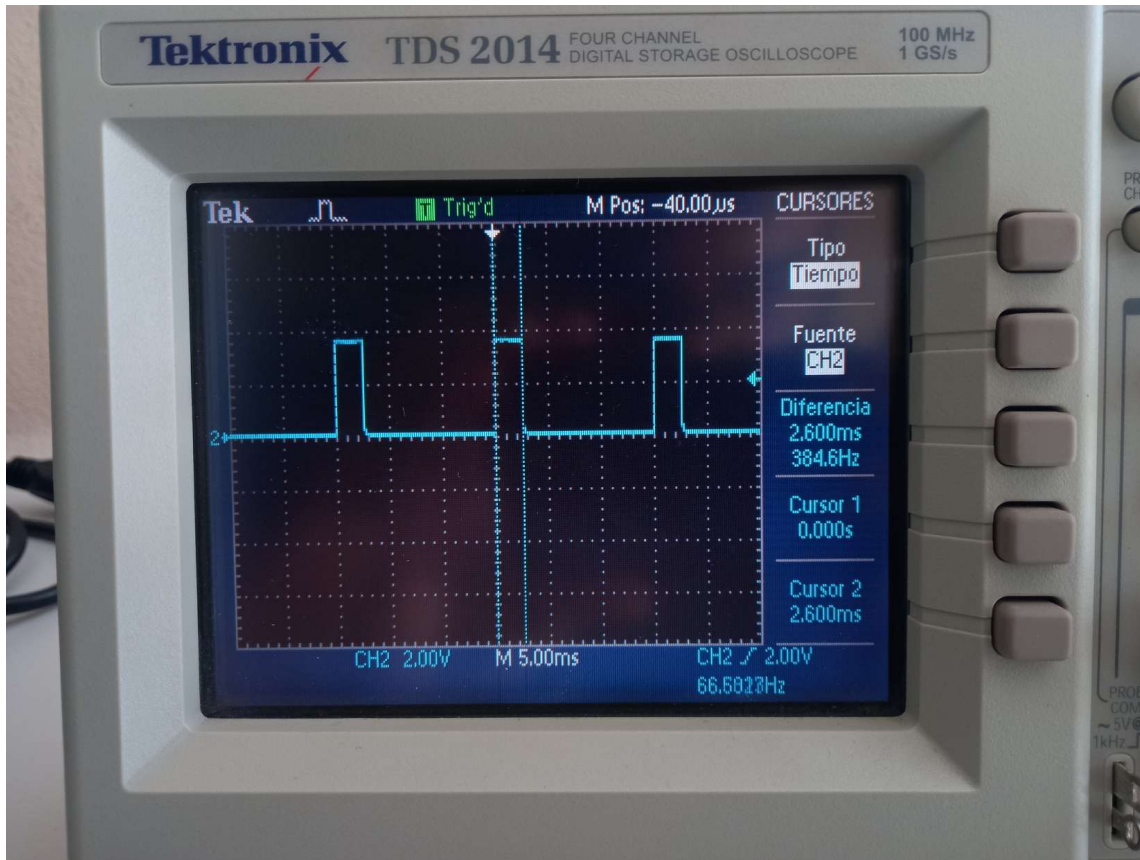


Figura 4.16. Tiempo de procesamiento del controlador E2.

En la figura (4.23) se observa la ley de control para el motor izquierdo y en (4.24).

En la figura (4.25) se muestra la ley de control para el motor izquierdo en el primer caso y en (4.26) para el segundo caso.

En la figura (4.27) se muestra el seguimiento de trayectoria para el primer caso y en (4.28) se muestra el seguimiento de trayectoria para el segundo caso.

Al igual que en la sección anterior, también se debe comparar los tiempos de procesamiento para cada caso en función del tiempo de muestreo. Estos tiempos de procesamiento se mide con un osciloscopio como se muestra en las figuras (4.29) y (4.30).

En la tabla (4.5) se muestra, de manera más explícita, la comparación de los tiempos de procesamiento. Se observa que el controlador R2 tiene mejor desempeño en este aspecto, ya que ocupa menos del 30 % de período de muestreo.

Se puede observar que ambos casos tienen tiempos de procesamiento similares, siendo un poco menor en el primer caso. Sin embargo, en el segundo caso hay mucho más tiempo disponible que se puede aprovechar para agregar más código para pruebas futuras. En el primer caso ya casi se llega al límite de tiempo disponible que se tiene.

Continuando con la comparación entre los dos casos se muestran índices de desempeño

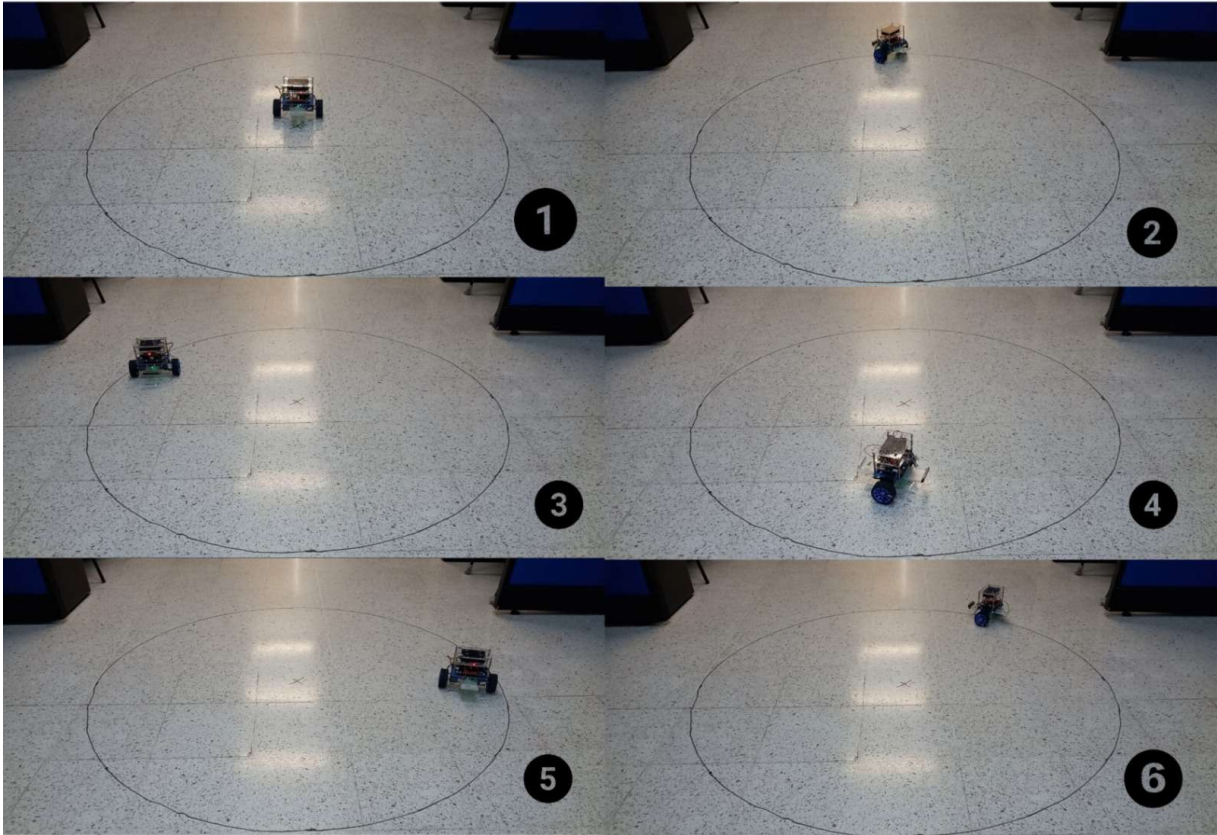


Figura 4.17. Prueba de seguimiento de trayectoria controlador R1.

en la tabla (4.6). Respecto al IEC del ángulo de inclinación entre ambos controladores existe una diferencia de 1293.9551 unidades, favoreciendo al caso uno. Respecto al IEC de las coordenadas X e Y, existe una diferencia de 18.8179 y 16.2083 unidades respectivamente, siendo el caso uno los que tiene los valores menores. Respecto al IVABSC del motor izquierdo y derecho, existe una diferencia de 1175.18 y 1004.18 unidades respectivamente. Siendo el caso uno el que tiene los valores menores. Analizando el IVAVSC del motor izquierdo y derecho, existen una diferencia de 464.5743 y 396.2959 unidades respectivamente. Igualmente, el caso uno tiene los valores más pequeños.

En general el primer caso tiene mejor desempeño, esto se debe a que el período de muestreo en el segundo caso es mayor y en combinación con la conmutación de controladores provoca que el control de seguimiento de trayectoria no actúe tan rápido como en el primer caso. Otra consecuencia de este comportamiento es que hay partes donde el ángulo de inclinación se incrementa significativamente, pero el robot vuelve a equilibrarse y continua con la trayectoria.

Otro aspecto por comparar es la duración de la batería mientras el robot está en movimiento. La prueba consiste en dejar el robot funcionando mientras se mantiene en

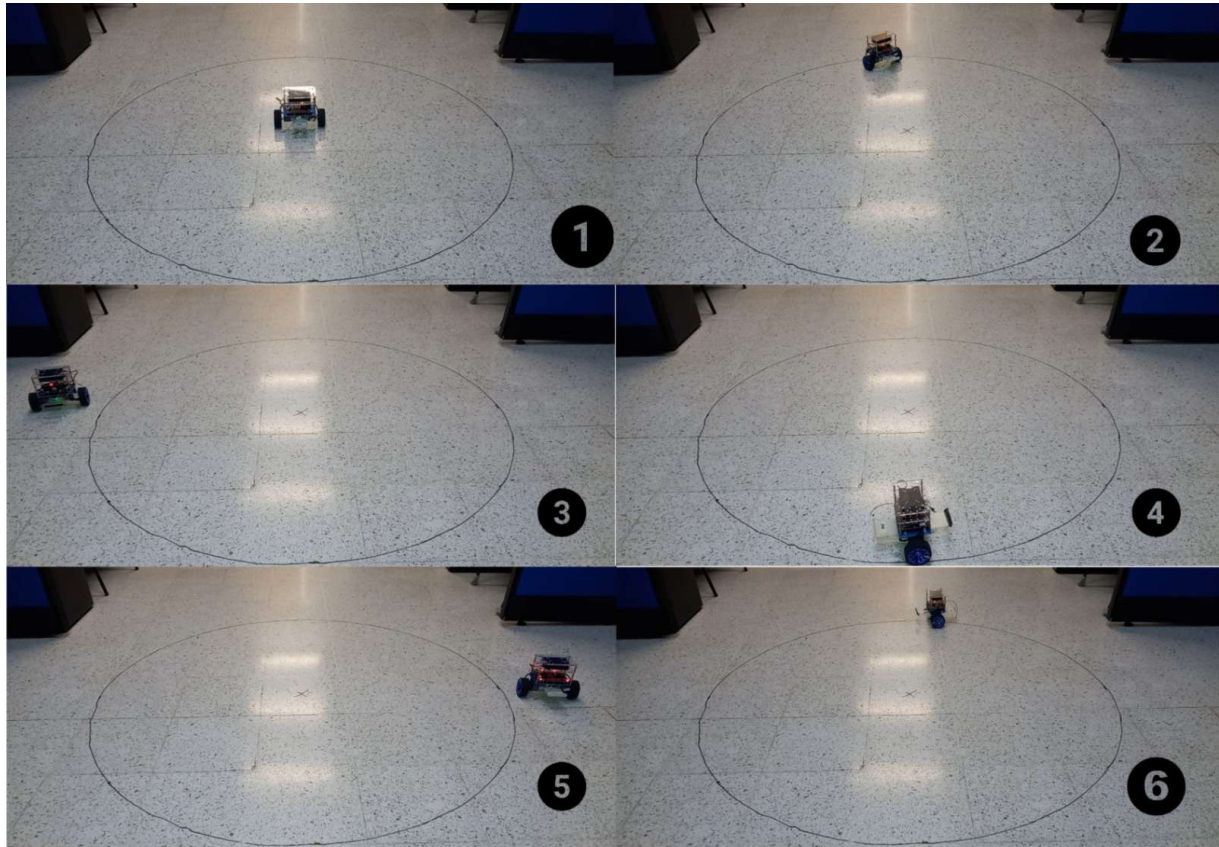


Figura 4.18. Prueba de seguimiento de trayectoria controlador R2.

equilibrio y sigue la trayectoria circular. De esta manera se tiene los dos controladores funcionando a la par. Asimismo, para realizar la prueba se utiliza en un solo robot y aparte en el mismo espacio de trabajo, de esta manera se obtiene una comparación lo más justa posible. La duración de la batería en ambos casos se muestra en la tabla (4.7).

Se observa que el controlador R1 fue más favorable en cuanto al consumo de energía de la batería, superando al controlador R2 por casi un 50 % respecto al tiempo máximo medido. Esto se debe a que en el controlador R2 el controlador basado en técnicas de Lyapunov tarda más en actuar y provoca más movimiento de los motores, mejorar este aspecto corresponde al trabajo futuro a realizar.

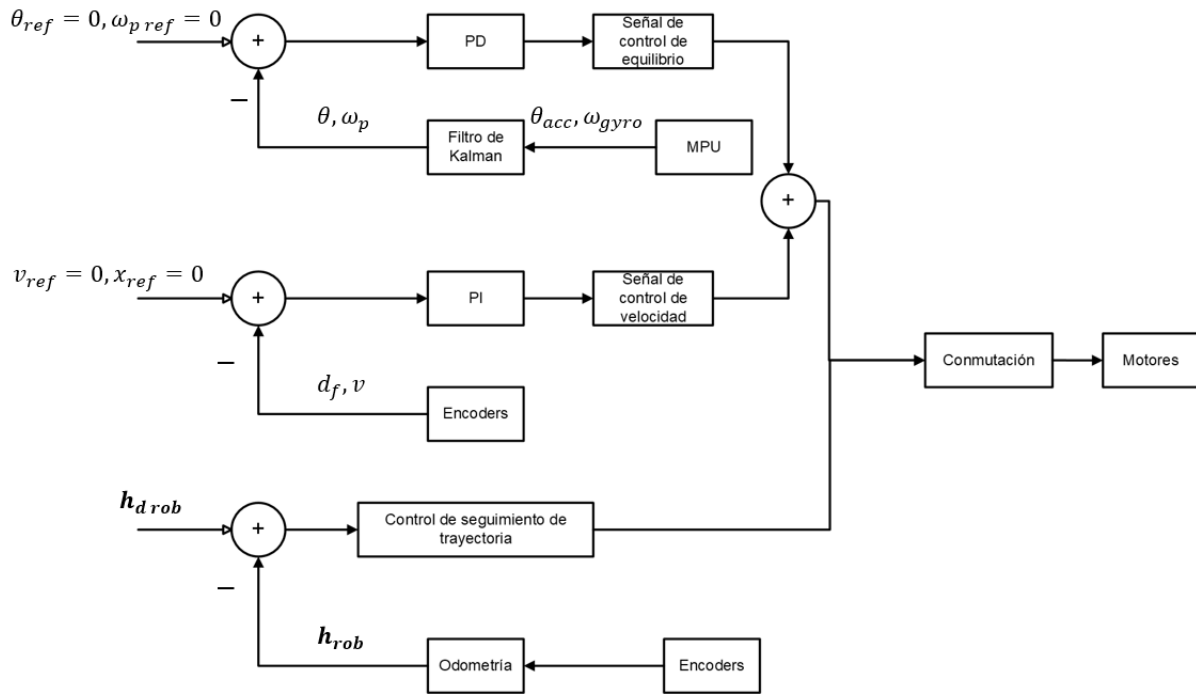


Figura 4.19. Controlador R1.

Controladores	Tiempo de procesamiento medido	Período de muestreo	Porcentaje usado del período de muestreo
Controlador R1 (PI-PD modificado y controlador basado en técnicas de Lyapunov)	3.6 ms	5 ms	72 %
Controlador R2 (Retroalimentación de estados y controlador basado en técnicas de Lyapunov)	4 ms	15 ms	26.66 %

Tabla 4.5. Comparación de los tiempos de procesamiento en los controladores R1 y R2 con una frecuencia del oscilador interno de 16 MHz.

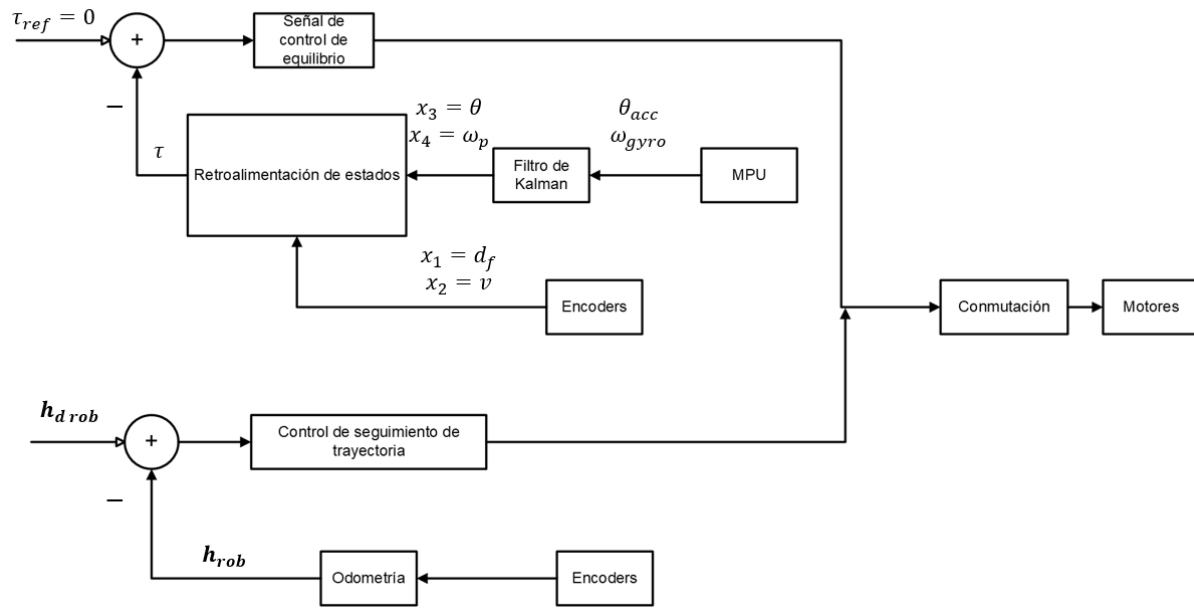


Figura 4.20. Controlador R2.

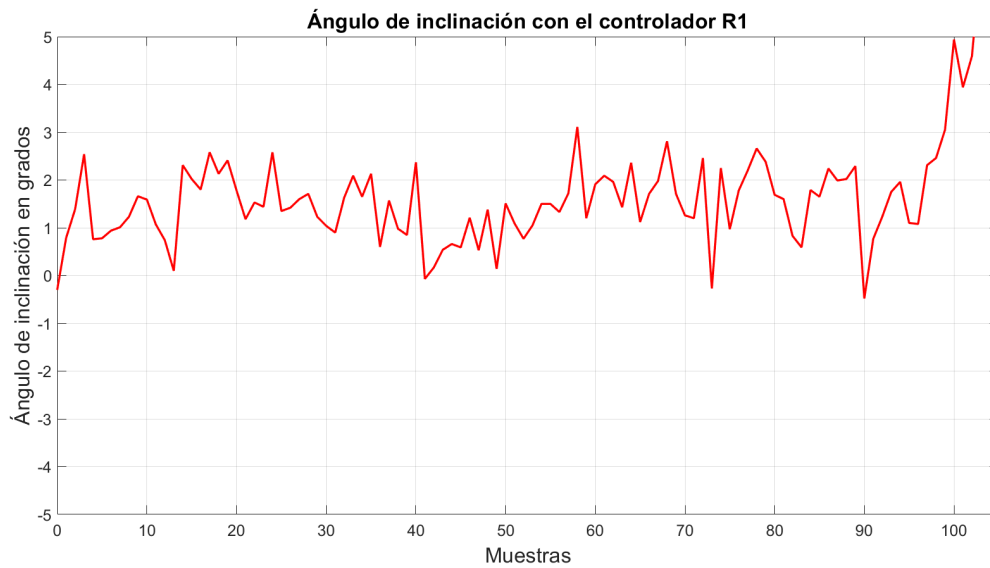


Figura 4.21. Ángulo de inclinación durante el seguimiento de trayectoria del controlador R1.

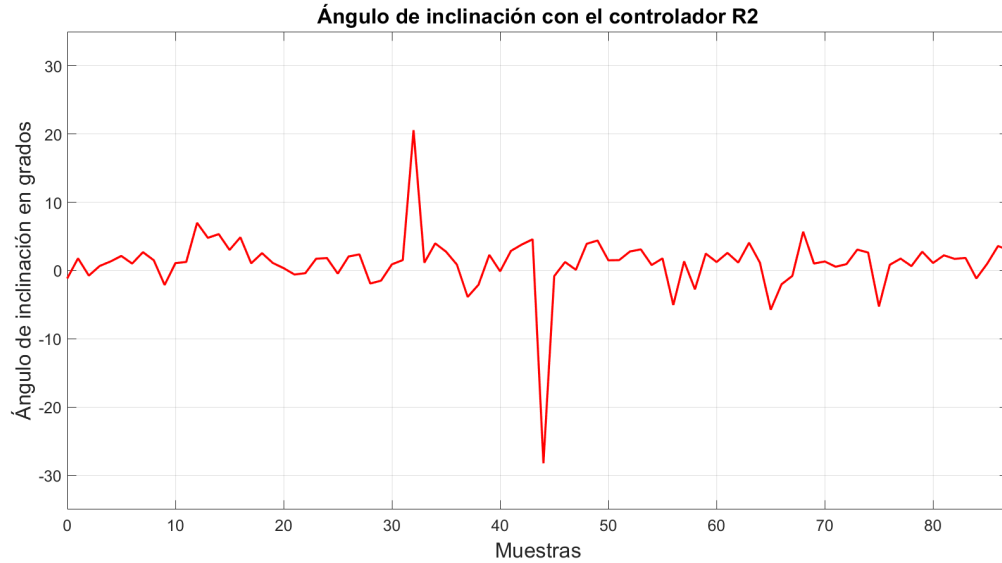


Figura 4.22. Ángulo de inclinación durante el seguimiento de trayectoria del controlador R2.

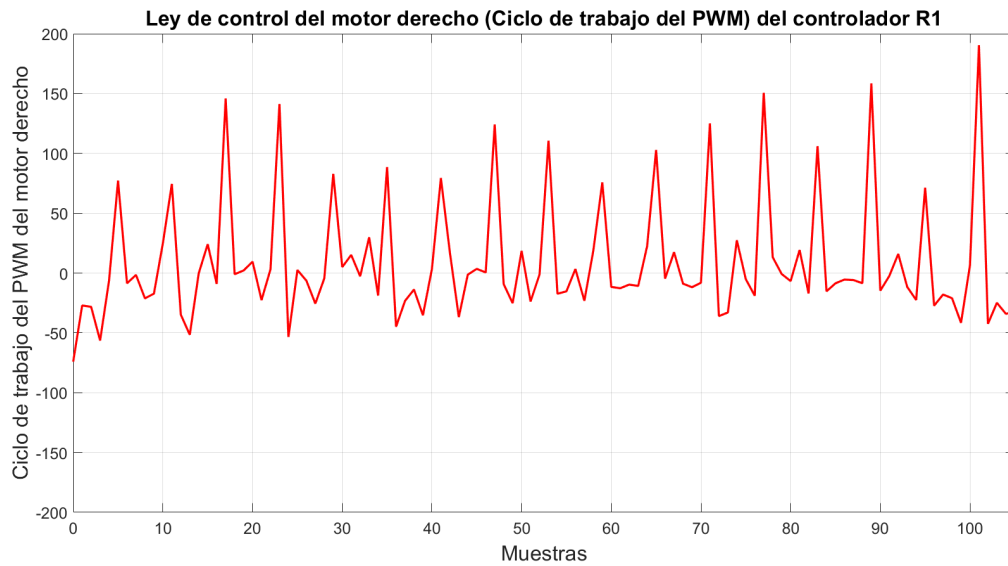


Figura 4.23. Ley de control del motor derecho durante el seguimiento de trayectoria del controlador R1.

	Controlador R1	Controlador R2
IEC del Ángulo de inclinación	515.7312	1809.6863
IEC de la coordenada en X	1.0049	19.8228
IEC de la coordenada en Y	0.4365	16.6448
IVABSC del motor izquierdo	3008.9600	4184.1400
IVABSC del motor derecho	3483.5900	4487.7700
IVAVSC del motor izquierdo	933.7460	1398.3203
IVAVSC del motor derecho	1123.5980	1519.8939

Tabla 4.6. Comparación de los controladores para el seguimiento de una trayectoria circular.

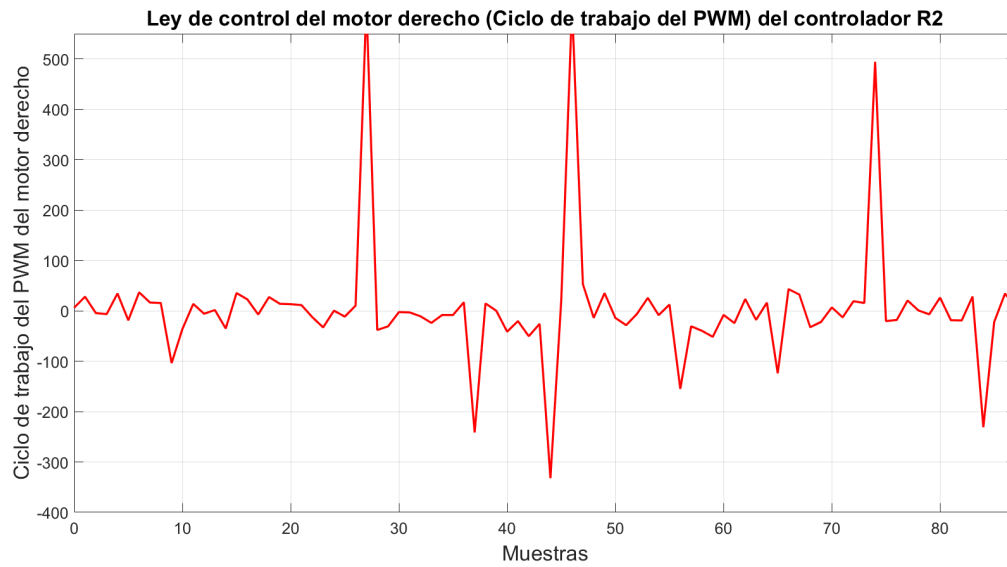


Figura 4.24. Ley de control del motor derecho durante el seguimiento de trayectoria del controlador R2.

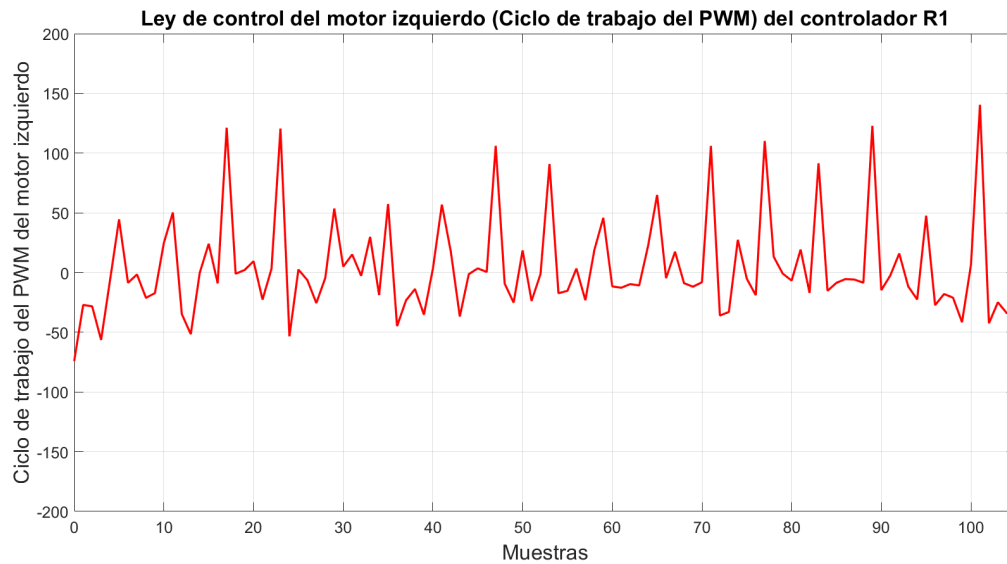


Figura 4.25. Ley de control del motor izquierdo durante el seguimiento de trayectoria del controlador R1

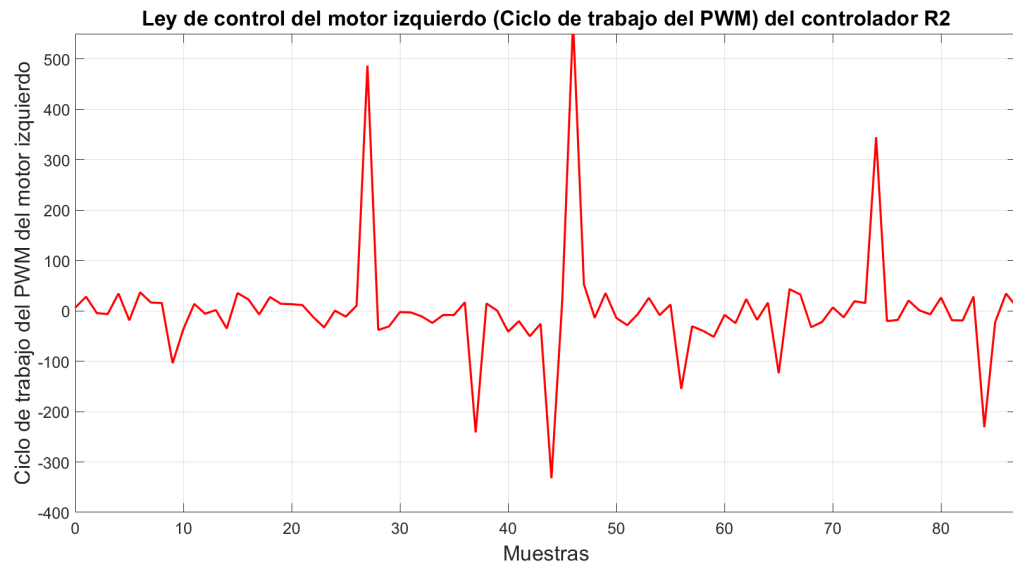


Figura 4.26. Ley de control del motor izquierdo durante el seguimiento de trayectoria del controlador R2.

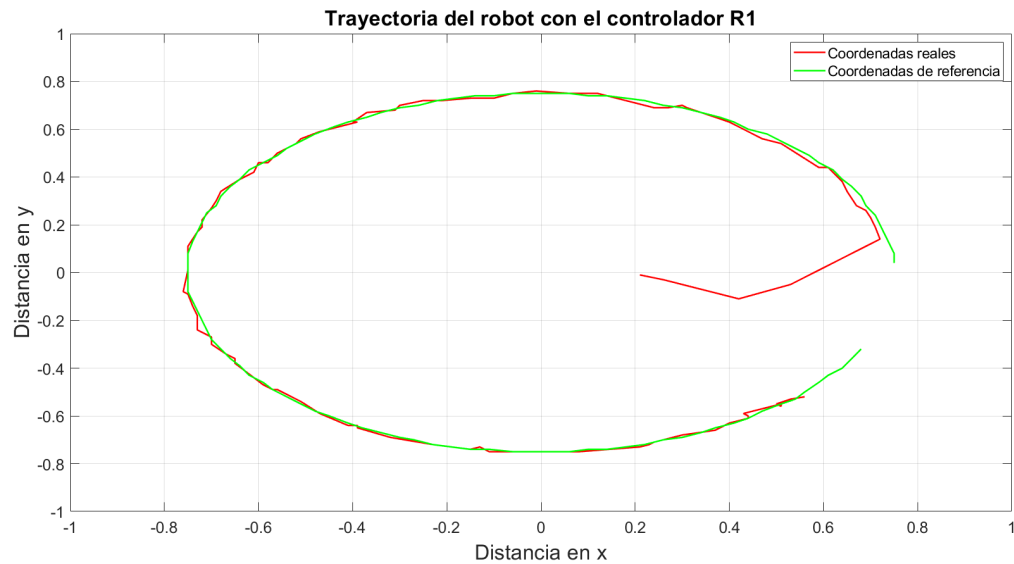


Figura 4.27. Seguimiento de trayectoria del controlador R1.

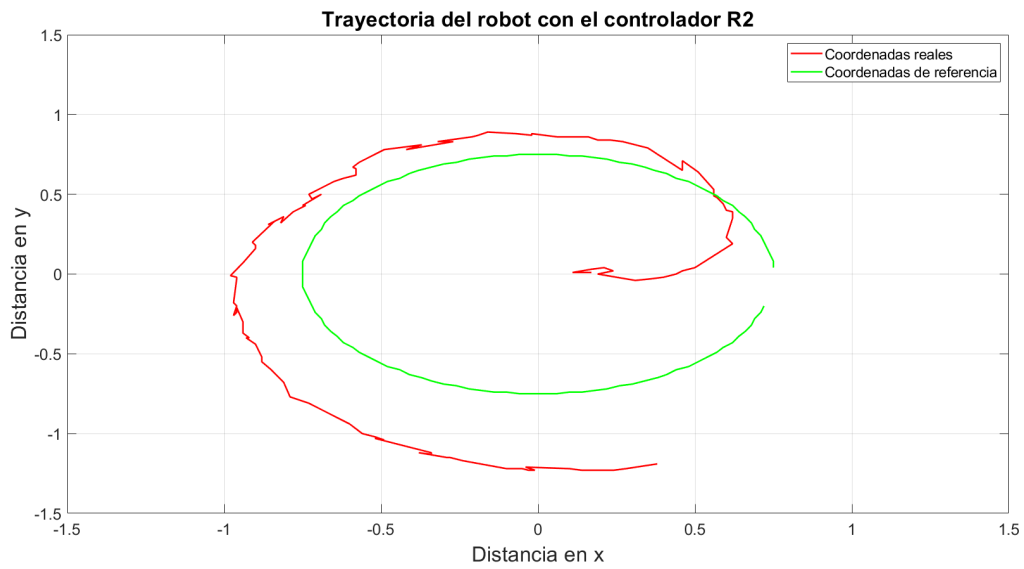


Figura 4.28. Seguimiento de trayectoria del controlador R2.

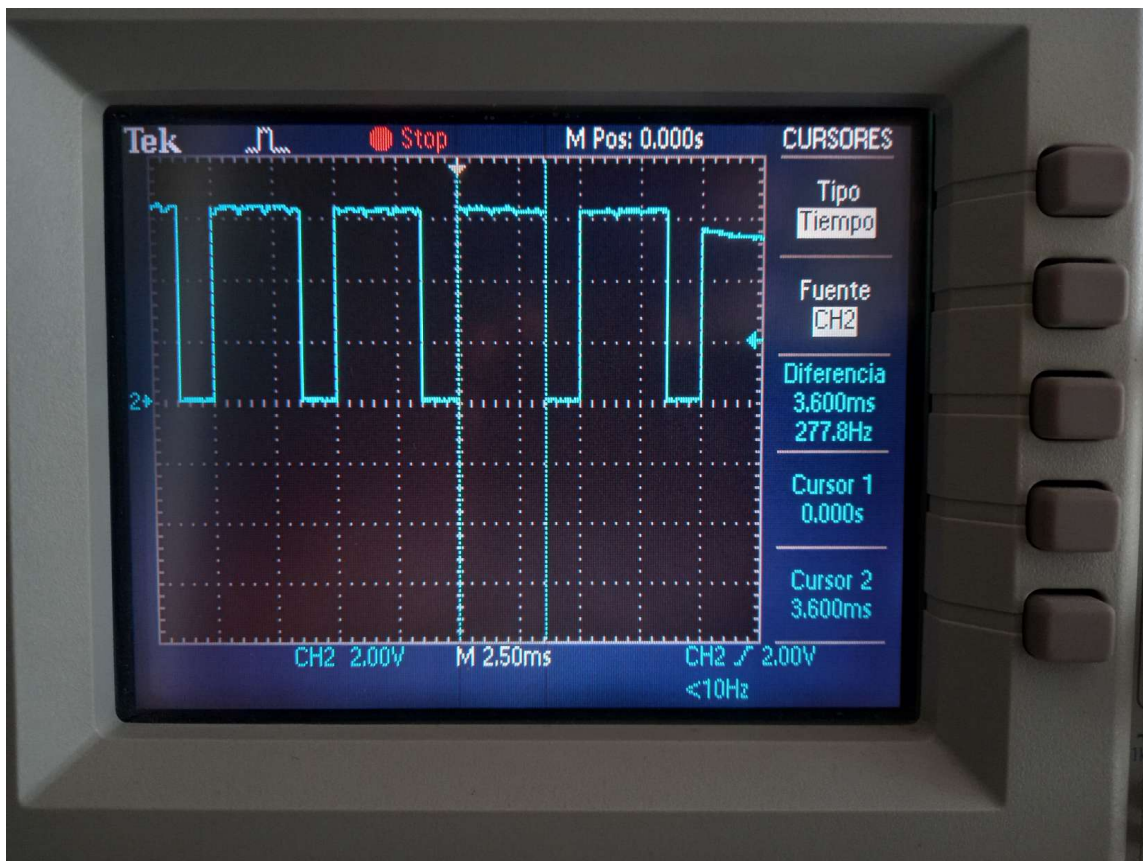


Figura 4.29. Tiempo de procesamiento del controlador R1.

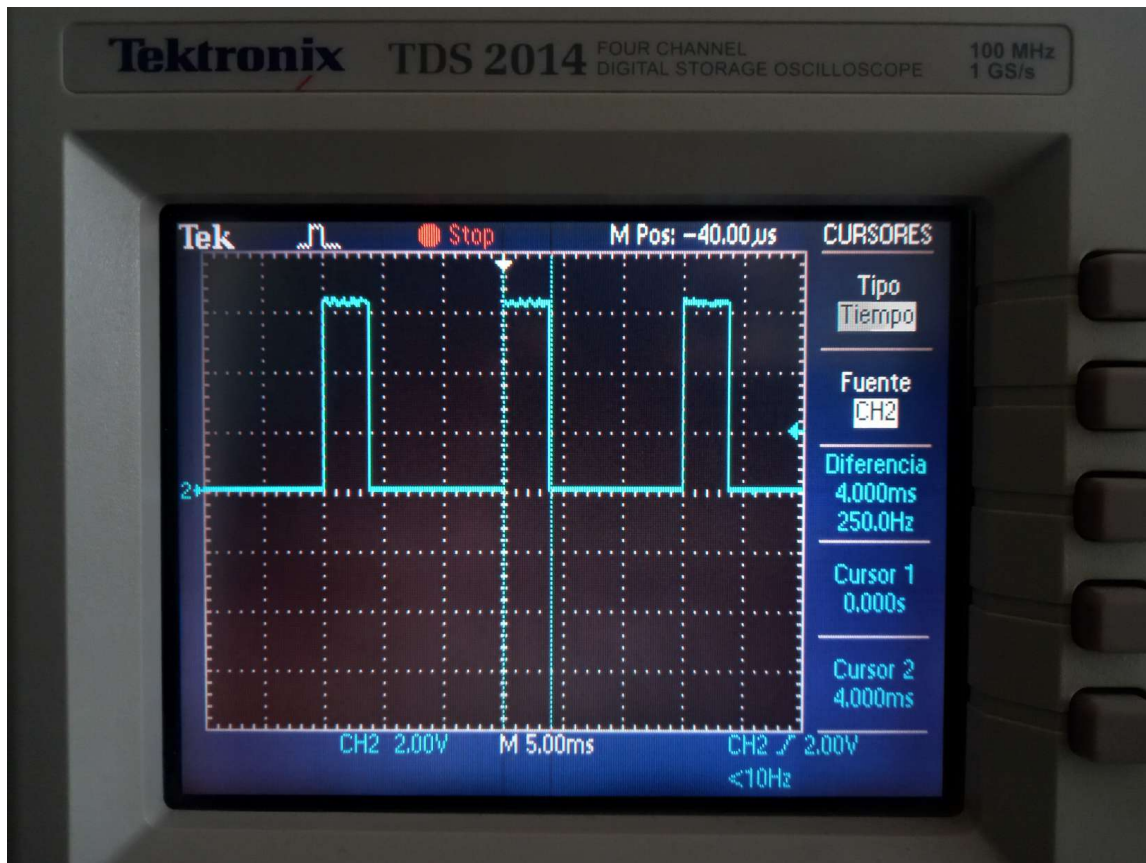


Figura 4.30. Tiempo de procesamiento del controlador R2.

Casos	Tiempo de duración de la batería	Tiempo máximo registrado	Porcentaje del tiempo máximo registrado
Controlador R1 (PI-PD y Lyapunov)	03:13:56 hrs	03:13:56 hrs	100 %
Controlador R2 (Retroalimentación de estados y Lyapunov)	01:47:29 hrs	03:13:56 hrs	46.96 %

Tabla 4.7. Duración de la batería para cada controlador.

CONCLUSIONES

Mediante la aplicación de técnicas modernas de control, fue posible diseñar un controlador por retroalimentación de estados, controlador E2, que mejoró el desempeño del robot en tareas de equilibrio con respecto al controlador originalmente implementado, controlador E1.

Asimismo, al integrar la odometría fue posible incorporar un controlador basado en técnicas de Lyapunov para el seguimiento de una trayectoria. Finalmente, utilizando una estrategia de conmutación fue posible integrar los dos algoritmos para satisfacer los objetivos de equilibrio y seguimiento de trayectoria. De esta manera se obtienen dos conjuntos de controladores: el controlador R1, PI-PD con el controlador basado en técnicas de Lyapunov, y el controlador R2, retroalimentación de estados con el controlador basado en técnicas de Lyapunov. Es necesario recalcar que el controlador R1 contiene un controlador PI-PD, pero con una sintonización distinta a la implementada de fábrica. Este cambio en la sintonización se debe a la implementación conjunta del controlador de seguimiento de trayectoria.

Los objetivos de la investigación fueron: obtener el modelo de un robot de autobalance de dos ruedas, desarrollar e implementar un controlador por retroalimentación de estados y otro basado en técnicas de Lyapunov para el seguimiento de trayectoria, y finalmente comparar el desempeño de ambos controladores en condiciones similares. Todos estos objetivos se cumplieron de manera satisfactoria. La formulación del modelo se basó en los enfoques de Euler-Lagrange y Trabajo Virtual. Para garantizar la precisión de los parámetros mecánicos, se calcularon los momentos de inercia y los centros de masa y se validaron mediante un modelo en SolidWorks. La exactitud del modelo quedó demostrada mediante simulaciones cuyo comportamiento coincidió con lo esperado. Contar con un modelo de este nivel de detalle facilitó significativamente el diseño del controlador por retroalimentación de estados.

El desempeño del sistema con los controladores propuestos se validó tanto mediante simulaciones como en pruebas experimentales. Las simulaciones se llevaron a cabo en tiempo continuo y discreto, asegurando su funcionamiento adecuado antes de implementarlos en el dispositivo digital del robot.

Para realizar las pruebas experimentales fue fundamental analizar el hardware y el

software del robot, lo cual, combinado con el modelo matemático, permitió implementar el controlador de seguimiento de trayectoria junto con el controlador de equilibrio. Gracias a ello, el robot logró mantenerse en equilibrio mientras seguía una trayectoria circular.

Dado que solo se consideró la odometría, se presentó error acumulativo durante el desplazamiento del robot. No obstante, para los propósitos de este trabajo la odometría resulta suficiente, ya que fue empleada únicamente para comparar el desempeño energético del robot en los dos casos analizados. Esta comparación mostró que el controlador R1 (PI-PD y el controlador basado en técnicas de Lyapunov) presenta un menor consumo de energía a diferencia del controlador R2 (retroalimentación de estados y el controlador basado en técnicas de Lyapunov). Además, el controlador R1 presenta un menor error al seguir la trayectoria circular.

En conclusión, el desarrollo e implementación de los controladores, así como su desempeño, fueron exitosos. No obstante, existen varias líneas de trabajo futuro que podrían mejorar los resultados obtenidos. En particular, se propone incorporar un observador que complemente la información de los sensores para mejorar el desempeño del controlador de equilibrio. Para eliminar el error acumulativo de la odometría, sería conveniente integrar un sensor de visión que permita ubicar al robot con mayor precisión en su entorno. Finalmente, dado que el modelo actual aborda por separado el equilibrio y la cinemática de un robot diferencial, se plantea como trabajo futuro la fusión de ambos comportamientos en un único modelo que permita comprender de manera más integral la dinámica del sistema y, con ello, desarrollar algoritmos de control más robustos y eficientes.

Además, el trabajo de investigación dio como resultado un artículo de congreso titulado "Modeling and Control of a Mobile Robot"[39] presentado en el XX Congreso Latinoamericano de Control Automático (CLCA 2025) que se llevó a cabo del 13 al 17 de octubre en la ciudad de Cancún, Quintana Roo.



CÁLCULO DE LOS PARÁMETROS MECÁNICOS DEL ROBOT

A.1. Cálculo de los parámetros mecánicos

En esta sección se muestra las actividades que se realizaron para poder obtener los parámetros mecánicos del robot a utilizar, los cuáles son los tensores de inercia y los centros de masa. Para este procedimiento se divide el robot a utilizar en cuerpos geométricos básicos, y a cada uno de estos se obtiene su tensor de inercia y centro de masa. En la figura (A.1) se observa que el robot que se utiliza está compuesto de piezas que se pueden aproximar a prismas y cilindros.

En las figuras (A.2) y (A.3) se muestran los cuerpos geométricos en los que se divide el robot.

En la ecuación (A.1) se muestra la forma general de los tensores de inercia.

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (A.1)$$

Para el prisma se obtiene el tensor de inercia mostrado en (A.2) y el del cilindro en (A.3).

$$I = \frac{m}{12} \begin{bmatrix} (b^2 + c^2) & 0 & 0 \\ 0 & (a^2 + c^2) & 0 \\ 0 & 0 & (a^2 + b^2) \end{bmatrix} \quad (A.2)$$

$$I = \begin{bmatrix} \frac{m}{12}(3R^2 + H^2) & 0 & 0 \\ 0 & \frac{m}{12}(3R^2 + H^2) & 0 \\ 0 & 0 & \frac{mR^2}{2} \end{bmatrix} \quad (A.3)$$



Figura A.1. Robot Elegoo Tumbler

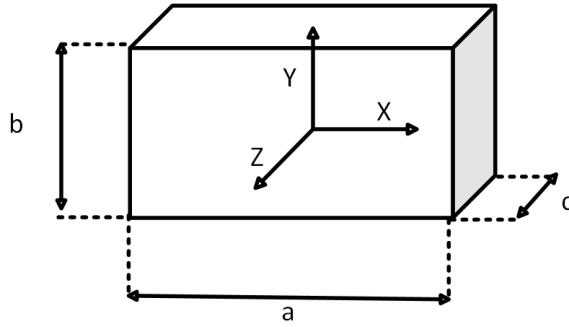


Figura A.2. Prisma rectangular.

Para las bases de los motores se utiliza la forma de medio cilindro, como se muestra en la figura (A.4).

El tensor de inercia para medio cilindro se muestra en la ecuación (A.4). Los términos de esta matriz se muestran en las ecuaciones (A.5), (A.6) y (A.7).

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (A.4)$$

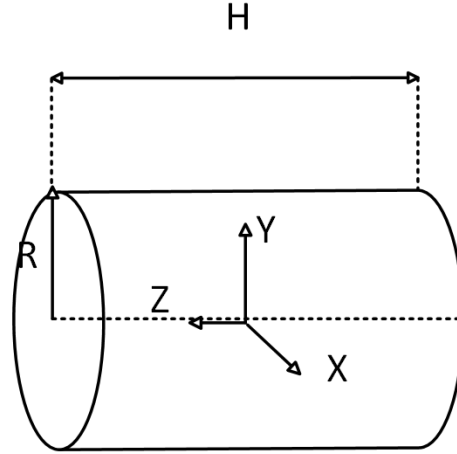


Figura A.3. Cilindro.

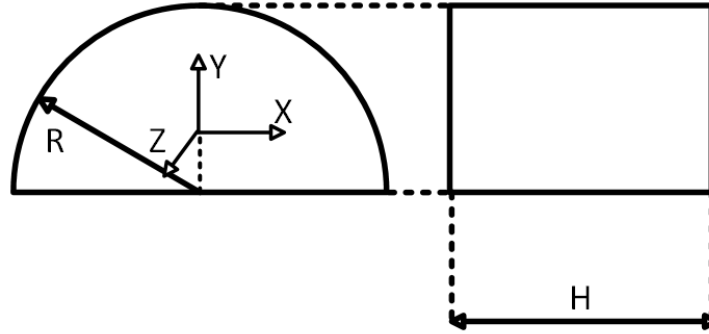


Figura A.4. Semicilindro.

$$I_{xx} = mR^2 \left(\frac{1}{2} + \frac{16}{9\pi^2} \right) + m \frac{H^2}{12} \quad (\text{A.5})$$

$$I_{yy} = m \left(\frac{R^2}{4} + \frac{H^2}{12} \right) \quad (\text{A.6})$$

$$I_{zz} = mR^2 \left(\frac{1}{2} + \frac{16}{9\pi^2} \right) \quad (\text{A.7})$$

Para poder encontrar el tensor de inercia de las formas anteriores respecto al marco de referencia del robot, es necesario utilizar el teorema de ejes paralelos. Para entender este teorema se utiliza la figura (A.5).

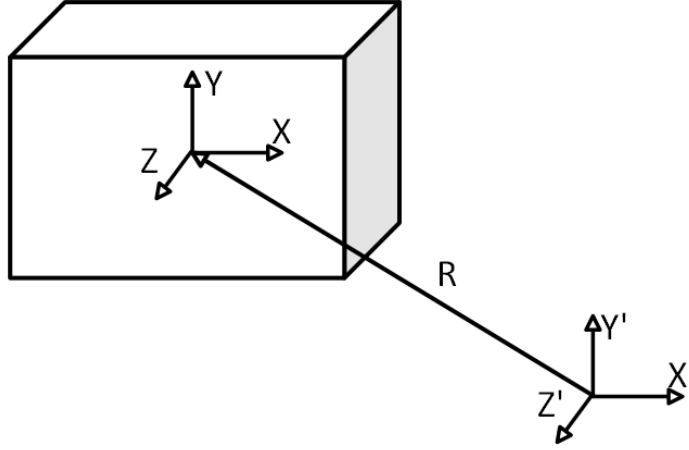


Figura A.5. Teorema de ejes paralelos.

Para obtener el tensor de inercia respecto al nuevo marco de referencia X', Y', Z' se utiliza la ecuación (A.8).

$$I' = I + m(\|R\|^2 \delta_{ij} - R_i R_j) \quad (\text{A.8})$$

donde:

- $R = [R_i, R_j, R_k]$ Es el vector que une los marcos de referencia.
- δ_{ij} es la Delta de Kronecker.

La Delta de Kronecker se muestra en la ecuación (A.9),

$$\delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \quad (\text{A.9})$$

La ecuación (A.8) al desarrollarse se desglosa en las ecuaciones (A.10) a (A.15).

$$I'_{xx} = I_{xx} + m(\|R\|^2 - R_x R_x) \quad (\text{A.10})$$

$$I'_{yy} = I_{yy} + m(\|R\|^2 - R_y R_y) \quad (\text{A.11})$$

$$I'_{zz} = I_{zz} + m(\|R\|^2 - R_z R_z) \quad (\text{A.12})$$

$$I'_{xy} = I'_{yx} = I_{xy} - m(R_x R_y) \quad (\text{A.13})$$

$$I'_{xz} = I'_{zx} = I_{xz} - m(R_x R_z) \quad (\text{A.14})$$

$$I'_{yz} = I'_{zy} = I_{yz} - m(R_y R_z) \quad (\text{A.15})$$

Para obtener las masas de las piezas se pesan cada una, un ejemplo de este procedimiento se muestra en la figura (A.6). Y mediante medidas se obtienen los vectores R , estos se muestran en las ecuaciones (A.16) y (A.17).



Figura A.6. Masa del motor.

$$R_{motor1} = [0, 0.0075m, 0.0345m] \quad (\text{A.16})$$

$$R_{motor2} = [0, 0.0075m, -0.0345m] \quad (\text{A.17})$$

En las ecuaciones (A.18) y (A.19) se muestran los resultados al aplicar el teorema de ejes paralelos al tensor de inercia del motor.

$$I_{motor1} = \begin{bmatrix} 2.274 \times 10^4 & 0 & 0 \\ 0 & 2.189 \times 10^{-4} & -3.881 \times 10^{-5} \\ 0 & -3.881 \times 10^{-5} & 3.41 \times 10^{-5} \end{bmatrix} \quad (A.18)$$

$$I_{motor2} = \begin{bmatrix} 2.274 \times 10^4 & 0 & 0 \\ 0 & 2.189 \times 10^{-4} & 3.881 \times 10^{-5} \\ 0 & 3.881 \times 10^{-5} & 3.41 \times 10^{-5} \end{bmatrix} \quad (A.19)$$

En la figura (A.7) se comprueban los resultados usando SolidWorks.

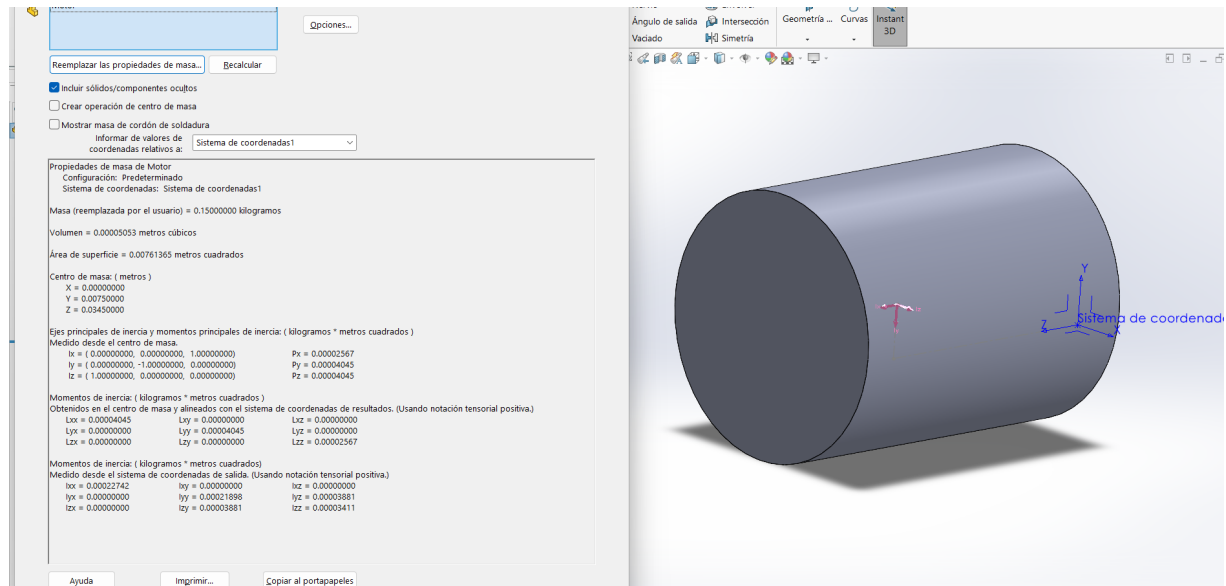


Figura A.7. Tensor de inercia del motor en SolidWorks.

Para calcular el centro de masa del conjunto de todas las figuras se usa la ecuación (A.20).

$$c.m. = \frac{\sum_i^n m_i(R)_i}{\sum_i^n m_i} \quad (A.20)$$

donde

- $c.m.$ es el centro de masa de todo el conjunto de figuras.
- m_i es la masa de cada una de las figuras.
- $(R)_i$ es el centro de masa de cada figura respecto a un marco de referencia global.

Se proceden a calcular los tensores de inercia de todo el robot y su respectivo centro de masa. Estos se muestran en las ecuaciones (A.21) y (A.22).

$$I_{robot} = \sum_{i=1}^n I_i = \begin{bmatrix} 308.575 \times 10^{-5} & 0 & 0 \\ 0 & 109.536 \times 10^{-5} & 0 \\ 0 & 0 & 240.307 \times 10^{-5} \end{bmatrix} \quad (A.21)$$

$$c.m = \begin{bmatrix} 0 & 0.0413m & 0 \end{bmatrix} \quad (A.22)$$

Los resultados anteriores se comprueban mediante SolidWorks, esto se muestra en la figura (A.8).

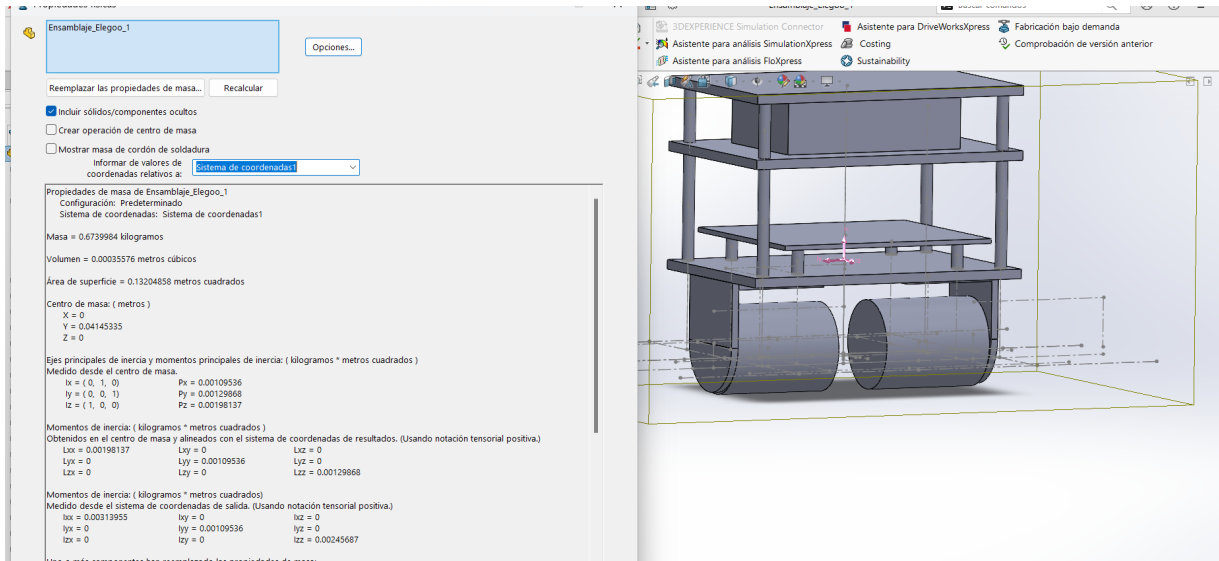


Figura A.8. Parámetros mecánicos del Robot obtenidos mediante SolidWorks.

Se realiza el mismo procedimiento para el conjunto de las dos ruedas. Estos resultados se muestran en las ecuaciones (A.23) y (A.24).

$$I_{ruedas} = I_{rueda1} + I_{rueda2} = \begin{bmatrix} 4.578 \times 10^{-4} & 0 & 0 \\ 0 & 4.578 \times 10^{-4} & 0 \\ 0 & 0 & 3.375 \times 10^{-5} \end{bmatrix} \quad (A.23)$$

$$c.m = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \quad (A.24)$$

Los resultados anteriores se comprueban mediante SolidWorks, esto se muestra en la figura (A.9).

En la tabla (A.1) se muestra la comparación de los valores calculados respecto a los obtenidos mediante SolidWorks. Se puede observar que los valores son prácticamente iguales.

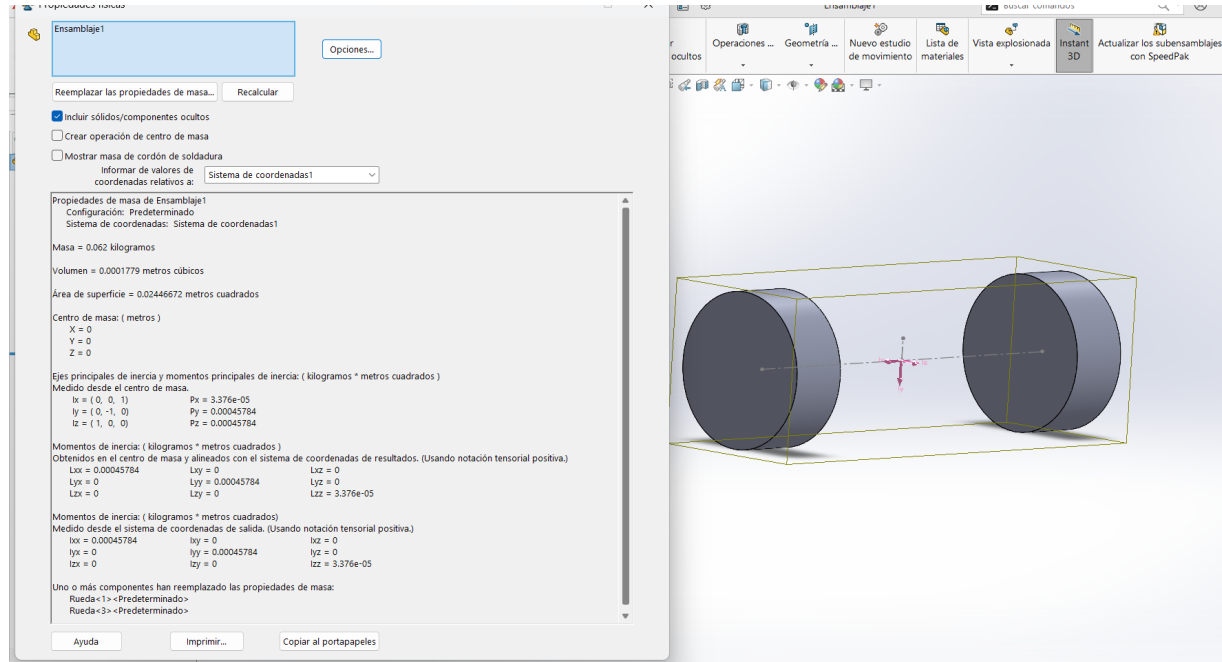


Figura A.9. Parámetros mecánicos de las ruedas obtenidos mediante SolidWorks.

Parámetro	Calculado	SolidWorks
Momento de inercia del robot (I_p)	$240.307 \times 10^{-5} kg m^2$	$245.687 \times 10^{-5} kg m^2$
Momento de inercia de las ruedas (I_R)	$3.3759 \times 10^{-5} kg m^2$	$3.376 \times 10^{-5} kg m^2$
Centro de masa del robot	$0.0413 m$	$0.04145335 m$
Centro de masa de las ruedas	$0 m$	$0 m$

Tabla A.1. Valores calculados y simulados de los parámetros mecánicos.

Finalmente se obtienen los parámetros del robot Elegoo. Estos se muestran en la ecuación (A.25).

$$\begin{aligned}l &= 0.0413m \\m_h &= 0.674kg \\I_p &= 240.307 \times 10^{-5} kg\,m^2 \\m_r &= 0.062Kg \\I_r &= 3.3759 \times 10^{-5} kg\,m^2 \\R &= 0.033m \\c_\omega &= 2.6 \times 10^{-3} \frac{Nms}{rad} \\c_v &= 2 \frac{Ns}{m}\end{aligned}\tag{A.25}$$

SISTEMAS NO HOLONÓMICOS

B.1. Sistemas no holonómicos

2.1.1. Modelado del sistema no holonómico

Los sistemas mecánicos pueden clasificarse en holonómicos y no holonómicos de acuerdo con la naturaleza de las ligaduras que restringen su movimiento. Una ligadura es una restricción impuesta al sistema que limita las configuraciones o las velocidades que este puede adoptar. En el caso de un mecanismo, una ligadura determina y limita el tipo de movimiento que pueden realizar sus componentes.

Un sistema no holonómico es aquel que posee al menos una ligadura cinemática no integrable, es decir, una restricción que depende de las velocidades generalizadas y que no puede expresarse únicamente como una relación algebraica entre las coordenadas generalizadas y el tiempo. Este tipo de ligaduras no reduce el espacio de configuración del sistema, sino que restringe las direcciones instantáneas de movimiento permitidas. Sistemas de este tipo aparecen comúnmente en robótica móvil, como en vehículos con ruedas que ruedan sin deslizamiento lateral [40].

2.1.1.1. Sistemas restringidos

Un sistema restringido es un sistema donde sus partículas no se pueden mover arbitrariamente, sino que hay una limitación que hace que se pueda mover sólo en una región determinada.

2.1.1.2. Ligaduras

Una ligadura es una expresión matemática que restringe el movimiento de un sistema. En la figura (B.1) se muestra a una partícula que solo se puede desplazar a lo largo de la

barra, esta limitación física es una ligadura que impide a la partícula el movimiento en el eje Y y Z . La ecuación que determina la fuerza de ligadura se muestra en (B.1).

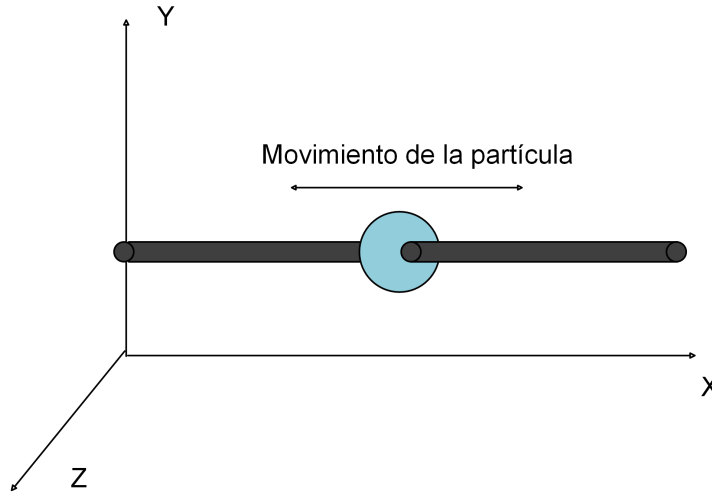


Figura B.1. Partícula insertada en una barra cilíndrica.

$$Y = Z = 0 \quad (B.1)$$

Algunas ligaduras pueden deberse al método de activación, por ejemplo, en la figura (B.2) se observa que un péndulo dependiendo de que cómo se active su movimiento puede tener un movimiento de semicírculo o cónico. De la misma forma sus fuerzas de ligaduras cambiarán.

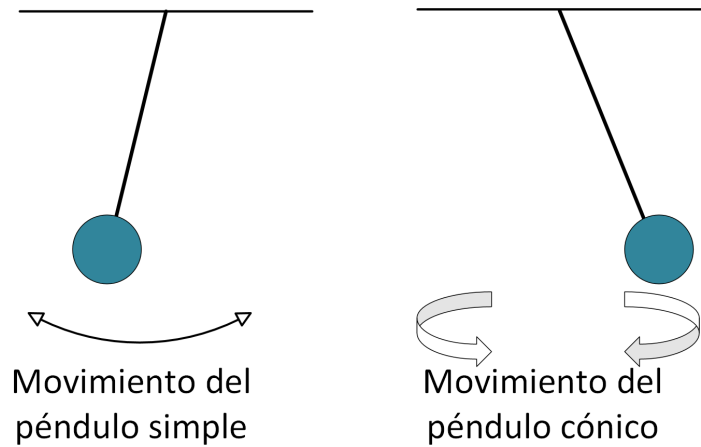


Figura B.2. Ligaduras por el método de activación.

2.1.1.3. Grados de libertad

Una vez que se define el concepto de ligadura se debe comprender lo qué es un grado de libertad. Para esto se tiene en cuenta que un mecanismo con N partículas se encuentra en un espacio tridimensional. Si estas partículas no tuvieran ningún tipo de ligadura sus movimientos serían arbitrarios y no sería un sistema restringido. Sin embargo, si las partículas se encuentran restringidas mediante c ecuaciones de ligaduras los grados de libertad se muestran en la ecuación (B.2).

$$n = 3N - c \quad (\text{B.2})$$

donde n es el número de grados de libertad.

De esta manera se puede definir a los grados de libertad como el número mínimo de las coordenadas independientes para poder describir el sistema de manera unívoca. Estas coordenadas se definen como coordenadas generalizadas.

2.1.1.4. Clasificación de los sistemas

De acuerdo con el tipo de ligaduras presentes en un sistema se puede realizar las siguientes clasificaciones:

- Sistemas holonómicos.
 - Sistemas holonómicos esclerónomos.
 - Sistemas holonómicos reónomos.
- Sistemas no holonómicos.

2.1.1.5. Sistemas holonómicos

Este tipo de sistemas tienen la característica de que todas sus ligaduras son del tipo geométricas y cinemáticas integrables. Este tipo de ligaduras se definen de la siguiente manera:

- Ligaduras geométricas: Son las ligaduras que dependen de la posición y tiempo de una partícula. Tienen la forma matemática de $f = f(r_i; t)$ donde:
 - r_i es la posición de la partícula respecto al marco de referencia.
 - t es el tiempo.

- Ligaduras cinemáticas integrables: Estas ligaduras son obtenidas por la derivación de una ligadura geométrica. Por lo tanto, las ligaduras geométricas imponen restricciones sobre las posiciones y también sobre las velocidades por medio de la ligadura cinemática integrable asociada. Tienen la forma matemática de $f = f(r_i, \dot{r}_i; t)$ donde:

- r_i es la posición de la partícula.
- $\dot{r}_i$ es la velocidad de la partícula.
- t es el tiempo del sistema.

2.1.1.6. Sistemas holonómicos esclerónomos

Son sistemas holonómicos que todas sus ligaduras son del tipo estacionarias.

- Ligaduras estacionarias: Son ligaduras que sólo dependen de las posiciones de las partículas. Y tienen la forma de $f = f(r_i)$.

2.1.1.7. Sistemas holonómicos reónomos

Son sistemas holonómicos donde alguna ligadura depende explícitamente del tiempo.

2.1.1.8. Sistemas no holonómicos

Es cuando alguna de sus ligaduras es cinemática no integrable.

2.1.1.9. Robots holonómicos

El manipulador plano mostrado en la figura (B.3) corresponde a un robot serial holonómico, ya que su movimiento está completamente determinado por las coordenadas articulares, sin la presencia de restricciones cinemáticas no integrables. Cada articulación impone una ligadura geométrica integrable, por lo que el sistema puede describirse mediante un conjunto de coordenadas generalizadas independientes.

La posición cartesiana del efector final puede expresarse mediante la cinemática directa como

$$x_e = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) + L_4 \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4) \quad (\text{B.3})$$

$$y_e = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) + L_4 \sin(\theta_1 + \theta_2 + \theta_3 + \theta_4) \quad (\text{B.4})$$

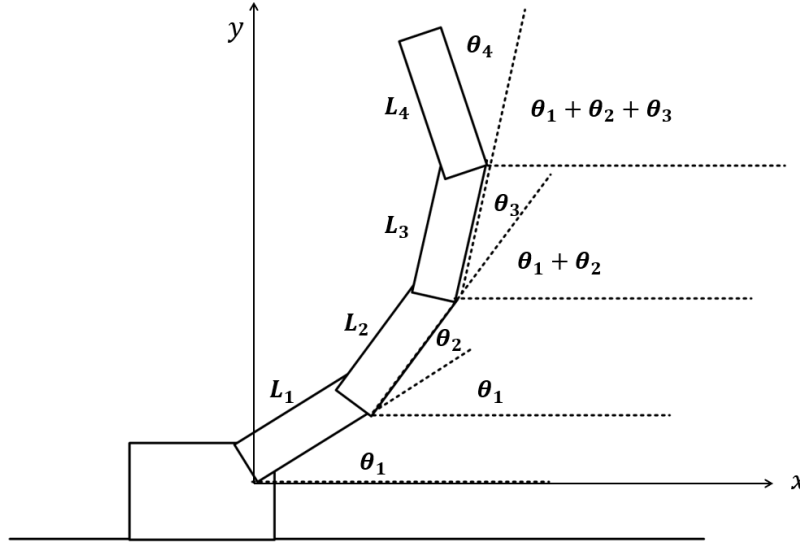


Figura B.3. Robot holonómico

Las ecuaciones anteriores corresponden a la cinemática directa del manipulador y no representan ligaduras activas sobre el sistema. El robot mostrado en la figura (B.3) es un sistema holonómico, ya que su configuración puede describirse completamente mediante un conjunto de coordenadas articulares independientes, sin la presencia de restricciones cinemáticas no integrables.

En este caso, el número de grados de libertad coincide con el número de articulaciones, por lo que el vector de coordenadas generalizadas puede definirse como

$$q = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4]^T \in \mathbb{R}^4 \quad (\text{B.5})$$

donde q describe de manera unívoca la configuración del robot. Al no existir ligaduras adicionales del tipo $g(q) = 0$, el espacio de configuración del sistema coincide con $\mathbb{R}^4$.

2.1.1.10. Restricciones de Pfaffian

Las ligaduras holonómicas pueden expresarse mediante relaciones algebraicas de la forma

$$g(q, t) = 0. \quad (\text{B.6})$$

Al derivar esta expresión respecto al tiempo, se obtiene una restricción cinemática en las velocidades. En el caso de ligaduras holonómicas estacionarias, dicha restricción puede escribirse en la forma de Pfaffian como

$$A(q)\dot{q} = 0, \quad (\text{B.7})$$

donde la matriz $A(q)$ está dada por

$$A(q) = \frac{\partial g}{\partial q} \in \mathbb{R}^{k \times n}, \quad (\text{B.8})$$

y puede expresarse de manera explícita como

$$A(q) = \begin{bmatrix} \frac{\partial g_1}{\partial q_1}(q) & \cdots & \frac{\partial g_1}{\partial q_n}(q) \\ \vdots & \ddots & \vdots \\ \frac{\partial g_k}{\partial q_1}(q) & \cdots & \frac{\partial g_k}{\partial q_n}(q) \end{bmatrix}. \quad (\text{B.9})$$

El vector de velocidades generalizadas está definido como

$$\dot{q} = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \cdots & \dot{q}_n \end{bmatrix}^T. \quad (\text{B.10})$$

2.1.1.11. Robot no holonómico

De acuerdo con [40], un robot móvil se clasifica como no holonómico cuando presenta ligaduras cinemáticas que no pueden expresarse como restricciones puramente geométricas del espacio de configuración, es decir, restricciones que no son integrables y que dependen de las velocidades generalizadas del sistema.

Un ejemplo clásico de este tipo de sistemas es el robot móvil tipo automóvil, el cual consta de dos ruedas traseras motrices y dos ruedas delanteras encargadas de la dirección, las cuales únicamente permiten el rodamiento en la dirección a la que están orientadas, sin generar tracción. Este modelo cinemático se muestra en la figura (B.4) y es ampliamente utilizado en la literatura para el estudio de sistemas no holonómicos.

En este modelo, las ruedas delanteras permiten el cambio de orientación del vehículo mediante el ángulo de dirección, mientras que las ruedas traseras generan el movimiento longitudinal. Debido a la geometría del sistema y al contacto rueda-suelo, se asume que no existe deslizamiento lateral en las ruedas, lo cual impone una restricción cinemática que impide el movimiento lateral instantáneo del robot. Esta restricción se ilustra en la figura (B.5).

La restricción al movimiento lateral es de tipo no holonómico, ya que corresponde a una relación cinemática lineal en las velocidades generalizadas que no puede integrarse para obtener una restricción del espacio de configuración. A partir del modelo de la figura (B.4), las componentes de la velocidad del robot móvil en los ejes X e Y se expresan como

$$\dot{x} = v \cos(\phi), \quad (\text{B.11})$$

$$\dot{y} = v \sin(\phi). \quad (\text{B.12})$$

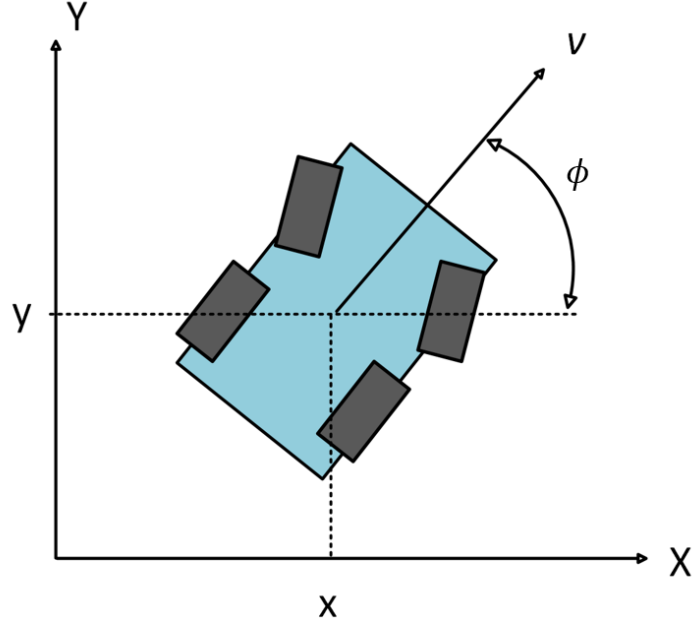


Figura B.4. Modelo cinemático de un robot móvil tipo automóvil.

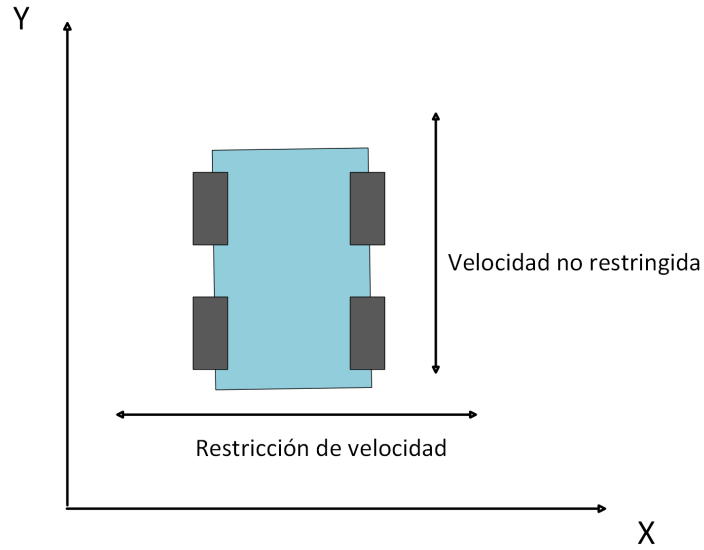


Figura B.5. Restricción cinemática de no deslizamiento lateral en un robot móvil tipo automóvil.

Despejando la velocidad longitudinal v de la ecuación (B.12) se obtiene

$$v = \frac{\dot{y}}{\sin(\phi)}. \quad (\text{B.13})$$

Sustituyendo (B.13) en (B.11) se obtiene

$$\dot{x} = \frac{\dot{y}}{\sin(\phi)} \cos(\phi), \quad (\text{B.14})$$

la cual puede reescribirse como la siguiente restricción cinemática:

$$\dot{x} \sin(\phi) - \dot{y} \cos(\phi) = 0. \quad (\text{B.15})$$

La ecuación (B.15) puede expresarse en forma matricial como

$$\begin{bmatrix} 0 & \sin(\phi) & -\cos(\phi) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{x} \\ \dot{y} \end{bmatrix} = 0. \quad (\text{B.16})$$

La expresión (B.16) corresponde a una restricción de tipo Pfaffian, ya que es una relación lineal en las velocidades generalizadas. Para analizar su integrabilidad, se define el vector de coordenadas generalizadas del sistema como

$$q = (\phi, x, y)^T, \quad (\text{B.17})$$

y se asume la existencia de una función escalar $g(q) = g(\phi, x, y)$, definida sobre el espacio de configuración del robot, tal que su diferencial total satisface

$$dg = \frac{\partial g}{\partial \phi} d\phi + \frac{\partial g}{\partial x} dx + \frac{\partial g}{\partial y} dy. \quad (\text{B.18})$$

Para que la restricción de Pfaffian sea holonómica, debe existir una función escalar $g(q)$ cuyo diferencial coincida con los coeficientes de la restricción. En este caso, dicha condición implicaría que

$$\frac{\partial g(\phi, x, y)}{\partial \phi} = 0, \quad (\text{B.19})$$

$$\frac{\partial g(\phi, x, y)}{\partial x} = \sin(\phi), \quad (\text{B.20})$$

$$\frac{\partial g(\phi, x, y)}{\partial y} = -\cos(\phi). \quad (\text{B.21})$$

Dado que no existe una función escalar $g(\phi, x, y)$ que satisfaga simultáneamente las ecuaciones (B.19) a (B.21), se concluye que la restricción no es integrable y, por lo tanto, el sistema es no holonómico [40].

Es importante destacar que, aunque esta ligadura restringe las velocidades instantáneas del sistema, no limita el espacio de trabajo del robot, ya que el vehículo puede alcanzar cualquier punto del plano X - Y mediante trayectorias adecuadas.

Finalmente, el modelo del robot móvil tipo automóvil cuenta únicamente con dos entradas de control: la velocidad longitudinal y el ángulo de dirección. Dado que el sistema posee

tres grados de libertad en el plano y solo dos entradas de control, se clasifica como un sistema subactuado.

REFERENCIAS

- [1] K. A. S. Ronald Ping Man Chan y C. R. Halkyard, “Review of modelling and control of two-wheeled robots”, *Annual Reviews in Control*, vol. 37, n.º 1, págs. 89-103, 2013, ISSN: 1367-5788. DOI: <https://doi.org/10.1016/j.arcontrol.2013.03.004> dirección: <https://www.sciencedirect.com/science/article/pii/S1367578813000060>
- [2] N. T y P. K T, “PID Controller Based Two Wheeled Self Balancing Robot”, en *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, 2021, págs. 1-4. DOI: [10.1109/ICOEI51242.2021.9453091](https://doi.org/10.1109/ICOEI51242.2021.9453091)
- [3] X. Zhang y S. Cheng, “Self-balancing and Velocity Control of Two-Wheeled Mobile Robot Based on LQR Sliding Mode”, en *2019 Chinese Automation Congress (CAC)*, 2019, págs. 1475-1480. DOI: [10.1109/CAC48633.2019.8997059](https://doi.org/10.1109/CAC48633.2019.8997059)
- [4] S.-K. Kim, C. K. Ahn y R. K. Agarwal, “Position-Tracking Controller for Two-Wheeled Balancing Robot Applications Using Invariant Dynamic Surface”, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, n.º 2, págs. 705-711, 2021. DOI: [10.1109/TSMC.2018.2882868](https://doi.org/10.1109/TSMC.2018.2882868)
- [5] S.-K. Kim y C. K. Ahn, “Self-Tuning Position-Tracking Controller for Two-Wheeled Mobile Balancing Robots”, *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, n.º 6, págs. 1008-1012, 2019. DOI: [10.1109/TCSII.2018.2869915](https://doi.org/10.1109/TCSII.2018.2869915)
- [6] C. Iwendi, M. A. Alqarni, J. H. Anajemba, A. S. Alfakeeh, Z. Zhang y A. K. Bashir, “Robust Navigational Control of a Two-Wheeled Self-Balancing Robot in a Sensed Environment”, *IEEE Access*, vol. 7, págs. 82337-82348, 2019. DOI: [10.1109/ACCESS.2019.2923916](https://doi.org/10.1109/ACCESS.2019.2923916)
- [7] C.-H. G. Li, L.-P. Zhou e Y.-H. Chao, “Self-Balancing Two-Wheeled Robot Featuring Intelligent End-to-End Deep Visual-Steering”, *IEEE/ASME Transactions on Mechatronics*, vol. 26, n.º 5, págs. 2263-2273, 2021. DOI: [10.1109/TMECH.2020.3036579](https://doi.org/10.1109/TMECH.2020.3036579)
- [8] A. Unluturk y O. Aydogdu, “Machine Learning Based Self-Balancing and Motion Control of the Underactuated Mobile Inverted Pendulum With Variable Load”, *IEEE Access*, vol. 10, págs. 104706-104718, 2022. DOI: [10.1109/ACCESS.2022.3210540](https://doi.org/10.1109/ACCESS.2022.3210540)

- [9] G. A. Muñoz-Hernandez, J. Díaz-Téllez, J. Estevez-Carreón y R. S. García-Ramírez, “ADRC Attitude Controller Based on ROS for a Two-Wheeled Self-Balancing Mobile Robot”, *IEEE Access*, vol. 11, págs. 94 636-94 646, 2023. DOI: 10.1109/ACCESS.2023.3308948
- [10] M. T. Watson, D. T. Gladwin, T. J. Prescott y S. O. Conran, “Dual-Mode Model Predictive Control of an Omnidirectional Wheeled Inverted Pendulum”, *IEEE/ASME Transactions on Mechatronics*, vol. 24, n.º 6, págs. 2964-2975, 2019. DOI: 10.1109/TMECH.2019.2943708
- [11] M. Wahab, F. Rios-Gutierrez y A. El Shahat, “Energy modeling of differential drive robots”, en *SoutheastCon 2015*, 2015, págs. 1-6. DOI: 10.1109/SECON.2015.7133023
- [12] A. Stefek, T. V. Pham, V. Krivanek y K. L. Pham, “Energy Comparison of Controllers Used for a Differential Drive Wheeled Mobile Robot”, *IEEE Access*, vol. 8, págs. 170 915-170 927, 2020. DOI: 10.1109/ACCESS.2020.3023345
- [13] J. Pliego-Jiménez, R. Martínez-Clark, C. Cruz-Hernández y A. Arellano-Delgado, “Trajectory tracking of wheeled mobile robots using only Cartesian position measurements”, *Automatica*, vol. 133, pág. 109 756, 2021, ISSN: 0005-1098. DOI: <https://doi.org/10.1016/j.automatica.2021.109756> dirección: <https://www.sciencedirect.com/science/article/pii/S0005109821002764>
- [14] Y. Kanayama, Y. Kimura, F. Miyazaki y T. Noguchi, “A stable tracking control method for an autonomous mobile robot”, en *Proceedings., IEEE International Conference on Robotics and Automation*, 1990, 384-389 vol.1. DOI: 10.1109/ROBOT.1990.126006
- [15] W. Dixon, M. de Queiroz y D. Dawson, “Adaptive tracking and regulation of a wheeled mobile robot with controller/update law modularity”, en *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 3, 2002, 2620-2625 vol.3. DOI: 10.1109/ROBOT.2002.1013626
- [16] Y. S. Ha y S. Yuta, “Trajectory tracking control for navigation of the inverse pendulum type self-contained mobile robot”, *Robotics and Autonomous Systems*, vol. 17, n.º 1-2, págs. 65-80, abr. de 1996, ISSN: 0921-8890. DOI: 10.1016/0921-8890(95)00062-3
- [17] C.-C. Tsai, Y.-X. Li y F.-C. Tai, “Backstepping sliding-mode leader-follower consensus formation control of uncertain networked heterogeneous nonholonomic wheeled mobile multirobots”, en *2017 56th Annual Conference of the Society of*

- Instrument and Control Engineers of Japan (SICE)*, 2017, págs. 1407-1412. DOI: 10.23919/SICE.2017.8105661
- [18] C.-C. Tsai, C.-K. Chan e Y.-H. Fan, “Planned navigation of a self-balancing autonomous service robot”, en *2008 IEEE Workshop on Advanced robotics and Its Social Impacts*, 2008, págs. 1-6. DOI: 10.1109/ARSO.2008.4653617
- [19] A. Hess y O. Medina, “A Self-Balancing Robot With Redundant Actuators for Obstacle Avoidance”, *IEEE Access*, vol. 13, págs. 134376-134384, 2025. DOI: 10.1109/ACCESS.2025.3590812
- [20] S. R. Garces, M. Y. Beb, A. Boubakari, H. H. Ammar y M. A. W. Shalaby, “Hybrid Self-Balancing and object Tracking Robot Using Artificial Intelligence and Machine Vision”, en *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, 2020, págs. 264-269. DOI: 10.1109/NILES50944.2020.9257916
- [21] K. Brkic y Z. Kovacic, “Decoupled control and path tracking of a two-wheeled self-balancing mobile robot”, en *2009 IEEE International Symposium on Industrial Electronics*, 2009, págs. 642-647. DOI: 10.1109/ISIE.2009.5214743
- [22] A. Ghahremani y A. K. Khalaji, “Simultaneous regulation and velocity tracking control of a two-wheeled self-balancing robot”, en *2023 International Conference on Control, Automation and Diagnosis (ICCAD)*, 2023, págs. 1-6. DOI: 10.1109/ICCAD57653.2023.10152339
- [23] H. N. Le, M.-K. Pham, D.-H. Pham y T.-V.-A. Nguyen, “Trajectory tracking and stabilization of two-wheeled balancing mobile robot with hierarchical and sliding mode control”, *International Journal of Dynamics and Control*, vol. 13, n.º 1, pág. 10, 2024, ISSN: 2195-2698. DOI: 10.1007/s40435-024-01518-0
- [24] L. Meirovitch, “Fundamentals of Analytical Mechanics”, en *Methods of Analytical Dynamics*. 1970, págs. 45-100.
- [25] B. Mahler y J. Haase, “Mathematical model and control strategy of a two-wheeled self-balancing robot”, en *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, 2013, págs. 4198-4203. DOI: 10.1109/IECON.2013.6699809
- [26] F. A. Azad, M. Reza Hairi Yazdi y M. T. Masouleh, “Kinematic and Dynamic Analysis of 3-DOF Delta Parallel Robot Based on the Screw Theory and Principle of Virtual Work”, en *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, 2019, págs. 717-724. DOI: 10.1109/KBEI.2019.8734994

- [27] H. D. Taghirad, “Jacobian Matrix”, en *Fundamentals of Robotics Applied Case of Studies with Matlab & Python*. 2025, págs. 173-199.
- [28] A. Kelly, “Dynamics”, en *Mobile Robotics: Mathematics, Models, and Methods*. 2013, págs. 173-269.
- [29] H. A. Secchi, “Una introducción a los robots móviles”, en *Una introducción a los robots móviles*. 2008.
- [30] C.-T. Chen, *Linear System Theory and Design*, 3rd. USA: Oxford University Press, Inc., 1998, ISBN: 0195117778.
- [31] R. Siegwart, I. R. Nourbakhsh y D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd. The MIT Press, 2011, ISBN: 0262015358.
- [32] R. Kelly y V. Santibáñez, *Control de movimiento de robots manipuladores* (Automática robótica). Pearson Educación, 2003, ISBN: 9788420538310.
- [33] Z.-P. Jiang, “Controlling Underactuated Mechanical Systems: A Review and Open Problems”, en *Advances in the Theory of Control, Signals and Systems with Physical Modeling*, J. Lévine y P. Müllhaupt, eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, págs. 77-88, ISBN: 978-3-642-16135-3. DOI: 10.1007/978-3-642-16135-3_7 dirección: https://doi.org/10.1007/978-3-642-16135-3_7
- [34] R. W. Brockett et al., “Asymptotic stability and feedback stabilization”, *Differential geometric control theory*, vol. 27, n.º 1, págs. 181-191, 1983.
- [35] J. P. Hespanha y A. Morse, “Switching between stabilizing controllers”, *Automatica*, vol. 38, n.º 11, págs. 1905-1917, 2002, ISSN: 0005-1098. DOI: [https://doi.org/10.1016/S0005-1098\(02\)00139-5](https://doi.org/10.1016/S0005-1098(02)00139-5) dirección: <https://www.sciencedirect.com/science/article/pii/S0005109802001395>
- [36] R. G. Castro Salguero, “Sistemas Mecánicos Subactuados”, Sección de Posgrado. Asesor: PhD Arturo Rojas, Tesis de Maestría, Universidad Nacional de Ingeniería, Facultad de Ingeniería Eléctrica y Electrónica, mar. de 2001.
- [37] J. J. Maldonado Ramos, R. Garrido y G. Castro Zavala, “Real-time control prototype and its use in Automatic Control courses”, en *Memorias del Congreso Nacional de Control Automático*, 2020, págs. 1-6.
- [38] R. C. Dorf y R. H. Bishop, *Modern Control Systems*, 13th. USA: Pearson, 2017.
- [39] A. A. Muñoz Flores, R. Alvarez Salas, A. Cárdenas, M. A. González García y V. Cárdenas, “Modeling and Control of a Mobile Robot”, en *Congreso Latinoamericano de Control Automático*, 2025.

- [40] A. M. Bloch, *Nonholonomic Mechanics and Control* (Interdisciplinary Applied Mathematics). Springer, 2003.

