



"El saber de mis hijos
hará mi grandeza"

UNIVERSIDAD DE SONORA

División de Ciencias Exactas y Naturales

Departamento de Investigación en Física

Maestría en Ciencias en Electrónica

CONTROL DE ESTABILIZACIÓN Y SEGUIMIENTO EN PÉNDULO SOBRE DOS RUEDAS

T E S I S

Presentada en cumplimiento de los requisitos para obtener el grado de
Maestro en Ciencias en Electrónica
presenta:

Ing. Fernando Morales Cabrera.

Director: **Dr. Luis Arturo García Delgado.**
Co-Director: **Dr. Jorge Alberto Orrante Sakanassi**

Hermosillo, Sonora, México, Agosto 2022

Universidad de Sonora

Repositorio Institucional UNISON



**"El saber de mis hijos
hará mi grandeza"**



Excepto si se señala otra cosa, la licencia del ítem se describe como openAccess

Índice general

Agradecimientos.	4
Dedicación.	4
1. Introducción	11
1.1. Objetivos	11
1.1.1. Objetivo general	11
1.1.2. Objetivos específicos	11
1.1.3. Justificación	12
1.2. Estado del arte	12
1.2.1. Antecedentes	12
1.3. Marco Teórico	15
1.3.1. Leyes de Newton	15
1.3.2. Mecánica Lagrangiana	15
2. Modelado Matemático	17
2.1. Modelo de Euler-Lagrange	17
2.1.1. Coordenadas Generalizadas	17
2.1.2. Fuerzas Generalizadas	20
2.1.3. Ecuaciones Euler-Lagrange	21
2.1.4. Modelado del motor de corriente continua	23
2.1.5. Linealización del sistema	25
2.1.6. Representación en forma matricial de la ecuación de estado	27
3. Estrategia de control	29
3.1. Control por realimentación de estado	29
3.1.1. Asignación de polos y realimentación de estados	30
3.1.2. Matriz de controlabilidad	31
3.1.3. Ecuación característica	31
3.1.4. Criterios de diseño	32
3.1.5. Determinación de la matriz de ganancias de realimentación \mathbf{K}	33
3.2. Diseño de un servosistema tipo 1	34
3.2.1. Características de la respuesta a un escalón unitario del sistema diseñado	36
4. Construcción del prototipo	37
4.1. Componentes utilizados	37
4.1.1. Motorreductores	37
4.1.2. Ruedas	38
4.1.3. Sensor inercial	38
4.1.4. Tarjetas de control	38

4.1.5. Controlador de potencia para motores	39
4.1.6. Batería	39
4.1.7. Cuerpo del PISDR	39
4.2. Prototipo construido	40
5. Resultados	43
5.1. Simulación del control por realimentación de estado	43
5.1.1. Simulación entorno de realidad virtual	43
5.2. Simulación de servosistema tipo 1	45
5.2.1. Simulación entorno de realidad virtual	45
5.3. Simulación por Controlador PID	46
6. Conclusiones	49
Apéndice	53
6.1. Código para diseñar sistema de realimentación de estado (sistema tipo regulador) . .	53
6.2. Código para diseño de servosistema tipo 1	54
6.3. Función de MATLAB del modelo del sistema para Simulink	56
6.4. Códigos de Arduino utilizados en la plataforma experimental	56
6.4.1. Código Arduino esclavo	56
6.4.2. Código Arduino maestro	58

Lista de Simbolos.

\mathcal{K}	Energía cinética total del sistema.
\mathcal{U}	Energía potencial total del sistema.
\mathcal{K}_p	Energía cinética total del Cuerpo del péndulo.
\mathcal{K}_r	Energía cinética total de la rueda.
\mathcal{U}_p	Energía potencial total del cuerpo del péndulo.
\mathcal{U}_r	Energía potencial total de la rueda.
m_r	Masa de la rueda.
m_p	Masa del péndulo.
r	Radio de la rueda.
R_a	Resistencia de armadura.
i_a	Corriente de armadura.
L_a	Inductancia de armadura.
v_a	Voltaje de armadura.
e	Fuerza contra-electromotriz.
l	Longitud al centro de masa del péndulo.
τ_m	Torque del motor.
u	Señal de entrada.
θ	Ángulo del péndulo respecto a la vertical.
$\dot{\theta}$	velocidad angular del péndulo respecto a la vertical.
φ	Ángulo girado por el eje de los motores.
$\dot{\varphi}$	Velocidad de giro del eje de los motores.
μ_e	Constante de fricción viscosa del eje del motor.
μ_s	Constante de fricción con el suelo.

Agradecimientos

Me gustaría expresar mi más sincero agradecimiento al Dr. Luis Arturo García Delgado, director de esta tesis y mi maestro desde que cursaba la licenciatura. Él creyó en mi y me brindó su apoyo de manera personal e institucional alentándome a concluir este trabajo. Resalto su calidad de profesor, director y sobre todo gran ser humano.

También estoy agradecido con el Dr. Jorge Alberto Orrante Sakanassi que formó parte en la co-dirección de este trabajo y quien aportó su conocimiento y experiencia en este proyecto.

A mi comité de revisión, Dr. Roberto Gómez Fuentes, Dra. Alicia Vera Marquina y el Dr. Alejandro García Juárez. Por su aportación y observaciones en este proyecto desde el inicio del mismo.

A todos mis profesores que formaron parte en mi formación en el posgrado, a mi institución la Universidad de Sonora que brindó recursos para algunos componentes de este trabajo.

Finalmente agradezco a mi familia y Mara quien me acompañó en algunos momentos de esta travesía. Sobre todo muchas gracias a Dios que me permitió concluir este trabajo.

Dedicación

Dedico esta tesis a Dios y mi familia por ayudarme a culminar esta obra, a Mara quien fuera mi pareja y estuvo en algunos momentos claves de este trabajo.

Capítulo 1

Introducción

El presente capítulo expone los antecedentes sobre el tema central de la tesis: sistemas de péndulo invertido sobre dos ruedas (PISDR), comenzando por los objetivos y la justificación, siguiendo con el actual estado del arte.

El sistema PISDR consiste en un robot móvil, carro de dos ruedas alineadas entre sí sobre su eje de rotación, y es de tipo diferencial o también conocido como tipo monociclo, el cual balancea una masa cuyo centro de gravedad está por encima del eje de las ruedas, de modo que el centro de masa se puede modelar como un péndulo. El objetivo de control consiste en mantener el centro de masa de la carga en la posición vertical hacia arriba, que es el punto de equilibrio inestable, a la vez que se pueda controlar la posición lineal del carro. El PISDR es un sistema subactuado e inestable en lazo abierto.

1.1. Objetivos

1.1.1. Objetivo general

Implementar un sistema capaz de estabilizar y dar seguimiento a un sistema subactuado, como lo es el péndulo invertido sobre dos ruedas (PISDR).

1.1.2. Objetivos específicos

- Desarrollar un algoritmo de control que sea capaz de estabilizar el PISDR.
- Diseñar y construir un sistema PISDR.
- Lograr la estabilización del vehículo en ángulo.

1.1.3. Justificación

El problema de control de los sistemas subactuados es de gran interés, debido a que se poseen más grados de libertad que entradas de control. El sistema PISDR es un prototipo que se suele utilizar como modelo de sistema subactuado por la parte del péndulo y como robot móvil.

Para lograr controlar un sistema subactuado de este tipo, es preciso desarrollar u obtener un buen modelo matemático, que pueda tomar en cuenta las dinámicas más importantes del sistema, mediante lo cual se permite el desarrollo de controladores basados en el modelo. El método de Euler-Lagrange es una técnica adecuada que permite el modelado dinámico de sistemas mecánicos, por lo cual será utilizado para modelar el sistema.

Un modelo matemático adecuado permite predecir mediante simulación computacional el comportamiento del sistema con los controladores diseñados.

Con estas bases científicas y motivaciones en cuanto al reto del problema, se plantean soluciones al problema del PISDR mediante controladores lineales por realimentación de estado, así como con combinaciones de PID.

1.2. Estado del arte

Los sistemas subactuados péndulos invertidos han sido ampliamente estudiados a lo largo del tiempo en el tema de teoría de control. El carro péndulo es un modelo de estudio en las materias de sistemas de control automático. Este tipo de sistemas siguen teniendo vigencia en la investigación sobre controladores de estabilización y seguimiento.

1.2.1. Antecedentes

Se encuentran registros que en el año 1976, Furuta y sus colaboradores realizaron el control de un péndulo invertido que se desplazaba dentro de un monoriel, siguiendo una trayectoria horizontal rectilínea para mantener el equilibrio sobre la vertical. El método de control propuesto fue un controlador por realimentación lineal en combinación con un observador de estado [1].

En el año 1992 K. Furuta, M. Yamakita y S. Kobayashi trabajaron en un péndulo invertido mediante un controlador de pseudo estados. Éste es un ejemplo de oscilador no lineal estudiado en los sistemas de control. Dicho péndulo se volvió muy conocido con el paso de los años y en la actualidad se le conoce como *péndulo de Furuta* [2].

Con el pasar del tiempo han surgido diferentes trabajos partiendo de lo propuesto por el principio del péndulo de Furuta. Se han ido adaptando nuevos parámetros al modelo propuesto, llegándose a proponer el doble péndulo de Furuta [3]. Dicho sistema tiene un actuador para mover el brazo y las articulaciones entre el brazo y el péndulo inferior, y entre el fondo péndulo y el de arriba son articulaciones libres, tal como se observa en la Figura 1.1.

Otra propuesta de péndulo invertido es el péndulo invertido sobre dos ruedas (PISDR), cuyos antecedentes se presentan a continuación. En el año 2002, en el Laboratorio de Electrónica industrial del Instituto Federal Suizo, se diseñó un péndulo invertido a escala, con pesos unidos en la barra del

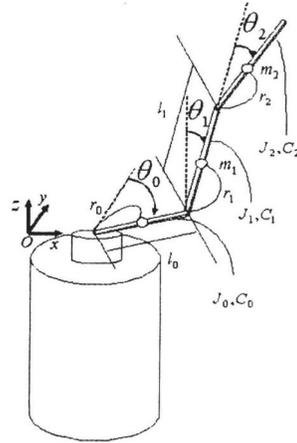


Figura 1.1: Diagrama de cuerpo libre de doble péndulo de Furuta [3].

péndulo para simular el peso de una persona. El prototipo fue construido con un peso en el lugar de un conductor y en escala reducida, para ahorrar costos y no poner en riesgo a pilotos de prueba [4]. En la Figura 1.2 se muestra el prototipo PSDR JOE.



Figura 1.2: Péndulo invertido JOE [4].

En 2003 se había desarrollado un péndulo invertido balancín en la Universidad de Australia. En el trabajo se presentan los procesos desarrollados y las consideraciones de un robot autónomo sobre dos ruedas basado en el modelo de péndulo invertido [5]. El prototipo del trabajo al que se hace mención se muestra en la Figura 1.3.

Otro de los trabajos realizados sobre esta variante de péndulo invertido es el de UMBRA, en el cual se emplea una ley de control tipo PID para estabilizarlo [6]. Una imagen de este sistema se muestra en la Figura 1.4.

Este tipo de péndulos invertidos han servido como plataforma experimental para estudio de diferentes técnicas de control tanto lineales como no lineales. Un ejemplo de ello fue el diseño de

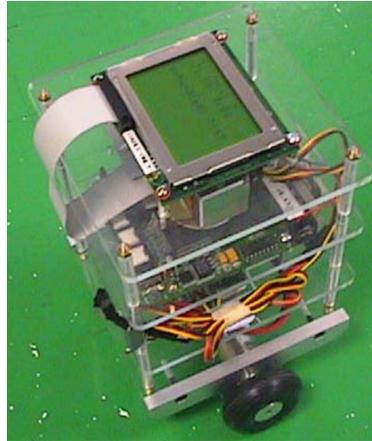


Figura 1.3: Robot autónomo auto-equilibrado [5].



Figura 1.4: Péndulo invertido UMBRA [6].

controlador mediante el método de Lyapunov, en el cual el objetivo primordial es proponer un controlador que estabilice asintóticamente el sistema alrededor del punto de equilibrio inestable [7].

De los más recientes trabajos de investigación sobre el tema está el trabajo publicado en 2021, en el que el análisis del modelo matemático parte de la formulación usando el principio de D'Alembert, después el algoritmo de control se basa en un análisis del tiempo discreto [8]. Un trabajo más del mismo año fue basado en el modelo de carro péndulo, donde el análisis matemático parte de una formulación Newtoniana y el control que se estudia para el diseño del sistema es un control óptimo cuadrático, capaz de hacer que el sistema tenga una respuesta más rápida, lográndose un sobreimpulso pequeño y un error de estado estacionario demasiado bajo [9].

En 2022, Messikh y sus colaboradores, analizan el ya conocido carro péndulo, partiendo de una modificación de un controlador por realimentación de estado convirtiéndolo a un control por realimentación de salida sin observador [10].

1.3. Marco Teórico

1.3.1. Leyes de Newton

Hay tres leyes muy conocidas en la mecánica clásica, las *leyes de Newton*. **La primera ley de Newton**, que trata de la conservación de la cantidad de movimiento, establece que la cantidad de movimiento, (*momentum*) total de un sistema mecánico es constante en ausencia de fuerzas externas. La cantidad de movimiento es el producto de la masa m y la velocidad v , o mv , para el momento de traslación o lineal. En el movimiento rotacional, la cantidad de movimiento es el producto del momento de inercia J y la velocidad angular ω , o $J\omega$, y se denomina cantidad de movimiento angular [11].

Segunda ley de Newton (del movimiento traslacional). Para un movimiento traslacional, la segunda ley de Newton dice que la aceleración de cualquier cuerpo rígido es directamente proporcional a la fuerza que actúe sobre él e inversamente proporcional a la masa del cuerpo. Esto es

$$\text{Fuerza} = (\text{masa})(\text{aceleración})$$

Supóngase que unas fuerzas actúan sobre una masa m . Si $\sum F$ es la suma de todas fuerzas que actúan en una dirección dada, entonces

$$\sum F = ma \tag{1.1}$$

Segunda ley de Newton (del movimiento rotacional). Para un cuerpo rígido en rotación pura alrededor de un eje fijo, la segunda ley de Newton establece que

$$\sum \text{Pares} = (\text{momento de inercia})(\text{aceleración angular}) \tag{1.2}$$

o bien,

$$\sum T = J\alpha \tag{1.3}$$

donde $\sum T$ es la suma de todos los pares que actúan alrededor de un eje dado, J es el momento de inercia del cuerpo alrededor de ese eje y α es la aceleración angular.

La segunda ley de Newton da la relación fuerza-aceleración de un cuerpo rígido o la relación de aceleración angular-par de un cuerpo rígido en rotación.

La **tercera ley de Newton** se refiere a la de acción y reacción, y en efecto, establece que a toda acción se opone una reacción de igual magnitud.

1.3.2. Mecánica Lagrangiana

La aplicación directa de las leyes de Newton al movimiento de sistemas mecánicos puede ser sustituida por una propuesta general, un método muy elegante y sofisticado para encontrar las ecuaciones de movimiento para todos los sistemas dinámicos, desarrollado por el matemático Joseph Louis Lagrange [12].

Para formular el Lagrangiano de un sistema, se debe calcular primero su energía cinética, \mathcal{K} , la cual comúnmente se expresa como la relación de

$$\mathcal{K}_{lineal} = \frac{1}{2}mv^2$$

donde m denota la masa del objeto y v denota su velocidad lineal. Si el movimiento es angular, la energía cinética se calcula como

$$\mathcal{K}_{angular} = \frac{1}{2}J\omega^2$$

donde J expresa la inercia del objeto y ω su velocidad angular.

La energía potencial debida a la gravedad, \mathcal{U} , comúnmente es expresada por una ecuación del tipo

$$\mathcal{U} = mgh$$

donde h es la altura del centro de masa del objeto.

Denotemos el vector de variables generalizadas, ya sean lineales o angulares, mediante q , y el vector de velocidades generalizadas mediante \dot{q}

La diferencia entre la energía cinética y la energía potencial será llamada el *Lagrangiano* del sistema, expresado matemáticamente por

$$\mathcal{L}(q_i, \dot{q}_i, t) = \mathcal{K}(\dot{q}_i, t) - \mathcal{U}(q_i, t) \quad (1.4)$$

A partir del Lagrangiano, se pueden formular las ecuaciones de movimiento de *Euler-Lagrange*, mediante la ecuación siguiente

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_k} - \frac{\partial \mathcal{L}}{\partial q_k} = \tau_k ; \quad k = 1, \dots, n \quad (1.5)$$

donde τ_k es la fuerza generalizada asociada con q_k , y n son los *grados de libertad* del sistema.

Capítulo 2

Modelado Matemático

En este capítulo se presenta el modelado matemático del sistema péndulo invertido sobre dos ruedas, en el cual, el análisis se hará por el método de Euler-Lagrange, con lo que partiendo de dos coordenadas generalizadas, se calculará la energía potencial y cinética, tanto de las llantas como del cuerpo de péndulo, para la construcción del Lagrangiano. Se tomará en cuenta el motorreductor que acciona cada rueda, así como las constantes de fricción, para el modelo completo del sistema.

2.1. Modelo de Euler-Lagrange

En esta sección se analiza el diagrama de cuerpo libre del PISDR, para posteriormente encontrar las ecuaciones que definen el sistema por el método de Euler-Lagrange y, para encontrar las ecuaciones de movimiento mediante mecánica Lagrangiana. Se parte por definir las coordenadas generalizadas del sistema, así como su Lagrangiano. Se asume que toda la masa se encuentra en un sólo plano bidimensional (x, y) paralelo a las ruedas y que éstas giran a la misma velocidad, por lo tanto se tiene un modelo reducido del sistema.

En el diagrama de la Figura 2.1, θ es el ángulo de inclinación del péndulo con respecto al eje de giro, x el desplazamiento lineal del eje de las ruedas, m_p , m_r , J_p , J_r la masa y momento de inercia del péndulo y de la rueda respectivamente, φ el ángulo de inclinación de las ruedas, τ_m es el torque ejercido sobre la llanta, g la fuerza de gravedad ejercida en el péndulo y l denota la longitud del péndulo.

2.1.1. Coordenadas Generalizadas

Como el sistema tiene dos grados de libertad, se necesita de dos coordenadas generalizadas para describir la dinámica del sistema. Por lo tanto, se toma como primer coordenada generalizada el ángulo θ , formado por el cuerpo del péndulo y, como segunda coordenada generalizada, el ángulo φ que es el ángulo de inclinación de la rueda.

$$q_1 = \theta \tag{2.1}$$

$$q_2 = \varphi \tag{2.2}$$

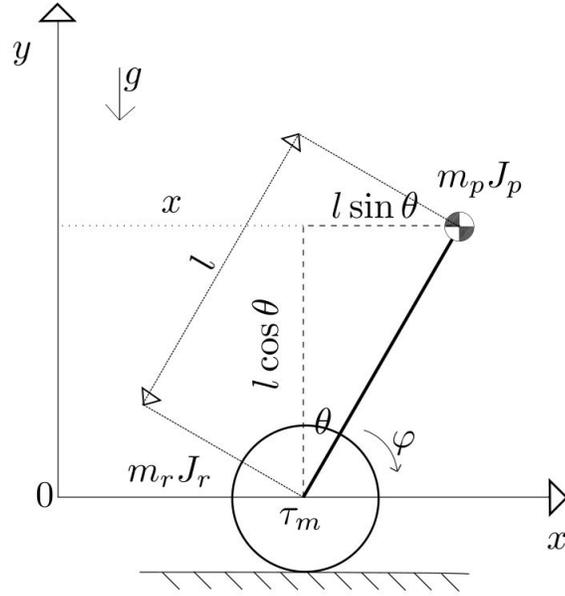


Figura 2.1: Diagrama de cuerpo libre PISDR.

Debido a que se restringe el movimiento del PISDR al plano $x - y$, la posición lineal del carro se relaciona con el ángulo de la rueda, q_2 , mediante

$$\varphi = q_2 = \frac{x}{r}, \quad \text{o} \quad x = r q_2 \quad (2.3)$$

donde r denota el radio de la rueda y x la posición lineal del eje de las ruedas. A su vez, la posición del centro de masa del péndulo se calcula como

$$x_1 = x + l \sin q_1 \quad (2.4)$$

$$y_1 = l \cos q_1 \quad (2.5)$$

Las velocidades generalizadas son, entonces

$$\begin{aligned} \dot{q}_1 &= \dot{\theta} \\ \dot{q}_2 &= \dot{\varphi} = \frac{\dot{x}}{r} \end{aligned}$$

La energía cinética total del cuerpo del péndulo es la suma de la energía cinética traslacional más la energía cinética rotacional:

$$\mathcal{K}_p = \mathcal{K}_{tras} + \mathcal{K}_{rot} \quad (2.6)$$

La energía cinética traslacional se calcula de la expresión

$$\begin{aligned}
\mathcal{K}_{tras} &= \frac{1}{2}m_p\vec{V}_p^2 = \frac{1}{2}m_p(\dot{x}_1^2 + \dot{y}_1^2) \\
&= \frac{1}{2}m_p\left[(\dot{x} + l\dot{q}_1 \cos q_1)^2 + (-l\dot{q}_1 \sin q_1)^2\right] \\
&= \frac{1}{2}m_p\left[\dot{x}^2 + 2xl\dot{q}_1 \cos q_1 + l^2\dot{q}_1^2 \cos^2 q_1 + (l^2\dot{q}_1^2 \sin^2 q_1)\right] \\
&= \frac{1}{2}m_p\left[\dot{x}^2 + 2xl\dot{q}_1 \cos q_1 + (l^2\dot{q}_1^2) \overbrace{(\cos^2 q_1 + \sin^2 q_1)}{=1}\right] \\
&= \frac{1}{2}m_p\left[\dot{x}^2 + 2xl\dot{q}_1 \cos q_1 + l^2\dot{q}_1^2\right]
\end{aligned} \tag{2.7}$$

donde \vec{V}_p denota el vector de velocidad lineal del péndulo, m_p denota la masa del péndulo y l representa la distancia desde el eje de las ruedas hasta el centro de masa del péndulo.

Mientras que, la energía cinética rotacional se calcula mediante

$$\mathcal{K}_{rot} = \frac{1}{2}J_p\vec{\omega}_p^2 = \frac{1}{2}J_p\dot{q}_1^2 \tag{2.8}$$

donde $\vec{\omega}_p$ es la velocidad angular del péndulo y J_p es el momento de inercia del péndulo.

Por lo tanto, utilizando las expresiones (2.7) y (2.8), en la Ecuación (2.6), se tiene

$$\mathcal{K}_p = \frac{1}{2}m_p\left[\dot{x}^2 + 2xl\dot{q}_1 \cos q_1 + l^2\dot{q}_1^2\right] + \frac{1}{2}J_p\dot{q}_1^2 \tag{2.9}$$

en la cual \mathcal{K}_p es la energía cinética del péndulo.

Ahora se procede a obtener la energía cinética total de la rueda, la cual es la suma de la energía cinética traslacional y rotacional de la rueda.

$$\mathcal{K}_r = \mathcal{K}_{rtras} + \mathcal{K}_{rrot} \tag{2.10}$$

aquí \mathcal{K}_r expresa la energía cinética de la rueda.

Para la energía traslacional de la rueda se tiene,

$$\begin{aligned}
\mathcal{K}_{rtras} &= \frac{1}{2}m_r\vec{v}_r^2 \\
&= \frac{1}{2}m_r(\dot{x} + \dot{y})^2 = \frac{1}{2}m_r(\dot{x} + 0)^2 \\
&= \frac{1}{2}m_r\frac{\dot{q}_2^2}{r^2}
\end{aligned} \tag{2.11}$$

en esta ecuación, m_r denota la masa de la rueda y r denota el radio de la rueda.

Para la energía rotacional de la rueda,

$$\mathcal{K}_{rrot} = \frac{1}{2}J_r\vec{\omega}_r^2 = \frac{1}{2}J_r\dot{q}_2^2 \tag{2.12}$$

donde J_r representa el momento de inercia de la rueda.

La energía cinética total del sistema es la suma de energía cinética total del péndulo y la energía cinética total de la rueda, por lo tanto

$$\begin{aligned}\mathcal{K} &= \mathcal{K}_p + \mathcal{K}_r \\ \mathcal{K} &= \frac{1}{2}m_p(\dot{x}^2 + 2\dot{x}\dot{\theta}l \cos \theta + l^2\dot{\theta}^2) + \frac{1}{2}J_p\dot{\theta}^2 + \frac{1}{2}m_r\frac{\dot{q}_2^2}{r^2} + \frac{1}{2}\frac{J_r}{r^2}\dot{x}^2\end{aligned}\quad (2.13)$$

Ahora, se procede a calcular la energía potencial del cuerpo del péndulo,

$$\begin{aligned}\mathcal{U}_p &= m_p g y_1 \\ \mathcal{U}_p &= m_p g l \cos q_1\end{aligned}\quad (2.14)$$

donde g es la constante de aceleración de la gravedad.

Debido a que la rueda se traslada en un plano horizontal, su energía potencial es nula, por lo que,

$$\mathcal{U}_r = m_r g y = 0 \quad (2.15)$$

La energía potencial total del sistema es la suma de la energía potencial del cuerpo del péndulo y la rueda,

$$\mathcal{U} = \mathcal{U}_p + \mathcal{U}_r = m_p g l \cos q_1 \quad (2.16)$$

Una vez obtenidas las energías cinética y potencial total del sistema, así como la energía disipada de las ruedas, se puede obtener su Lagrangiano, el cual se define por la diferencia de la energía cinética y la energía potencial, con lo cual se obtiene

$$\begin{aligned}\mathcal{L} &= \mathcal{K} - \mathcal{U} \\ &= \frac{1}{2}m_p(\dot{x}^2 + 2\dot{x}\dot{q}_1 l \cos \theta + l^2\dot{q}_1^2) + \frac{1}{2}J_p\dot{q}_1^2 + \frac{1}{2}m_r\dot{x}^2 + \frac{1}{2}\frac{J_r}{r^2}\dot{x}^2 - m_p g l \cos q_1\end{aligned}\quad (2.17)$$

2.1.2. Fuerzas Generalizadas

Las fuerzas generalizadas se pueden obtener a partir del cálculo del trabajo virtual, δW , de las fuerzas aplicadas. La definición de trabajo virtual se expresa de la siguiente manera

$$\delta W = \sum_{i=1}^n \mathbf{F}_i \delta \mathbf{r}_i \quad (2.18)$$

Por lo que las fuerzas generalizadas para el torque del motor se expresa

$$W_{\tau_m} = \tau_m (\varphi - \theta) = \tau_m \left(\frac{x}{r} - \theta \right) \quad (2.19)$$

$$Q_{\tau_m, x} = \frac{\partial W_{\tau_m}}{\partial x} = \frac{\tau_m}{r} \quad (2.20)$$

$$Q_{\tau_m, \theta} = \frac{\partial W_{\tau_m}}{\partial \theta} = \tau_m \quad (2.21)$$

Ahora para la Fuerza viscosa ejercida en el eje

$$W_{Fe} = -(\dot{\varphi} - \dot{\theta}) \mu_e (\varphi - \theta) \quad (2.22)$$

$$= -\left(\frac{\dot{x}}{r} - \dot{\theta}\right) \mu_e \left(\frac{x}{r} - \theta\right) \quad (2.23)$$

Por lo que las fuerzas generalizadas en el eje se expresa

$$Q_{Fe, x} = \frac{\partial W_{Fe}}{\partial x} = -\frac{\mu_e}{r} \left(\frac{\dot{x}}{r} - \dot{\theta}\right) \quad (2.24)$$

$$Q_{Fe, \theta} = \frac{\partial W_{Fe}}{\partial \theta} = \mu_e \left(\frac{\dot{x}}{r} - \dot{\theta}\right) \quad (2.25)$$

Para la fuerza de fricción debido al suelo

$$W_{Fs} = -\dot{x} \mu_s x \quad (2.26)$$

$$Q_{Fs, x} = \frac{\partial W_{Fs}}{\partial x} = -\dot{x} \mu_s \quad (2.27)$$

$$Q_{Fs, \theta} = \frac{\partial W_{Fs}}{\partial \theta} = 0 \quad (2.28)$$

2.1.3. Ecuaciones Euler-Lagrange

La ecuación que define la dinámica de movimiento puede ser descrita por la formulación lagrangiana como

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = Q_i \quad (2.29)$$

donde Q_i son las fuerzas generalizadas obtenidas con anterioridad.

Entonces, las ecuaciones de Euler-Lagrange, en función de las fuerzas obtenidas son

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{x}} \right) - \frac{\partial \mathcal{L}}{\partial x} = \frac{\partial w_{\tau_m}}{\partial x} + \frac{\partial w_{F_e}}{\partial x} + \frac{\partial w_{F_s}}{\partial x} \quad (2.30)$$

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}} \right) - \frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial w_{\tau_m}}{\partial \theta} + \frac{\partial w_{F_e}}{\partial \theta} + \frac{\partial w_{F_s}}{\partial \theta} \quad (2.31)$$

Calculando los términos de las ecuaciones de Lagrange para la posición lineal del carro se tiene

$$\frac{\partial \mathcal{L}}{\partial x} = 0 \quad (2.32)$$

$$\frac{\partial \mathcal{L}}{\partial \dot{x}} = m_p \dot{x} + m_p \dot{\theta} l \cos \theta + m_r \dot{x} + \frac{J_r}{r^2} \dot{x} \quad (2.33)$$

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{x}} \right) = \left(m_p + m_r + \frac{J_r}{r^2} \right) \ddot{x} + m_p \ddot{\theta} l \cos \theta - m_p \dot{\theta}^2 l \sin \theta \quad (2.34)$$

Y desarrollando los términos del Lagrangiano para el ángulo del péndulo, se tiene

$$\frac{\partial \mathcal{L}}{\partial \theta} = -m_p \dot{x} \dot{\theta} l \sin \theta + m_p g l \sin \theta \quad (2.35)$$

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}} = m_p \dot{x} l \cos \theta + m_p \dot{\theta} l^2 + J_p \dot{\theta} \quad (2.36)$$

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}} \right) = m_p \ddot{x} l \cos \theta - m_p \dot{x} \dot{\theta} l \sin \theta + (m_p l^2 + J_p) \ddot{\theta} \quad (2.37)$$

Por lo tanto, se obtienen las ecuaciones no lineales que representan la planta:

$$\left(m_p + m_r + \frac{J_r}{r^2} \right) \ddot{x} + m_p l (\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta) = \frac{\tau_m}{r} - \dot{x} \left(\frac{\mu_e}{r^2} + \mu_s \right) + \frac{\mu_e}{r} \dot{\theta} \quad (2.38)$$

$$(m_p l^2 + J_p) \ddot{\theta} + m_p \dot{x} l \cos \theta - m_p l g \sin \theta = -\tau_m + \mu_e \left(\frac{\dot{x}}{r} - \dot{\theta} \right) \quad (2.39)$$

Las expresiones (2.38) y (2.39) pueden escribirse en forma matricial de la siguiente forma:

$$D(\theta) \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} -\left(\frac{\mu_e}{r^2} + \mu_s \right) & m_p l \dot{\theta} \sin \theta + \frac{\mu_e}{r} \\ \frac{\mu_e}{r} & -\mu_e + m_p g l \sin \theta \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} \frac{\tau_m}{r} \\ -\tau_m \end{bmatrix} \quad (2.40)$$

en donde la matriz $D(\theta)$ se define mediante

$$D(\theta) = \begin{bmatrix} m_p + m_r + \frac{J_r}{r^2} & m_p l \cos \theta \\ m_p l \cos \theta & m_p l^2 + J_p \end{bmatrix}$$

Reacomodando y reidentificando las constantes de masas e inercias de la matriz $D(\theta)$ como nuevos parámetros, se obtiene lo siguiente:

$$I = \left(m_p + m_r + \frac{J_r}{r^2} \right)$$

$$J = \left(J_p + m_p l^2 \right)$$

Ahora despejando las aceleraciones de la Ecuación (2.40) y sustituyendo las nuevas constantes, se tienen las siguientes expresiones:

$$\ddot{x} = -\frac{m_p l}{I} \left(\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta \right) + \frac{1}{rI} \tau_m - \frac{\left(\mu_s + \frac{\mu_e}{r^2} \right)}{I} \dot{x} + \frac{\mu_e}{rI} \dot{\theta} \quad (2.41)$$

$$\ddot{\theta} = -\frac{m_p l}{J} \ddot{x} \cos \theta + \frac{m_p l g}{J} \sin \theta - \frac{1}{J} \tau_m + \frac{\mu_e}{J} \left(\frac{\dot{x}}{r} - \dot{\theta} \right) \quad (2.42)$$

2.1.4. Modelado del motor de corriente continua

La Figura 2.2 muestra el diagrama de la parte eléctrica y de la parte mecánica de un motor de corriente continua.

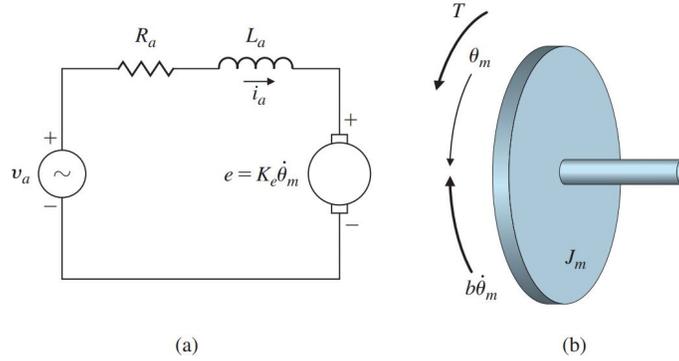


Figura 2.2: Diagramas a) parte eléctrica del motor; b) de cuerpo libre del rotor [13].

Las ecuaciones del motor dan el par τ_m en el rotor en términos de la corriente de armadura i_a y expresa el voltaje de la fuerza contra-elektromotriz e en términos de la velocidad de rotación del eje $\dot{\theta}_m$, mediante las siguientes expresiones

$$\tau_m = K_t i_a \quad (2.43)$$

$$e = K_e \dot{\theta}_m \quad (2.44)$$

El diagrama de cuerpo libre del rotor define la dirección positiva y los dos pares de fuerza aplicada, T y $b\dot{\theta}_m$. Aplicando la segunda ley Newton para la sumatoria de fuerzas:

$$J_m \ddot{\theta}_m + b \dot{\theta}_m = K_t i_a \quad (2.45)$$

donde J_m es el momento de inercia del rotor y b es su coeficiente de fricción viscosa.

Analizando el circuito por la ley de voltajes de Kirchoff y tomando en cuenta el sentido de la tensión producida por la fuerza contraelectromotriz, se tiene que

$$L_a \frac{di_a}{dt} + R_a i_a = v_a - K_e \dot{\theta}_m \quad (2.46)$$

Aplicando la transformada de Laplace en (2.45) se tiene

$$J_m \Theta_m(s) s^2 + b \Theta_m(s) s = K_t I_a(s) \quad (2.47)$$

así mismo, transformando la Ecuación (2.46) se obtiene

$$L_a s I_a(s) + R_a I_a(s) = V_a(s) - K_e s \Theta_m(s) \quad (2.48)$$

De las Ecuaciones (2.47) y (2.48) se puede obtener la función de transferencia del motor, que es

$$\frac{\Theta_m(s)}{V_a(s)} = \frac{K_t}{s[(J_m s + b)(L_a s + R_a) + K_t K_e]} \quad (2.49)$$

En muchos casos, el efecto relativo de la inductancia es insignificante en comparación con el movimiento mecánico y puede despreciarse en la Ecuación (2.46). Si es así, podemos combinar las Ecuaciones (2.46) y (2.45) en una ecuación para obtener:

$$J_m \ddot{\theta}_m + \left(b + \frac{K_t K_e}{R_a} \right) \dot{\theta}_m = \frac{K_t}{R_a} v_a \quad (2.50)$$

Es claro que en este caso el efecto de la fuerza contra-electromotriz es indistinguible de la fricción, y la función de transferencia es

$$\frac{\Theta_m(s)}{V_a(s)} = \frac{\frac{K_t}{R_a}}{J_m s^2 + \left(b + \frac{K_t K_e}{R_a} s \right)} \quad (2.51)$$

$$= \frac{K}{s(\tau s + 1)} \quad (2.52)$$

donde se definió

$$K = \frac{K_t}{b R_a + K_t K_e}$$

$$\tau = \frac{R_a J_m}{b R_a + K_t K_e}$$

En ciertos casos, se requiere una función de transferencia que relacione la entrada del motor y la velocidad de salida ($\omega = \dot{\theta}_m$). En este caso la función de transferencia es

$$\frac{\Omega(s)}{V_a(s)} = s \frac{\Theta_m(s)}{V_a(s)} = \frac{K}{\tau s + 1} \quad (2.53)$$

Si se considera que el efecto de la inductancia del motor en el circuito es despreciable, de este modo $L_a \approx 0$, y la corriente se le puede describir como una funcion lineal de la tensión aplicada v_a y la velocidad del motor $\dot{\theta}_m$

$$i_a = \frac{v_a}{R_a} - \frac{K_e}{R_a} \dot{\theta}_m \quad (2.54)$$

$$\tau_m = \frac{K_t}{R_a} v_a - \frac{K_t K_e}{R_a} \dot{\theta}_m \quad (2.55)$$

Además, si la tensión aplicada es referenciada al valor de la señal de control u , entonces se tiene

$$v_a = K_s u \quad (2.56)$$

$$\tau_m = \frac{K_t K_s}{R_a} u - \frac{K_t K_e}{R_a} \dot{\theta}_m \quad (2.57)$$

La velociad angular $\dot{\theta}_m$ generada por el eje del motor en rotacion es igual a la de la llanta $\dot{\varphi}$ ya que se encuentran unidos en el mismo eje, sin embargo, existe el caso donde el péndulo se puede mover generando que el eje rote mientras las ruedas están en reposo, por lo que a su vez $\dot{\theta}_m = \dot{\varphi} - \dot{\theta}$, con lo que

$$\dot{\theta}_m = \frac{\dot{x}}{r} - \dot{\theta}$$

Sustituyendo el torque dado por la Ecuación (2.57) en (2.41) y (2.42), se tienen las expresiones

$$\begin{aligned} \ddot{x} &= -\frac{m_p l}{I} (\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta) + \frac{1}{r I} \left(\frac{K_t K_s}{R_a} u - \frac{K_t K_e}{R_a} \left(\frac{\dot{x}}{r} - \dot{\theta} \right) \right) - \frac{\left(\mu_s + \frac{\mu_e}{r^2} \right)}{I} \dot{x} + \frac{\mu_e}{r I} \dot{\theta} \\ \ddot{\theta} &= -\frac{m_p l}{J} \ddot{x} \cos \theta + \frac{m_p l g}{J} \sin \theta - \frac{1}{J} \left(\frac{K_t K_s}{R_a} u - \frac{K_t K_e}{R_a} \left(\frac{\dot{x}}{r} - \dot{\theta} \right) \right) + \frac{\mu_e}{J} \left(\frac{\dot{x}}{r} - \dot{\theta} \right) \end{aligned}$$

Reordenando y factorizando términos se pueden reescribir las ecuaciones que definen el sistema como

$$\ddot{x} = -\frac{m_p l}{I} (\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta) + \frac{K_t K_s}{R_a r I} u - \left(\frac{\mu_s + \frac{\mu_e}{r^2} + \frac{K_t K_e}{R_a r^2}}{I} \right) \dot{x} + \left(\frac{\mu_e}{r I} + \frac{K_t K_e}{R_a r I} \right) \dot{\theta} \quad (2.58)$$

$$\ddot{\theta} = -\frac{m_p l}{J} \ddot{x} \cos \theta + \frac{m_p l g}{J} \sin \theta - \frac{K_t K_s}{J R_a} u + \left(\frac{K_t K_e}{J R_a r} + \frac{\mu_e}{J r} \right) \dot{x} - \left(\frac{K_t K_e}{J R_a} + \frac{\mu_e}{J} \right) \dot{\theta} \quad (2.59)$$

2.1.5. Linealización del sistema

Se considera que la pérdida de energía de las ruedas con el suelo se puede despreciar $\mu_s = 0$. Lo cual se cumple cuando la energía disipada debido a la fricción en el acople entre carro y rueda haya sido mucho mayor a la disipada por la rueda-piso.

Retomando las Ecuaciones (2.58)-(2.59) que caracterizan la dinamica del sistema y reemplazando constantes, podemos escribir

$$\ddot{x} = -\frac{m_p l}{I}(\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta) + K_1 u - K_2 \dot{x} + K_6 \dot{\theta} \quad (2.60)$$

$$\ddot{\theta} = -\frac{m_p l}{J} \ddot{x} \cos \theta + K_3 \sin \theta - K_4 u - K_5 \dot{\theta} + K_7 \dot{x} \quad (2.61)$$

donde se utilizaron las constantes

$$\begin{aligned} K_1 &= \frac{K_t K_s}{R_a r I} & K_2 &= \frac{\mu_e}{r^2 I} + \frac{K_t K_e}{R_a r^2 I} & K_3 &= \frac{m_p l g}{J} \\ K_4 &= \frac{K_t K_s}{J R_a} & K_5 &= \frac{K_t K_e}{J R_a} + \frac{\mu_e}{J} & K_6 &= \frac{\mu_e}{r I} + \frac{K_t K_e}{R_a r I} \\ K_7 &= \frac{K_t K_e}{J R_a r} + \frac{\mu_e}{J r} \end{aligned}$$

Ahora, como se debe mantener el péndulo invertido en posición vertical, se puede suponer de que el ángulo θ se mantiene en valores pequeños, por lo que es válido aproximar $\sin \theta \approx \theta$ y $\cos \theta \approx 1$, además $\dot{\theta}^2 = 0$, entonces las ecuaciones se pueden reescribir de la siguiente manera

$$\ddot{x} = -\frac{m_p l}{I} \ddot{\theta} + K_1 u - K_2 \dot{x} + K_6 \dot{\theta} \quad (2.62)$$

$$\ddot{\theta} = -\frac{m_p l}{J} \ddot{x} + K_3 \theta - K_4 u - K_5 \dot{\theta} + K_7 \dot{x} \quad (2.63)$$

Para desacoplar dinámicas de \ddot{x} y $\ddot{\theta}$, se sustituye (2.63) en (2.62), de este modo se tiene

$$\begin{aligned} \ddot{x} &= -\frac{m_p l}{I} \overbrace{\left[-\frac{m_p l}{J} \ddot{x} + K_3 \theta - K_4 u - K_5 \dot{\theta} + K_7 \dot{x} \right]}^{\ddot{\theta}} + K_1 u - K_2 \dot{x} + K_6 \dot{\theta} \\ &= \frac{m_p^2 l^2}{I J} \ddot{x} + K_3 K_8 \theta - K_4 K_8 u - K_5 K_8 \dot{\theta} + K_7 K_8 \dot{x} + K_1 u - K_2 \dot{x} + K_6 \dot{\theta} \\ &= \frac{K_3 K_8}{K_9} \theta + \frac{K_1 - K_4 K_8}{K_9} u + \frac{K_6 - K_5 K_8}{K_9} \dot{\theta} + \frac{K_7 K_8 - K_2}{K_9} \dot{x} \end{aligned} \quad (2.64)$$

donde se definieron ahora las constantes

$$K_8 = -\frac{m_p l}{I} \quad (2.65)$$

$$K_9 = 1 - \frac{m_p^2 l^2}{I J} \quad (2.66)$$

Ahora se sustituye (2.62) en (2.63), para desacoplar las dinámicas de $\ddot{\theta}$, entonces se tiene

$$\begin{aligned} \ddot{\theta} &= -\frac{m_p l}{J} \overbrace{\left[-\frac{m_p l}{I} \ddot{\theta} + K_1 u - K_2 \dot{x} + K_6 \dot{\theta} \right]}^{\ddot{x}} + K_3 \theta - K_4 u - K_5 \dot{\theta} + K_7 \dot{x} \\ &= \frac{m_p^2 l^2}{I J} \ddot{\theta} + K_1 K_{10} u - K_2 K_{10} \dot{x} + K_6 K_{10} \dot{\theta} + K_3 \theta - K_4 u - K_5 \dot{\theta} + K_7 \dot{x} \\ &= \frac{K_1 K_{10} - K_4}{K_9} u + \frac{K_7 - K_2 K_{10}}{K_9} \dot{x} + \frac{K_6 K_{10} - K_5}{K_9} \dot{\theta} + \frac{K_3}{K_9} \theta \end{aligned} \quad (2.67)$$

donde se definió, por último, la constante

$$K_{10} = -\frac{m_p l}{J} \quad (2.68)$$

2.1.6. Representación en forma matricial de la ecuación de estado

La representación matricial de la ecuación lineal de estado, permite aplicar técnicas de álgebra lineal para la solución de la ecuación de estado y para el posterior diseño de controladores. En la Figura 2.3 se muestra una representación en diagrama de bloque de la ecuación de estado en forma matricial.

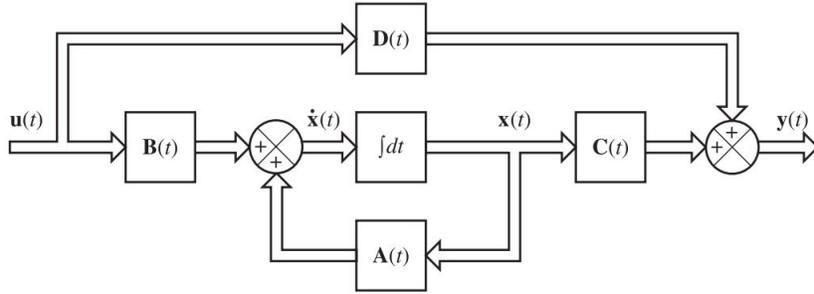


Figura 2.3: Diagrama a bloques del sistema de control lineal en tiempo continuo representado en el espacio de estado [14].

La ecuación de estado en forma matricial tiene la forma

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (2.69)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \quad (2.70)$$

Si se definen como variables de estado

$$x_1 = x, \quad x_2 = \dot{x} \quad (2.71)$$

$$x_3 = \theta, \quad x_4 = \dot{\theta} \quad (2.72)$$

entonces, las Ecuaciones (2.64)-(2.67) se pueden escribir en la forma (2.69) como

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{K_7 K_8 - K_2}{K_9} & \frac{K_3 K_8}{K_9} & \frac{K_6 - K_5 K_8}{K_9} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{K_7 - K_2 K_{10}}{K_9} & \frac{K_3}{K_9} & \frac{K_6 K_{10} - K_5}{K_9} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_1 - K_4 K_8}{K_9} \\ 0 \\ \frac{K_1 K_{10} - K_4}{K_9} \end{bmatrix} u \quad (2.73)$$

Y la ecuación de salida en la forma (2.70) es

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (2.74)$$

para obtener como salidas las variables x y θ . Entonces, las matrices \mathbf{A} , \mathbf{B} y \mathbf{C} son

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{K_7 K_8 - K_2}{K_9} & \frac{K_3 K_8}{K_9} & \frac{K_6 - K_5 K_8}{K_9} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{K_7 - K_2 K_{10}}{K_9} & \frac{K_3}{K_9} & \frac{K_6 K_{10} - K_5}{K_9} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ \frac{K_1 - K_4 K_8}{K_9} \\ 0 \\ \frac{K_1 K_{10} - K_4}{K_9} \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Capítulo 3

Estrategia de control

En el presente capítulo se abordará el método de diseño de una ley de control por realimentación de estado, el cual toma cada variable de estado para formar la señal de control. Además se verá el requisito principal que tiene que cumplir el sistema a controlar para aplicar dicha ley de control en la asignación de polos, también se abordará la existencia de una matriz de transformación cuando el sistema no está expresado en alguna de sus formas canónicas.

Siguiendo con la construcción de los criterios de diseño deseados, así como la construcción y simulación de un sistema de tipo regulatorio; es decir no sigue ninguna referencia de entrada.

Por último se tomará en cuenta el diseño de un servosistema tipo 1 para el péndulo blanco con respuesta a un escalón.

3.1. Control por realimentación de estado

La ley de control por realimentación de estados parte del modelo en variables de estado de la forma

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (3.1)$$

$$y = \mathbf{C}\mathbf{x} + Du \quad (3.2)$$

donde \mathbf{x} es el vector de estado (vector de dimensión n); y es la señal de salida (escalar); u denota la señal de control (escalar); \mathbf{A} es la matriz de coeficientes constantes de $n \times n$; \mathbf{B} es la matriz de entrada, de $n \times 1$; \mathbf{C} es la matriz de salida, de $1 \times n$; finalmente, D es la matriz de transmisión directa, de 1×1 .

El control por realimentación de estado se constituye por una señal de control de la forma

$$u = -\mathbf{K}\mathbf{x}$$

donde \mathbf{K} es la *matriz de ganancias de realimentación de estado*.

3.1.1. Asignación de polos y realimentación de estados

Para la ubicación de polos deseados en lazo cerrado hay que tener en consideración un requisito para poderlos asignar de manera arbitraria. La exigencia es que el sistema a controlar debe ser de estado completamente controlable.

En este apartado se probará que el sistema PISDR es de estado completamente controlable. Sea un sistema de control dado por:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (3.3)$$

$$y = \mathbf{C}\mathbf{x} + Du \quad (3.4)$$

El primer paso en el método de diseño en espacio de estado, consiste en encontrar una ley de control como realimentación de una combinación lineal de las variables de estado de la forma

$$u = -\mathbf{K}\mathbf{x} = - \begin{bmatrix} k_1 & k_2 & \cdots & k_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (3.5)$$

Con el fin de realizar simulación numérica, se supondrán valores para los parámetros presentes en el modelo dinámico. En la Tabla 3.1 se muestran los valores supuestos para simulación, donde las masas m_r, m_p fueron medidas de manera experimental del primer prototipo que se construyó. Para los cálculos de momento de inercias J_r y J_p se usaron las fórmulas como en [15] suponiendo cuerpos totalmente sólidos y rígidos para simplicidad del cálculo y para el cálculo de las constantes derivadas del motor se tomó las RPM del mismo por [16] además de medir experimentalmente su resistencia.

Tabla 3.1: Valores numéricos utilizados en las constantes del modelo del PISDR.

Parámetro	Valor
m_p	1.55 kg
m_r	0.025 kg
r	0.04 m
l	0.23 m
$J_r = \frac{m_r r^2}{2}$	2×10^{-5} kg m ²
$J_p = \frac{m_p l^2}{3}$	0.0273 kg m ²
R	5Ω
k_e	0.3342 V/rad/s
k_t	3.5958 Nm/A
k_s	0.01
g	9.8 m/s ²
μ_e	0.2

3.1.2. Matriz de controlabilidad

Como se había mencionado con anterioridad, se demostrará que para colocar los polos en lazo cerrado de manera arbitraria, se tiene que construir la matriz de controlabilidad y que ésta sea de estado totalmente controlable; es decir que sea de rango pleno, que todas sus columnas sean linealmente independientes. Para aplicar la ley de control hay que comprobar que el sistema sea de estado completamente controlable, por lo que se procede a construir la matriz de controlabilidad dada por

$$\mathbb{C} = [\mathbf{B} \quad \mathbf{AB} \quad \dots \quad \mathbf{A}^{n-1}\mathbf{B}] \quad (3.6)$$

Para este caso en particular la matriz de controlabilidad está dada por

$$\mathbb{C} = [\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A}^2\mathbf{B} \quad \mathbf{A}^3\mathbf{B}] \quad (3.7)$$

Se tiene que cumplir la siguiente condición

$$\text{rango}(\mathbb{C}) = n \quad (3.8)$$

Se calcula en MATLAB y se comprueba que es de rango 4, es decir de rango completo y que todas sus columnas son linealmente independientes. Con esto se concluye que se trata de un sistema controlable, condición suficiente para implementar la acción de control por realimentación de estado.

3.1.3. Ecuación característica

Ahora se procede a calcular la ecuación característica del sistema, la cual es independiente del tipo de representación del sistema (forma canónica controlable, canónica observable, canónica diagonal ó canónica de jordan).

Los valores propios de una matriz \mathbf{A} de $n \times n$ son las raíces de la ecuación característica, por lo que basta con resolver la siguiente ecuación

$$|\lambda\mathbf{I} - \mathbf{A}| = 0 \quad (3.9)$$

Los valores propios, λ_i , también se denominan raíces características. La ecuación característica es

$$\begin{aligned} |\lambda\mathbf{I} - \mathbf{A}| &= \begin{vmatrix} \lambda & -1 & 0 & 0 \\ 0 & \lambda + \frac{K_2 - K_7 K_8}{K_9} & -\frac{K_3 K_8}{K_9} & -\frac{K_6 - K_5 K_8}{K_9} \\ 0 & 0 & \lambda & -1 \\ 0 & -\frac{K_7 - K_2 K_{10}}{K_9} & -\frac{K_3}{K_9} & \lambda + \frac{K_5 - K_6 K_{10}}{K_9} \end{vmatrix} \\ &= \lambda^4 + \left(\frac{K_2 K_9 + K_5 K_9 - K_7 K_8 K_9 - K_6 K_9 K_{10}}{K_9^2} \right) \lambda^3 \\ &\quad + \left(-\frac{K_3 K_9 - K_2 K_5 + K_6 K_7 + K_2 K_5 K_8 K_{10} - K_6 K_7 K_8 K_{10}}{K_9^2} \right) \lambda^2 \\ &\quad + \left(-\frac{K_2 K_3 - K_2 K_3 K_8 K_{10}}{K_9^2} \right) \lambda \end{aligned}$$

La ecuación característica resultante al sustituir los valores numéricos de la Tabla 3.1 es

$$\lambda^4 + 146.1\lambda^3 - 258.7\lambda^2 - 2282.2\lambda = 0 \quad (3.10)$$

Mediante esta última ecuación, los valores propios del sistema en lazo abierto son

$$\lambda_1 = 0, \quad \lambda_2 = -147.7827 \quad \lambda_3 = 4.8379 \quad \lambda_4 = -3.1921 \quad (3.11)$$

Los valores obtenidos muestran que el sistema en lazo abierto describen un sistema inestable, debido al valor con raíz positiva, por lo que se utilizará un control por realimentación de estado para posicionar los valores propios del sistema en lazo cerrado en ciertos puntos deseados, que estabilicen el sistema y brinden una respuesta adecuada.

3.1.4. Criterios de diseño

Se sabe que la repuesta de sistemas de orden superior se puede aproximar a la repuesta de un sistema de segundo orden. Podemos tener un comportamiento de un sistema de segundo orden proponiendo dos polos dominantes para dicho sistema, y los otros dos restantes proponerlos alejados, hacia la izquierda de los polos dominantes y sobre el eje real, para que su aportación a la repuesta sea mínima. La forma estándar de un sistema de segundo orden se describe mediante

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (3.12)$$

donde ζ denota el factor de amortiguamiento y ω_n es la frecuencia natural no amortiguada. A partir de ζ y ω_n , se puede obtener la frecuencia natural amortiguada, ω_d y la atenuación, σ , mediante las expresiones siguientes

$$\begin{aligned} \omega_d &= \omega_n \sqrt{1 - \zeta^2} \\ \sigma &= \zeta \omega_n \end{aligned}$$

Los parámetros que se analizan en la repuesta de un sistema de segundo orden son los siguientes: M_p (sobreenlogación máxima), t_p (tiempo pico), t_s (tiempo de asentamiento), dados por:

$$M_p = e^{-\left(\frac{\zeta}{\sqrt{1-\zeta^2}}\pi\right)} \quad (3.13)$$

$$t_p = \frac{\pi}{\omega_d} \quad (3.14)$$

$$t_s = \frac{4}{\sigma} \quad (3.15)$$

En el proceso de diseño de un controlador por realimentación de estado, es necesario establecer la ubicación de todos los polos en lazo cerrado. En base a la información previa, se propone un tiempo de asentamiento y sobreelongación máxima deseadas, a partir de lo cual, en proceso inverso a las ecuaciones previas, se debe encontrar el factor de amortiguamiento, ζ y la frecuencia natural no amortiguada, ω_n que se sustituyen en la ecuación de la forma estándar, para obtener los polos dominantes de la Ecuación (3.12) y finalmente se proponen los polos no dominantes.

Tomamos como criterios de diseño un factor de amortiguamiento de $\zeta = 0.5$, lo cual genera un sobreimpulso de alrededor del 16 %, y además proponemos un tiempo de asentamiento de $t_s = 4$ seg (tomando el criterio del 2 %), de lo cual resulta una frecuencia natural de $\omega_n = 2$, por lo que los polos del polinomio característico deseado en lazo cerrado serán dos polos complejos conjugados, mientras que los otros dos polos no dominantes se colocarán alejados a la izquierda y sobre el eje real, para que su respuesta tenga menor magnitud y decaiga más rápido que la respuesta transitoria de los polos dominantes. Para el sistema numérico propuesto, los polos deseados en lazo cerrado se ubicarán en

$$\lambda_1 = -1 + j\sqrt{3}, \quad \lambda_2 = -1 - j\sqrt{3}, \quad \lambda_3 = -5, \quad \lambda_4 = -5$$

Con lo cual, el polinomio característico formado por dichos polos es

$$\lambda^4 + 12\lambda^3 + 49\lambda^2 + 90\lambda + 100 \quad (3.16)$$

3.1.5. Determinación de la matriz de ganancias de realimentación \mathbf{K}

Sea $\hat{\mathbf{K}}$ la matriz de ganancias de realimentación, la cual se obtiene de restar la ecuación característica deseada (3.16) a la ecuación característica en lazo abierto (3.10)

$$\lambda^4 + 12\lambda^3 + 49\lambda^2 + 90\lambda + 100 - (\lambda^4 + 146.1\lambda^3 - 258.7\lambda^2 - 2282.2\lambda)$$

de lo cual se obtiene

$$-134.1\lambda^3 + 307.7\lambda^2 + 2372.2\lambda + 100$$

los coeficientes del polinomio obtenido forman los elementos de $\hat{\mathbf{K}}$, es decir

$$\hat{\mathbf{K}} = \begin{bmatrix} -134.1 & 307.7 & 2372.2 & 100 \end{bmatrix} \quad (3.17)$$

Debido a que el sistema no está en forma canónica controlable, se debe multiplicar el vector $\hat{\mathbf{K}}$ por una matriz de similitud \mathbf{P} . Para encontrar la matriz de similitud basta con resolver una matriz que es dependiente de la matriz de controlabilidad. Se parte de la ecuación característica de la forma

$$\lambda^4 + \alpha_1\lambda^3 + \alpha_2\lambda^2 + \alpha_3\lambda + \alpha_4 = 0 \quad (3.18)$$

Considerando que el sistema es de estado completamente controlable, entonces existe una transformación de similitud $\bar{x} = \mathbf{P}x$ tal que

$$\mathbf{Q} = \mathbf{P}^{-1} = \left(\mathbb{C} \begin{bmatrix} 1 & \alpha_1 & \alpha_2 & \alpha_3 \\ 0 & 1 & \alpha_1 & \alpha_2 \\ 0 & 0 & 1 & \alpha_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right)^{-1} \quad (3.19)$$

que, al sustituir los valores numéricos propuestos da

$$\mathbf{Q} = \begin{bmatrix} 0 & 4.6040 & 2.6813 & -0.1812 \\ 0 & 0.3223 & 0.1877 & 0.1651 \\ 0 & 0.0226 & 0.1909 & 0.0116 \\ -0.3052 & 0.0229 & 0.0378 & 0.0118 \end{bmatrix} \quad (3.20)$$

De esta forma, el vector de ganancias de realimentación de estado se calcula mediante

$$\mathbf{K} = \hat{\mathbf{K}}\mathbf{Q}$$

que, sustituyendo los valores empleados se expresa como

$$\mathbf{K} = \begin{bmatrix} -134.1 & 307.7 & 2372.2 & 100 \end{bmatrix} \begin{bmatrix} 0 & 4.6040 & 2.6813 & -0.1812 \\ 0 & 0.3223 & 0.1877 & 0.1651 \\ 0 & 0.0226 & 0.1909 & 0.0116 \\ -0.3052 & 0.0229 & 0.0378 & 0.0118 \end{bmatrix}$$

Por lo tanto, se tiene

$$\mathbf{K} = \begin{bmatrix} -30.5227 & -681.5807 & 27.1866 & 10.0376 \end{bmatrix} \quad (3.21)$$

3.2. Diseño de un servosistema tipo 1

El diseño de un servosistema nos resultará útil cuando se desea que este sistema siga una referencia distinta del origen. El principio básico del diseño de un servosistema de tipo 1 es insertar un integrador en el camino directo entre el comparador de error y la planta, como se muestra en la Figura 3.1.

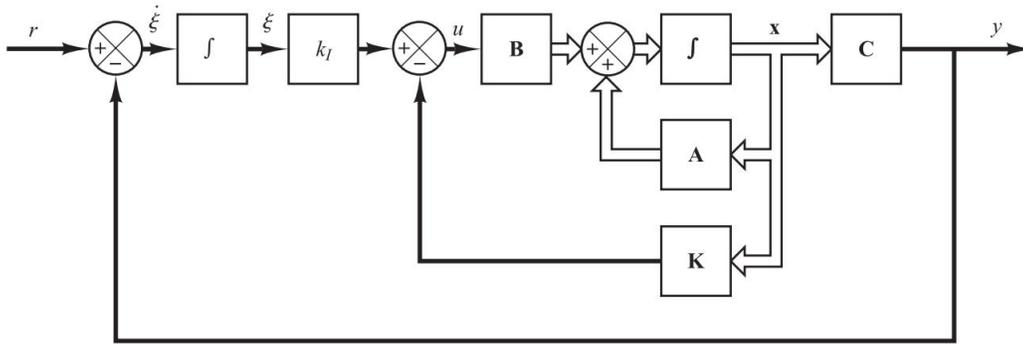


Figura 3.1: Sistema de control para un péndulo invertido (servosistema tipo 1 cuando la planta no tiene integrador) [14].

Teniendo en cuenta las Ecuaciones (2.71)-(2.72) y (2.73), y considerando la posición angular del péndulo, θ o x_3 como salida del sistema, se obtienen las ecuaciones para el sistema del modo siguiente

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (3.22)$$

$$y = \mathbf{C}\mathbf{x} \quad (3.23)$$

$$u = -\mathbf{K}\mathbf{x} + k_I\xi \quad (3.24)$$

$$\dot{\xi} = r - y = r - \mathbf{C}\mathbf{x} \quad (3.25)$$

Para el servo sistema de tipo 1, se debe definir la ecuación de estado del error.

$$\dot{\mathbf{e}} = \hat{\mathbf{A}}\mathbf{e} + \hat{\mathbf{B}}u_e \quad (3.26)$$

en donde las matrices $\hat{\mathbf{A}}$ y $\hat{\mathbf{B}}$ se definen mediante

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & 0 \\ -\mathbf{C} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{K_7 K_8 - K_2}{K_9} & \frac{K_3 K_8}{K_9} & \frac{K_6 - K_5 K_8}{K_9} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & \frac{K_7 - K_2 K_{10}}{K_9} & \frac{K_3}{K_9} & \frac{K_6 K_{10} - K_5}{K_9} & 0 \\ -1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.27)$$

y

$$\hat{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{K_1 - K_4 K_8}{K_9} \\ 0 \\ \frac{K_1 K_{10} - K_4}{K_9} \\ 0 \end{bmatrix} \quad (3.28)$$

Una vez definida la dinámica del error, la señal de control está dada por

$$u_e = -\hat{\mathbf{K}}\mathbf{e} \quad (3.29)$$

en la cual, la matriz de realimentación de estado es

$$-\hat{\mathbf{K}} = [\mathbf{K} \quad \vdots \quad -k_I] = [k_1 \quad k_2 \quad k_3 \quad k_4 \quad \vdots \quad -k_I] \quad (3.30)$$

Se seleccionan los polos en lazo cerrado en $s = \mu_i$ ($i = 1, 2, 3, 4, 5$), conforme los criterios vistos anteriormente como

$$\mu_1 = -1 + j\sqrt{3}, \quad \mu_2 = -1 - j\sqrt{3}, \quad \mu_3 = \mu_4 = \mu_5 = -5$$

Usando MATLAB se determinará la matriz de ganancia de realimentación del estado. Antes de continuar, se debe examinar el rango de la matriz \mathbf{P} , la cual se define de la siguiente manera

$$\mathbf{P} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ -\mathbf{C} & 0 \end{bmatrix} \quad (3.31)$$

La matriz \mathbf{P} , en este caso, está dada por

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{K_7 K_8 - K_2}{K_9} & \frac{K_3 K_8}{K_9} & \frac{K_6 - K_5 K_8}{K_9} & \frac{K_1 - K_4 K_8}{K_9} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & \frac{K_7 - K_2 K_{10}}{K_9} & \frac{K_3}{K_9} & \frac{K_6 K_{10} - K_5}{K_9} & \frac{K_1 K_{10} - K_4}{K_9} \\ 0 & -1 & 0 & 0 & 0 \end{bmatrix} \quad (3.32)$$

Se calcula el rango de la matriz \mathbf{P} y se comprueba que es de rango 5; es decir de rango completo, por lo tanto, el sistema es de estado completamente controlable y es posible la asignación arbitraria de los polos.

El siguiente paso es calcular la matriz de ganancias de realimentación de estado. MATLAB proporciona una solución sencilla para este problema, se puede usar una función llamada **acker** que se basa en el método de Ackermann. Dicha función recibe como parámetros de entrada las matrices $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$ y los polos en lazo cerrado deseados. Al aplicar esta función para el sistema que se está diseñando se obtiene

$$\mathbf{K} = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix} = \begin{bmatrix} -7.3971 & -43.7748 & -114.5602 & -31.5662 \end{bmatrix} \quad (3.33)$$

y

$$k_I = -36.9857 \quad (3.34)$$

3.2.1. Características de la respuesta a un escalón unitario del sistema diseñado

Una vez que se han determinado la matriz de ganancia de realimentación \mathbf{K} y la constante de ganancia integral k_I , la respuesta a un escalón en la posición del carro se puede obtener resolviendo la siguiente ecuación:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} \mathbf{A} - \mathbf{BK} & \mathbf{B}k_I \\ -\mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \xi \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \quad (3.35)$$

La salida del sistema es $x_1(t)$ o bien

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \xi \end{bmatrix} \quad (3.36)$$

Capítulo 4

Construcción del prototipo

En este capítulo se dan aspectos técnicos sobre los elementos utilizados para la construcción del sistema PISDR. Como se ha mencionado anteriormente, un PISDR necesita una plataforma con dos ruedas separadas cierta distancia a lo largo del eje de giro. Para accionar las ruedas se utilizaron motores de corriente continua, que son fáciles de modelar y controlar. Adicionalmente estos motores cuentan con encoders de cuadratura para medir su posición. Sobre la plataforma de base donde se montan las ruedas, se añaden, a modo de pisos, más plataformas rectangulares, de las mismas dimensiones que la base, para montar sobre ellas los componentes electrónicos, la batería, etcétera, y se añaden estas plataformas hacia arriba, para que el centro de masa del PISDR esté por encima de la base del carro, con lo cual se logra el comportamiento de péndulo.

4.1. Componentes utilizados

A continuación se describen los principales componentes utilizados.

4.1.1. Motorreductores

Como actuador para cada rueda, se seleccionó un motorreductor de 12 V, voltaje nominal, con una relación de vueltas de 34:1. Alcanza una velocidad de 290 RPM, y un torque 8.5 kg-cm. El motor es de la marca pololu. En la Figura 4.1 se aprecia dicho motor. El motor cuenta con un encoder de efecto Hall, lo cual produce dos señales de cuadratura. El encoder cuenta con 48 CPR, por lo que usándolo en modo de cuadratura, se tienen 1632.67 cuentas por revolución.



Figura 4.1: Motor pololu relación 34:1.

4.1.2. Ruedas

Como ruedas se utilizaron un par de llantas de la marca pololu, de 90×10 mm, en color negro, que tienen un peso de 23 g, cada una. La Figura 4.2 muestra un par de ruedas como las utilizadas en el prototipo.



Figura 4.2: Ruedas marca pololu utilizadas en el PISDR.

4.1.3. Sensor inercial

El módulo de medición inercial MPU6050, como el de la Figura 4.3, tiene un giroscopio de tres ejes con el cual se puede medir velocidad angular y un acelerómetro también de 3 ejes con el que se analiza la medición de ángulo de los componentes x, y, z , así como su aceleración. Los sensores de giroscopio y acelerómetro son de tecnología MEMS. Es un circuito compacto, de $2 \times 1.6 \times 0.1$ mm. En Arduino se cuenta con librerías para leer este módulo y su comunicación es por vía I2C.



Figura 4.3: Módulo acelerómetro y giroscopio MPU6050.

4.1.4. Tarjetas de control

Como tarjetas de control se utilizaron dos tarjetas Arduino nano y una tarjeta Arduino MEGA. Los Arduino nano se utilizaron para medir la posición y velocidad de cada rueda y enviar esa información al Arduino MEGA. Se utilizó un Arduino para detectar la posición en cada rueda debido a que de esta manera las interrupciones temporales no perderán pulsos de alguna de las ruedas y por otra parte así se libera al Arduino MEGA de tiempo de cálculo, utilizándolo mejor para dedicarlo al controlador. La comunicación entre los Arduinos se realizó por protocolo I2C. En la Figura 4.4 se muestra una imagen de las tarjetas Arduino MEGA 2560 y Arduino nano.

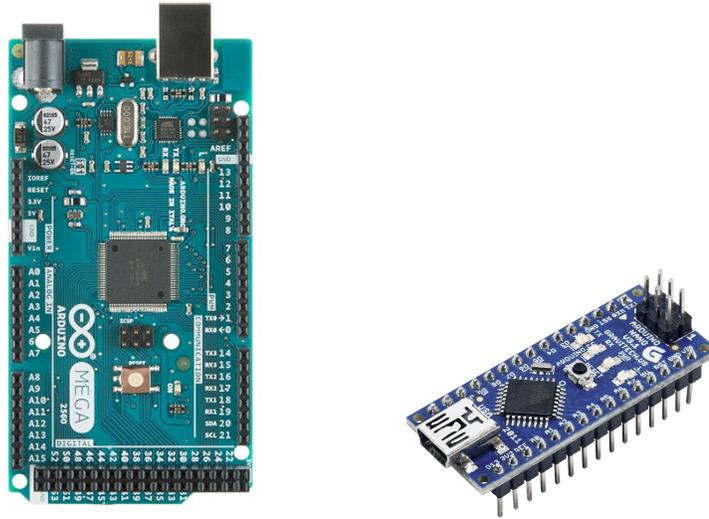


Figura 4.4: Tarjetas Arduino MEGA 2560 y Arduino nano.

4.1.5. Controlador de potencia para motores

El controlador para motores es un circuito que tiene como base el circuito integrado L298N, el cual es un doble puente H. Este circuito es capaz de manejar niveles altos de voltaje y de corriente. La alimentación de voltaje puede variar entre 6 a 48 V, y entrega a la salida 5 V y hasta 2 A en cada canal. Por lo tanto es ideal para manejar motores de corriente continua en ambos sentidos. La Figura 4.5 muestra una imagen del módulo para control de motores.



Figura 4.5: controlador para motores con circuito integrado L298N.

4.1.6. Batería

Como fuente de alimentación se usó una batería de Litio-Polímero de 4 celdas, entrega un voltaje de 14.8 V y tiene una capacidad de carga de 2200 mAh. La imagen de una batería como la utilizada se muestra en la Figura 4.6

4.1.7. Cuerpo del PISDR

Para la construcción del cuerpo del péndulo se utilizaron plataformas de madera e triplay de 1/2" con dimensiones 15 × 20 cm, conformado por tres niveles, teniendo una separación de 10 cm entre cada nivel. La separación entre niveles se realizó mediante tornillos sin-fin de 1/4", ver Figura 4.7, y tuercas hexagonales de 1/4".



Figura 4.6: Batería LiPo.



Figura 4.7: Tornillo sin fin.

4.2. Prototipo construido

En el prototipo como finalmente fue construido se muestra en la Figura 4.8. El primer nivel se colocó la batería lipo encargada de suministrar energía a los motores.

El segundo nivel se utilizó para montar microcontroladores del tipo Arduino nano en la modalidad de esclavos para la medición de velocidad de cada rueda y el módulo MPU6050, todos ellos alimentados por una batería de 9 V y conectados mediante comunicación I2C al Arduino MEGA.

Finalmente, en el tercer nivel contiene a un Arduino Mega cuya función es de maestro y encargado de procesar la información obtenida de posición y velocidad de las ruedas enviada por los Arduino nano, además de procesar las lecturas del módulo MP6050. También se encuentran montados 2 módulos puente H para controlar los motores con consignas enviadas por el Arduino MEGA. Los programas (*sketch*) utilizados en los Arduinos se muestran en el Apéndice.

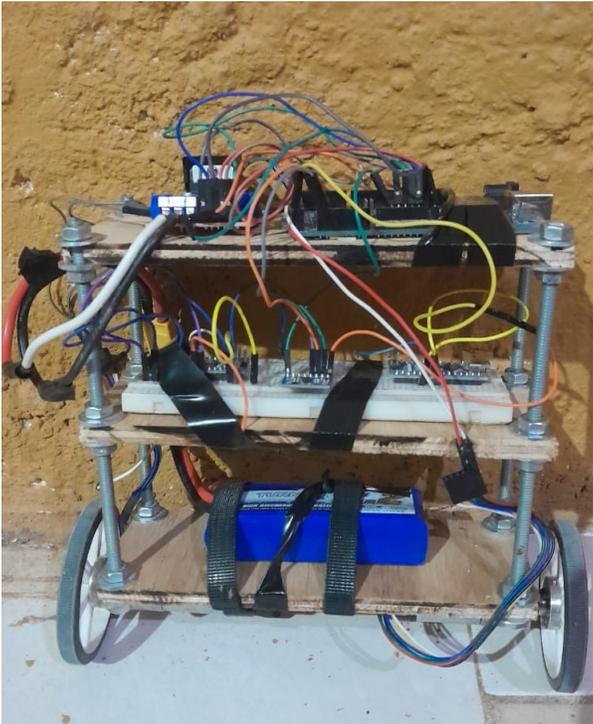


Figura 4.8: Prototipo PISDR construido.

Capítulo 5

Resultados

En este capítulo se presentan los resultados de las simulaciones hechas para el PISDR considerando primero los resultados de simulación para los controladores lineales de realimentación de estado, el de regulación y el de seguimiento, los cuales toman en cuenta todas las variables de estado y brindan un comportamiento deseado de la respuesta en lazo cerrado. Posteriormente se presenta la simulación de un doble controlador PID para controlar el ángulo del péndulo y su posición, utilizado directamente sobre el modelo no lineal del sistema. Las simulaciones fueron realizadas en MATLAB 2020, utilizando en algunos programas archivos .m y en otros Simulink.

5.1. Simulación del control por realimentación de estado

El primer controlador diseñado es un sistema definido como “regulador”, ya que el sistema no presenta alguna referencia y trata de mantener la estabilidad del sistema rechazando cualquier tipo de perturbación que se presenten a lo largo de la ejecución del sistema. En la Sección 3.1.4, se propusieron ciertos criterios de diseño del sistema de control, en donde se buscaba que mediante realimentación de estado, el nuevo sistema en lazo cerrado responda con un tiempo de asentamiento de alrededor de 4 segundos y un sobreimpulso de alrededor del 16 %. Con esos criterios se desarrolló el análisis para obtener la matriz de ganancias de realimentación de estado, \mathbf{K} , de la Ecuación (3.21). Haciendo la simulación con el Código de MATLAB 6.1 que aparece en el Apéndice, se obtienen las gráficas de la Figura 5.1. En dicha figura, del resultado de la variable x_1 , que representa la posición del carro, se puede observar que el tiempo de asentamiento efectivamente es de aproximadamente 4 segundos y el sobretiro es de alrededor de 16 %, como se propuso en el diseño del controlador. Para dicha simulación se tomó como condición inicial $x_1 = 0$, $x_2 = 0$, $x_3 = 45^\circ$ y $x_4 = 0$.

5.1.1. Simulación entorno de realidad virtual

Para hacer más realista la simulación se creó un ambiente de realidad virtual mediante V-Realm Builder, el cual es un software donde se pueden construir entornos virtuales. Para poder ejecutar la simulación del mundo virtual se reprodujo en Simulink el mismo sistema lineal con realimentación de estado. Se realiza un bloque de MatLabFunction en Simulink donde se colocan las ecuaciones diferenciales dadas por la Ecuación (2.73) y variables de la Tabla 3.1 que definen la planta del sistema. En el Programa 6.3 se muestra el código implementado. La Figura 5.2 muestra el modelo

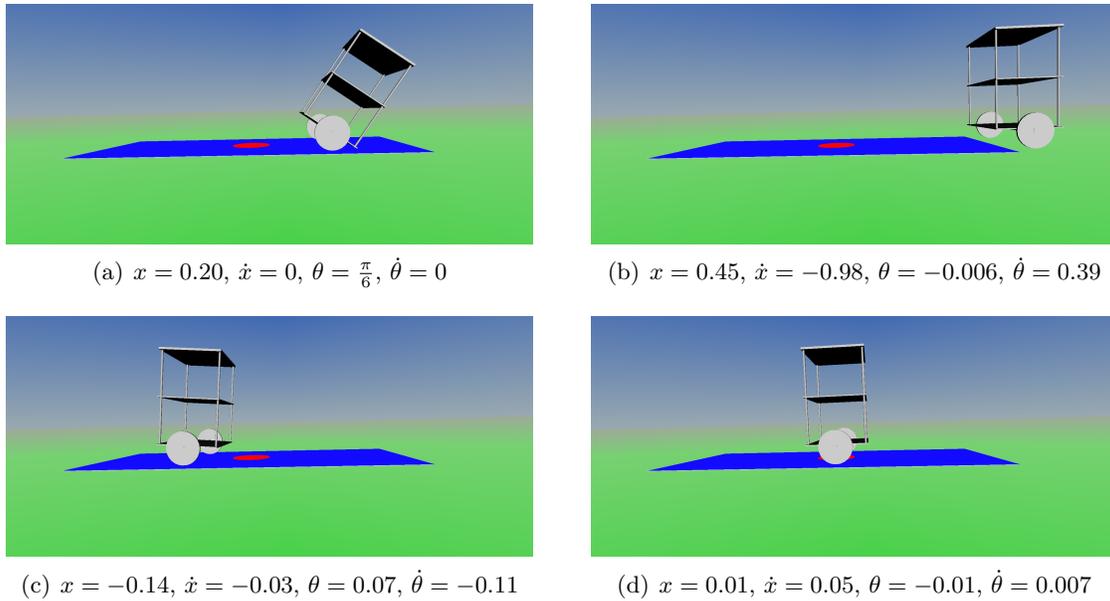


Figura 5.3: Dinámica del sistema tipo regulatorio en $t = 0, 1.43, 2.55, 4$ segundos, respectivamente.

5.2. Simulación de servosistema tipo 1

El objetivo en este caso es mantener la posición del péndulo en la vertical hacia arriba, mientras la posición del carro es controlada a una posición que puede ser distinta de cero. Se añadió un integrador al sistema para que el error de posición en estado estable sea cero. Se seleccionaron los polos en lazo cerrado conforme a los criterios descritos anteriormente, por lo que se espera un tiempo de asentamiento de alrededor de 4 segundos y un sobreimpulso de alrededor del 16%. Con esos criterios se desarrolló el análisis para obtener la matriz de ganancias de realimentación de estado, $\hat{\mathbf{K}}$, donde se obtuvieron los valores dados en (3.33) y (3.34).

La Figura 5.4 presenta la solución del sistema de realimentación de estado con integrador. Observe que la salida $y(t) = x_1(t)$ tiene aproximadamente el 15% de sobreelongación y el tiempo de asentamiento es aproximadamente de 4 seg., como se había propuesto. El código de MATLAB para realizar esta simulación se muestra en el Programa 6.2.

5.2.1. Simulación entorno de realidad virtual

El servo-sistema para regular a un punto distinto del origen se muestra en la Figura 5.5. El bloque del modelo dinámico del PISDR es una función de MATLAB que se muestra en el Programa 6.3. Algunas escenas de la animación simulada se muestran en la Figura 5.6, donde se presentan imágenes del mundo virtual al correr la simulación en tiempos $t = 0, 0.51, 1.70$ y 5 segundos.

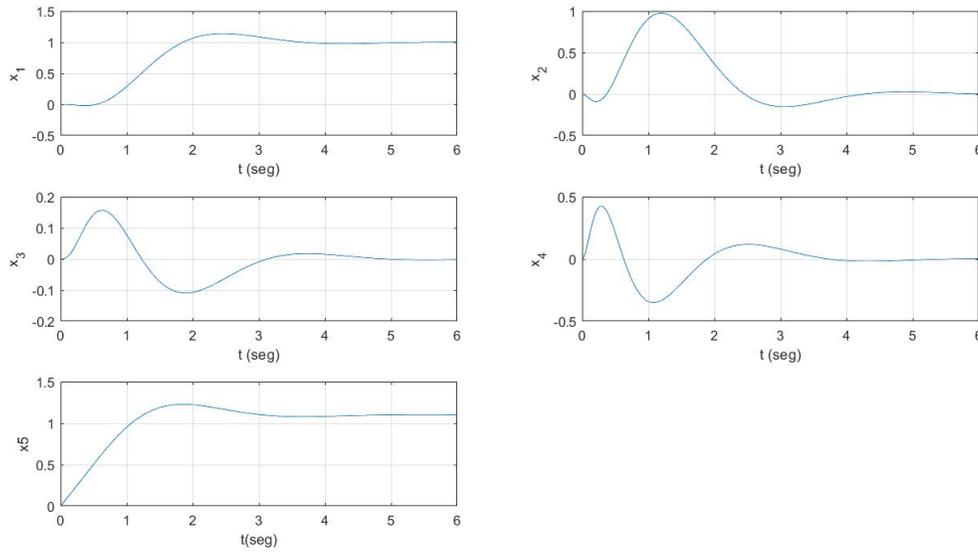


Figura 5.4: Respuesta a un escalon unitario de cada estado con respecto al tiempo $x_1 = x$, $x_2 = \dot{x}$, $x_3 = \theta$, $x_4 = \dot{\theta}$, $x_5 = \xi$.

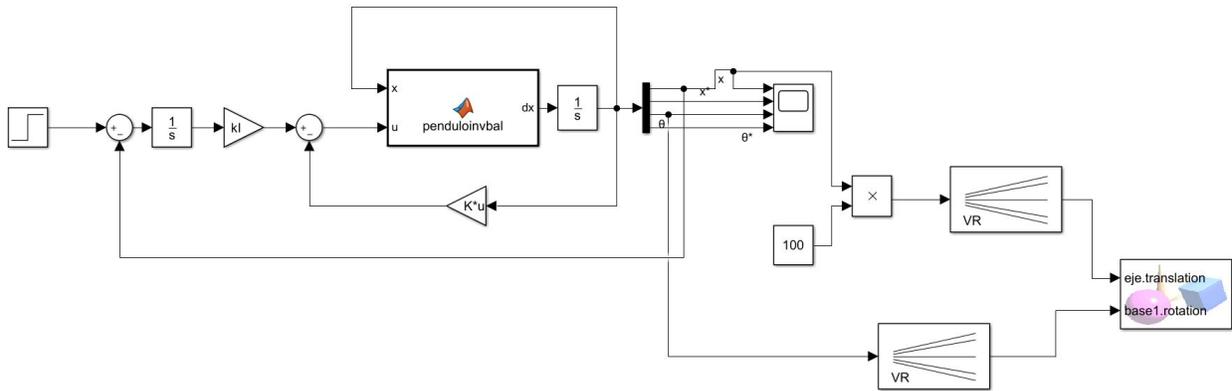


Figura 5.5: Servo-sistema tipo 1 en Simulink, para desplegar en realidad virtual.

5.3. Simulación por Controlador PID

En esta sección se muestra una simulación de un doble lazo de control PID, que permite estabilizar el ángulo θ y la posición x del carro. El doble lazo PID quiere decir que se toman dos salidas, por un lado la posición del carro y se realimenta un controlador PID de posición, y por otro lado se toma la salida de ángulo del péndulo y ésta también entra a un lazo de control PID. Finalmente, ambas señales de control se suman para formar la entrada de control al sistema. En la Figura 5.7 se muestra el diagrama de bloques hecho en Simulink para el control por el doble lazo PID. Como condiciones iniciales se tienen los valores $\theta = \frac{\pi}{4}$, $x = 0.5$, $\dot{\theta} = 0$ y $\dot{x} = 0$ y se establecen como referencias $\theta = 0$ y $x = 1$. La sintonización del controlador se hizo a prueba y error y no fue posible mejorar los resultados obtenidos con los controladores por realimentación de estado, como se puede observar en la Figura 5.8.

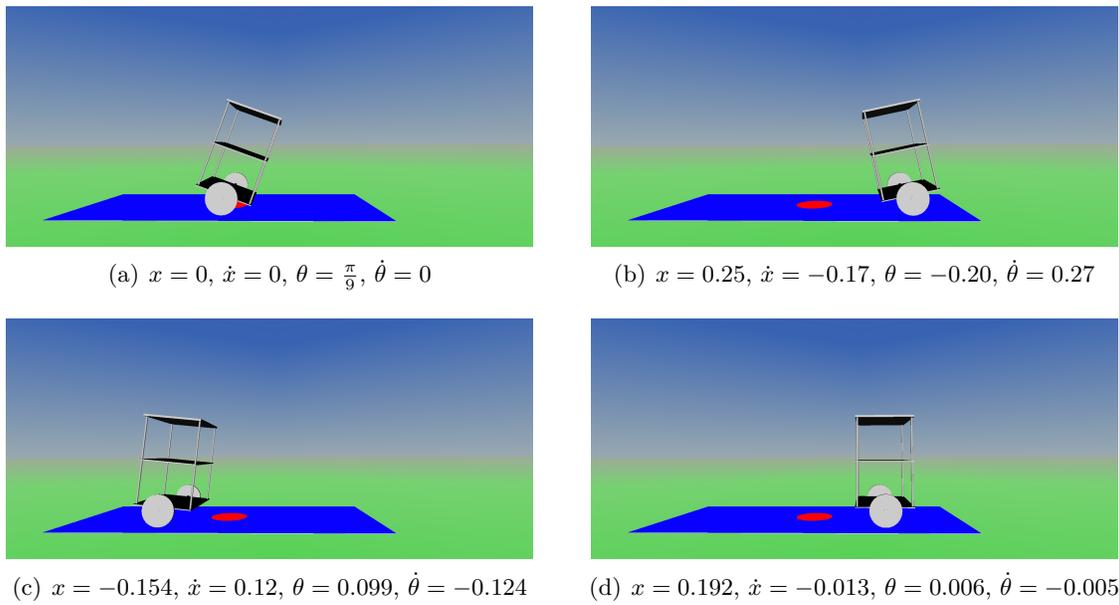


Figura 5.6: Dinámica del Servo-sistema tipo 1 en $t = 0, 0.51, 1.70, 5$ segundos, respectivamente.

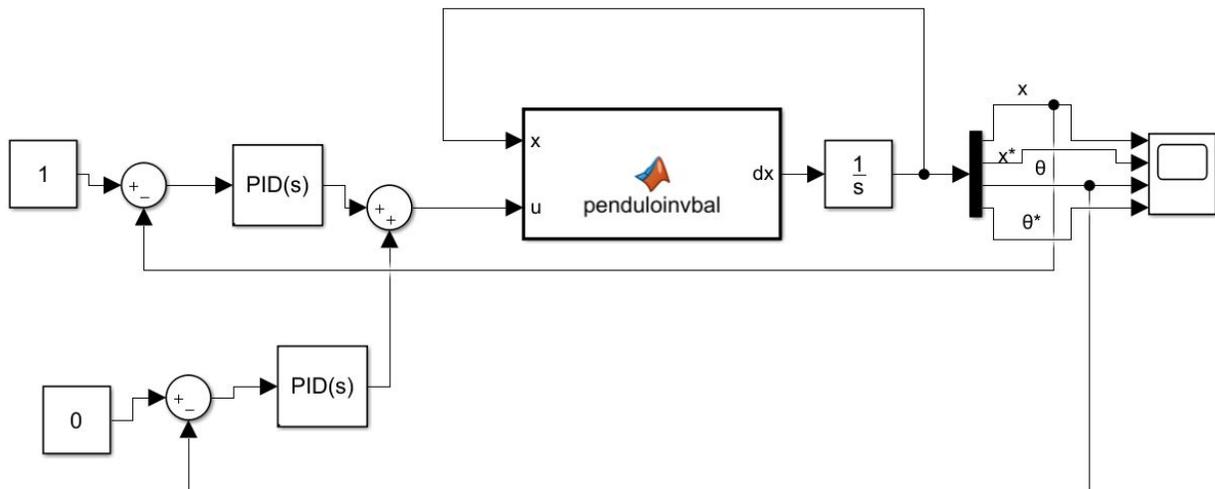
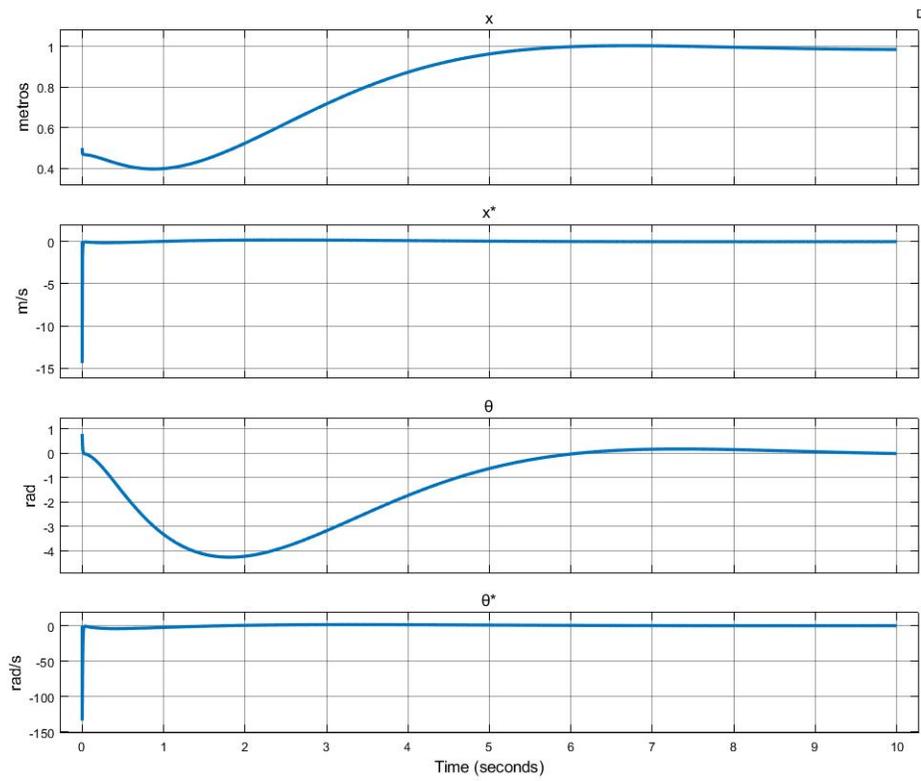


Figura 5.7: Diagrama a bloques de PISDR.

Figura 5.8: Comportamiento de θ , $\dot{\theta}$, x , \dot{x} .

Capítulo 6

Conclusiones

En esta tesis se plantea el diseño y control de un sistema PISDR. Para poder diseñar un buen controlador, se debe obtener un buen modelo matemático que exprese las principales fuerzas que influyen el movimiento de dicho sistema. Se desarrolló un modelo no lineal considerando las principales fuerzas que actúan en la dinámica del PISDR y se consideró incluso en el modelo una señal de control referenciada al voltaje de entrada de cada motor. Así mismo, se hicieron consideraciones sobre un ángulo θ pequeño, de manera que alrededor de $\theta = 0$ se puede aproximar el modelo del sistema como un modelo lineal. Con el modelo lineal es posible diseñar controladores por realimentación de estado siguiendo unos cuantos pasos.

Los controladores por realimentación de estado diseñados cumplen los requisitos de diseño, en donde se estableció un sobreimpulso y un tiempo de asentamiento esperados, y tras diseñar la matriz de realimentación de estado, en cada caso, las simulaciones muestran que la salida cumple con las consignas propuestas, que además de cumplir con las referencias deseadas, esto se hace también con parámetros deseados en la respuesta transitoria.

Para dar mayor realismo a las simulaciones realizadas, se desarrolló una interfaz virtual, en la que se puede apreciar de manera visual el desarrollo de las simulaciones. Con la animación virtual se tiene mejor idea del desplazamiento sobre todo del ángulo del péndulo.

Finalmente, se desarrolló físicamente un prototipo de PISDR, en el cual se ha ido probando su funcionamiento en cada componente, tanto de sensores como de actuadores, y por falta de tiempo no se realizó un buen resultado experimental.

Bibliografía

- [1] S. Mori, H. Nishihara, and K. Furuta, “Control of unstable mechanical system control of pendulum,” *International Journal of Control*, vol. 23, no. 5, pp. 673 – 692, 1976.
- [2] K. Furuta, M. Yamakita, and S. Kobayashi, “Swing-up control of inverted pendulum using pseudo-state feedback,” vol. 206, no. 4, pp. 263–269, 1992.
- [3] T. Kobayashi, M. Iwase, S. Suzuki, and K. Furuta, “Swinging-up and balancing control of double furuta pendulum by using state-dependent riccati equation,” *IFAC Proceedings Volumes*, vol. 37, no. 14, pp. 337–341, Sep. 2004, 3rd IFAC Symposium on Mechatronic Systems 2004, Sydney, Australia, 6-8 September, 2004.
- [4] F. Grasser, A. D’Arrigo, S. Colombi, and A. C. Rufer, “Joe: a mobile, inverted pendulum,” *IEEE Trans. Ind. Electron*, vol. 49, pp. 107–114, 2002.
- [5] O. R. Chi, “Balancing a two-wheeled autonomous robot,” Thesis, School of Mechanical Engineering, the University of Western Australia, Nov. 2003.
- [6] E. R. E. Y. J. O. R. M. Ontiveros Rodriguez, “Diseño y control de umbra - un péndulo invertido sobre dos ruedas,” *Memorias del XVI Congreso Latinoamericano de Control Automático*, Oct. 2014.
- [7] O. O. G. Frías, “Estabilización del péndulo invertido sobre dos ruedas mediante el método de lyapunov,” *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 10, no. 1, pp. 30–36, 01 2013.
- [8] F. Zhang, C. Zhu, and X. Hou, “Analysis of swinging method of straight inverted pendulum.” Kunming, China: IEEE, 2021, pp. 4719–4723.
- [9] X. Jiarui, Z. Chunguang, and W. Ruixin, “Single stage inverted pendulum controller design.” Changsha, China: IEEE, 2021, pp. 441–444.
- [10] L. Messikh, E.-H. Guechi, F. Bourahala, and S. Blazic, “Stabilization of the cart-inverted-pendulum system using trivial state-feedback to output-feedback control conversion,” *Automa-*

tika, vol. 63, no. 4, pp. 640–655, 12 2022.

- [11] K. Ogata, *Dinámica de Sistemas*. Prentice Hall, 1987.
- [12] T. S. C, *Introducción a la Mecánica de Lagrange y Hamilton*, 1st ed. La Universidad Del Zulia, 2013.
- [13] J. D. P. y. A. E. N. Gene F. Franklin, *Feedback Control of Dynamic Systems*, 8th ed., Pearson, Ed., 2020.
- [14] K. Ogata, *Ingeniería de control moderna*, 5th ed., Pearson, Ed. Prentice Hall, 2010.
- [15] J. J. E. R. . P. J. C. Ferdinand P. Beer, *MECÁNICA VECTORIAL PARA INGENIEROS DINÁMICA.*, 9th ed., S. d. C. Mcgraw-Hill/Interamericana Editores, Ed. MC Graw Hill., 2010.
- [16] P. R. . Electronics, “Motorreductor pololu 34:1.” [Online]. Available: <https://www.pololu.com/product/4844>

Apéndice

6.1. Código para diseñar sistema de realimentación de estado (sistema tipo regulador)

El Código de MATLAB 6.1 muestra los pasos para el diseño de un sistema regulador de realimentación de estado.

MATLAB Script 6.1: Realimentación de estado (sistema tipo regulador)

```
clc
clear all
close all
kt=3.5958; ks=0.01; mr=0.025; mp=1.550; r=0.04;
Jr=(mr*r^2)/2; l=0.23; Jp=mp*l^2/3;
R=5; mue=0.01; ke=0.3342;
g=9.8;

I=(mr+mp+Jr/r^2);
J=Jp+mp*l^2;
k1=kt*ks/(I*r*R);
k2=(mue/(r^2*I))+kt*ke/(I*R);
k3=(mp*l*g)/J;
k4=(kt*ks)/(J*R);
k5=(mue/J)+kt*ke/(J*R);
k6=mue/(r*I)-(kt*ke)/(I*r*R);
k7=(kt*ke*r/(J*R))+mue/(J*r);
k8=-(mp*l)/I;
k9=1-mp^2*l^2/(I*J);
k10=-mp*l/J;
%Sistema en lazo abierto NO esta en la forma FCC
A=[0 1 0 0;
    0 ((k7*k8)-k2)/k9 k3*k8/k9 (k6-(k5*k8))/k9;
    0 0 0 1;
    0 (k7-(k2*k10))/k9 k3/k9 ((k6*k10)-k5)/k9];
B=[0; (k1-(k4*k8))/k9; 0; ((k1*k10)-k4)/k9];
C=[1 0 0 0];
% Se calcula el sistema en Espacio de Estado
sys=ss(A,B,C,0)

% matriz de controlabilidad y su rango se calculan con
Co = ctrb(sys)
rank(Co);
```

```

% Los coeficientes de la ecuación característica
E_Ca = poly(eig(sys))

% Ubicación de los polos deseados
J_o = [-1+i*sqrt(3) -1-i*sqrt(3) -5 -5];
Ps = poly(J_o)
Ed = roots(Ps);

% Matriz de ganancias K_bar
k_bar=Ps(2:end)-E_Ca(2:end)

% Cálculo de matriz de similitud
Ci = toeplitz(E_Ca(1:end-1));
Ci = triu(Ci);
Q=(Co*Ci)^(-1)

% Cálculo de matriz de ganancias de realimentación de estado
k_1 = k_bar*Q
k_2 = place(A,B,Ed)

% El sistema en lazo cerrado es
sys = ss(A-B*k_2, eye(4), eye(4), eye(4));
t = 0:0.02:6;
% x = solución del sistema de realimentación de estado
x = initial(sys, [0;0;(45*pi)/180;0], t);
x1 = [1 0 0 0]*x';
x2 = [0 1 0 0]*x';
x3 = [0 1 0 0]*x';
x4 = [0 0 0 1]*x';
subplot(4,1,1); plot(t, x1), grid
title('Respuesta a condicion inicial')
ylabel('Variable x_1')
subplot(4,1,2); plot(t, x2), grid
ylabel('Variable x_2')
subplot(4,1,3); plot(t, x3), grid
xlabel('t (seg)')
ylabel('Variable x_3')
subplot(4,1,4); plot(t, x4), grid
xlabel('t (seg)')
ylabel('Variable x_4')

```

6.2. Código para diseño de servosistema tipo 1

En esta sección se muestra el código de MATLAB 6.2 utilizado para diseñar un servosistema de tipo 1, donde se agrega un integrador al control por realimentación de estado y se logra controlar la posición a un punto distinto del origen.

MATLAB Script 6.2: Diseño de servosistema tipo 1

```

clear all
% definición de constantes
kt=3.5958; ks=0.01; mr=0.025; mp=1.550; r=0.04;
Jr=(mr*r^2)/2; l=0.23; Jp=mp*l^2/3; R=5; mue=0; ke=0.3342;
g=9.8;
I=(mr+mp+Jr/r^2);
J=Jp+mp*l^2;

```

```

k1=kt*ks/(I*r*R);
k2=(mue/(r^2*I))+(kt*ke/(I*R*r^2));
k3=(mp*l*g)/J;
k4=(kt*ks)/(J*R);
k5=(mue/J)+((kt*ke)/(J*R));
k6=(mue/(r*I))+((kt*ke)/(I*r*R));
k7=((kt*ke)/(J*R*r))+ (mue/(J*r));
k8=-(mp*l)/I;
k9=1-(mp^2*l^2)/(I*J);
k10=-mp*l/J;

% Matrices del sistema en lazo abierto
A=[0 1 0 0;
    0 ((k7*k8)-k2)/k9 k3*k8/k9 (k6-(k5*k8))/k9;
    0 0 0 1;
    0 (k7-(k2*k10))/k9 k3/k9 ((k6*k10)-k5)/k9];
B=[0; (k1-(k4*k8))/k9; 0; ((k1*k10)-k4)/k9];
C=[1 0 0 0];
D=[0];

% sistema aumentado con integrador
Ahat = [A zeros(4,1); -C 0];
Bhat = [B; 0];

% Ubicación de los polos deseados
J_o = [-1+j*sqrt(3) -1-j*sqrt(3) -5 -5 -5];

% Matriz de realimentación de estado
Khat = acker(Ahat, Bhat, J_o);

% El sistema de control en lazo cerrado es
AA = [A-B*K B*kI; -C 0];
BB = [0; 0; 0; 0; 1];
CC = [C 0];
DD = [0];
t = 0:0.02:6;

% la solución se obtiene mediante
[y, x, t] = step(AA, BB, CC, DD, 1, t);
x1 = [1 0 0 0 0]*x';
x2 = [0 1 0 0 0]*x';
x3 = [0 0 1 0 0]*x';
x4 = [0 0 0 1 0]*x';
x5 = [0 0 0 0 1]*x';
subplot(3,2,1); plot(t, x1); grid
xlabel('t (seg)'); ylabel('x_1')
subplot(3,2,2); plot(t, x2); grid
xlabel('t (seg)'); ylabel('x_2')
subplot(3,2,3); plot(t, x3); grid
xlabel('t (seg)'); ylabel('x_3')
subplot(3,2,4); plot(t, x4); grid
xlabel('t (seg)'); ylabel('x_4')
subplot(3,2,5); plot(t, x5); grid
xlabel('t (seg)'); ylabel('x_5')

```

6.3. Función de MATLAB del modelo del sistema para Simulink

El código de la función de MATLAB que se utilizó para simular la dinámica del PISDR se muestra en el Programa 6.3.

MATLAB Script 6.3: Función de MATLAB del modelo del sistema para Simulink

```
function dx = penduloinvbal(x,u)

%definicion de parametros
mp =1.55; mr=0.025; l=0.23; g=9.8;r=0.04; Jp=(mp*l^2)/3; Jr=(mr*r^2)/2;
mue=0.2; R=5; kt=3.5958; ke=0.3342; ks=0.01;

I=mr+mp+(Jr/r^2);
J=Jp+mp*l^2;
k1=kt*ks/(I*r*R);
k2=(mue/(r^2*I))+(kt*ke/(I*R*r^2));
k3=(mp*l*g)/J;
k4=(kt*ks)/(J*R);
k5=(mue/J)+((kt*ke)/(J*R));
k6=(mue/(r*I))+((kt*ke)/(I*r*R));
k7=((kt*ke)/(J*R*r))+ (mue/(J*r));
k8=-(mp*l)/I;
k9=1-((mp^2*l^2)/(I*J));
k10=-(mp*l)/J;

% ESTADOS
x1 = x(1);
x2 = x(2);
x3 = x(3);
x4 = x(4);

%donde
%x1=dx , x2=dx , x3=\theta ,x4=d\theta

dx1=x2;
dx2=(( (k7*k8)-k2)/k9)*x2+(k3*k8/k9)*x3+((k6-k5*k8)/k9)*x4+((k1-k4*k8)/k9)*u;
dx3=x4;
dx4=(( (k7-k2*k10)/k9)*x2+(k3/k9)*x3+((k6*k10-k5)/k9)*x4+((k1*10-k4)/k9)*u;

dx=[dx1;dx2;dx3;dx4];
```

6.4. Códigos de Arduino utilizados en la plataforma experimental

6.4.1. Código Arduino esclavo

El Programa de Arduino 6.4 se muestra el código implementado en los Arduino nano, que se encarga de leer los encoder mediante interrupciones externas, calcular la posición de la rueda y enviarla al Arduino MEGA cuando llegue una solicitud de envío de información.

Arduino Sketch 6.4: Programa Arduinos nano

```
#include <Wire.h>
byte bytes[4];

// define encoder structure
typedef struct {
```

```

int pinA;
int pinB;
long pos;
int del;
} Encoder;
volatile Encoder Enc = { 0, 0, 0, 0 };

/* Interrupt Service Routine: change on pin A for Encoder 0 */
void isrPinA() {

    // lee valor de pinB del encoder
    int drB = digitalRead(Enc.pinB);

    // posiblemente espera antes de leer el pin A, luego lo lee
    delayMicroseconds(Enc.del);
    int drA = digitalRead(Enc.pinA);

    /* this updates the counter */
    if (drA == HIGH) { /* low->high on A? */
        if (drB == LOW) { /* check pin B */
            Enc.pos++; /* going clockwise: increment */
        } else {
            Enc.pos--; /* going counterclockwise: decrement */
        }
    } else { /* must be high to low on A */
        if (drB == HIGH) { /* check pin B */
            Enc.pos++; /* going clockwise: increment */
        } else {
            Enc.pos--; /* going counterclockwise: decrement */
        }
    } /* end counter update */
} /* end ISR pin A Encoder 0 */

/* Interrupt Service Routine: change on pin B for Encoder 0 */
void isrPinB() {
    /* read pin A right away */
    int drA = digitalRead(Enc.pinA);
    /* possibly wait before reading pin B, then read it */
    delayMicroseconds(Enc.del);
    int drB = digitalRead(Enc.pinB);
    /* this updates the counter */
    if (drB == HIGH) { /* low->high on B? */
        if (drA == HIGH) { /* check pin A */
            Enc.pos++; /* going clockwise: increment */
        } else {
            Enc.pos--; /* going counterclockwise: decrement */
        }
    } else { /* must be high to low on B */
        if (drA == LOW) { /* check pin A */
            Enc.pos++; /* going clockwise: increment */
        } else {
            Enc.pos--; /* going counterclockwise: decrement */
        }
    } /* end counter update */
} /* end ISR pin B Encoder 0 */

void setup() {
    Serial.begin(9600); // configurar monitor serie a 9600

```

```

Wire.begin(23); // unirse al bus i2c con la direccion #23
Wire.onRequest(eventoSolicitud); // registrar evento de solicitud de datos
//Wire.onReceive(eventoRecepcion); // registrar evento de recepcion de datos

Enc.pinA = 2;
Enc.pinB = 3;
Enc.del = 10; // un retardo entre 0 y 100 us para evitar rebotes

// activa como entradas los pines del encoder
pinMode(Enc.pinA, INPUT);
pinMode(Enc.pinB, INPUT);
// activa las resistencias internas de pullup
digitalWrite(Enc.pinA, HIGH);
digitalWrite(Enc.pinB, HIGH);
// enlaza interrupción
attachInterrupt(digitalPinToInterrupt(Enc.pinA), isrPinA, CHANGE);
attachInterrupt(digitalPinToInterrupt(Enc.pinB), isrPinB, CHANGE);
}

void loop() {
}

void eventoSolicitud() {

bytes[0] = (byte)((Enc.pos & 0xFF000000) >> 24);
bytes[1] = (byte)((Enc.pos & 0x00FF0000) >> 16);
bytes[2] = (byte)((Enc.pos & 0x0000FF00) >> 8);
bytes[3] = (byte)((Enc.pos & 0x000000FF));

for (int i = 0; i < 4; i++) {
Wire.write(bytes[i]);
}
}

```

6.4.2. Código Arduino maestro

El Programa de Arduino 6.5 es el programa principal para control del PISDR. El programa se encarga de leer los sensores por vía I2C, calcular la velocidad angular y aplicar un control PID de ángulo del péndulo.

Arduino Sketch 6.5: Programa Arduino MEGA

```

// Librerias I2C para controlar el mpu6050
// la libreria MPU6050.h necesita I2Cdev.h, I2Cdev.h necesita Wire.h
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"

// La direccion del MPU6050 puede ser 0x68 o 0x69, dependiendo
// del estado de AD0. Si no se especifica, 0x68 estara implicito
int ENA = 5; // pin de conexion de llanta derecha (pin de PWM)
int IN1=8;
int IN2=9;

```

```

int ENB = 7;          // pin de conexion de llanta derecha   (pin de PWM)
int IN3=10;
int IN4=11;

MPU6050 sensor(0x69);

// Valores RAW (sin procesar) del acelerometro y giroscopio en los ejes x,y,z
int ax, ay, az;
int gx, gy, gz;

long tiempo_prev;
float dt;
float ang_x, ang_y;
float ang_x_prev, ang_y_prev;
float omega_y;

unsigned long tActualR = 0; // variables para definicion del tiempo de interrupcion
                          // y calculo de la velocidad motor derecho
unsigned long tAnteriorR = 0;
unsigned long deltatR = 0;
long EncPosActualR = 0;
long EncPosAnteriorR = 0;
long deltaPosR = 0;
double VelocidadR = 0;

float resolucion = 1632.67;

float rpmR = 0.0;

char valor[8] = {};
int dato = 0;
char PID[]={};
char pwm[]={};
// PID constants *****
double kp = 80;
double ki = 0;
double kd = 2;

unsigned long tiempoanterior;
unsigned long tiempoactual;

double tiempostranscurrido;
double error;
double errorpasado;
double input, output, Setpoint=-5;
double erroracumulado, rateError;
//*****

void setup() {
  Wire.begin();          // unirse al bus i2c como master
  Serial.begin(9600);    // configurar monitor serie a 9600
  sensor.initialize();  //Iniciando el sensor

  if (sensor.testConnection()) Serial.println("Sensor iniciado correctamente");
  else Serial.println("Error al iniciar el sensor");
}

void loop() {
  // Leer las aceleraciones y velocidades angulares

```

```

sensor.getAcceleration(&ax, &ay, &az);
sensor.getRotation(&gx, &gy, &gz);

dt = (millis()-tiempo_prev)/1000.0;
tiempo_prev=millis();

//Calcular los Ángulos con acelerometro
float accel_ang_x=atan(ay/sqrt(pow(ax,2) + pow(az,2)))*(180.0/3.14);
float accel_ang_y=atan(-ax/sqrt(pow(ay,2) + pow(az,2)))*(180.0/3.14);

Serial.println(accel_ang_y);
//Calcular los angulos con acelerometro *****
//float accel_ang_x=atan(ax/sqrt(pow(ay,2) + pow(az,2)))*(180.0/3.14);
// float accel_ang_y=atan(-ax/sqrt(pow(ay,2) + pow(az,2)))*(180.0/3.14);

//Calcular angulo de rotacion con giroscopio y filtro complemento
ang_x = 0.98*(ang_x_prev+(gx/131)*dt) + 0.02*accel_ang_x;
//ang_y = 0.98*(ang_y_prev+(gy/131)*dt) + 0.02*accel_ang_y;

ang_x_prev=ang_x;
//ang_y_prev=ang_y;

//Serial.println(ang_x);
omega_y=gy/131;
//Mostrar los angulos separadas por un [tab]

//*****
// Serial.print("Inclinacion en X: ");
// Serial.print(accel_ang_x);
//*****
// Serial.print("Rotacion en X: ");
//Serial.print(ang_x);
// Serial.print("Rotacion en Y: ");
// Serial.print(ang_y);

//*****
//Serial.print("velocidad angular en Y: ");
// Serial.println(omega_y);
//**

// delay(500);
//-----//
//-----//
Wire.requestFrom(23, 8); // Solicitar 1 byte del esclavo #23

int i = 0; //counter for each bite as it arrives
while (Wire.available()) {
  valor[i] = Wire.read();
  i = i + 1;
}
// Serial.println(valor); // imprimir el caracter en el monitor serial
String myString = String(valor);
//Serial.println(myString);
//double Encpos=myString.toDouble();
long Encpos = myString.toInt();
//Serial.println(Encpos);

tActualR = micros(); // se asigna el tiempo de ejecucion a el
muestreo actual

```

```

//Serial.println(tActualR);
deltatR = tActualR - tAnteriorR; // diferencia tiempos de interrupciones de ticks
del motor
//Serial.println("delta T");
//Serial.println(deltatR);
// se actualiza el tiempo de interrupcion
anterior
//EncPosActualR=Encpos;
tAnteriorR = tActualR;

deltaPosR = Encpos - EncPosAnteriorR;

EncPosAnteriorR = Encpos;

rpmR = (double) (deltaPosR*1000000*60) / (deltatR);
//Serial.print(Encpos);
//Serial.print(" ");
//Serial.print(deltatR);
//Serial.print(" ");
//Serial.println(rpmR);
VelocidadR=4*rpmR*(3.1416/30); // velocidad lineal cm/s
//Serial.println(VelocidadR);

/* Para por cada vuelta del rotor antes de la caja de engranes, se producen 48
pulsos (o pasos) del encoder. El motor tiene una relacion de reduccion en su
caja de engranes de 34:1. Por tanto, se tienen 1632.67 ticks o pulsos del
encoder por cada revolucion del rotor despues de la caja de engranes. Por lo que
360/1632.67 = 0.22049...*/

//*****

input = accel_ang_y; //angulo de inclinacion del eje x
output = calculoPID(input);
delay(100);

//analogWrite(3, output); //control the motor based on PID value
//*****
}

double calculoPID(double inp){
tiempoactual = millis(); //se obtiene el tiempo actual

tiempotrancurrido = (double)(tiempoactual - tiempoanterior); //calcular el tiempo
trancurrido desde el calculo anterior

error = Setpoint - inp; // determina el error
erroracumulado += error*tiempotrancurrido; //la integral del error es el error
acumulativo en el tiempo
rateError = (error - errorpasado)/tiempotrancurrido; // compute derivative

double out = kp*error + ki*erroracumulado + kd*rateError; //PID output

//*****
//Serial.println(out);
if (out > 255)
{
out = 255; // Se pone un limite en el valor maximo que puede alcanzar el valor
de pid
}
}

```

```

else if (out < -255)
{
  out = -255; //Se pone un limite en el valor minimo que puede alcanzar el valor
  de pid
}
// Serial.println(out);
if (out < 0)
{
  //digitalWrite(4,LOW);
  //digitalWrite(2,HIGH);// el carro se mueve hacia atrás

  int salida=map(out,-255,0,255,0);// se mapea PID para que el pulso pwm se maneje
  dentro del rango

  analogWrite(5, salida); //control the motor based on PID value
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);

  analogWrite(7, salida); //control the motor based on PID value
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
  //

  // dtostrf ( salida , 1 , 0 , pwm );
  //Serial.println(salida);
  // Wire.beginTransmission(23); // Comunicarse con esclavo #23
  //Wire.write(pwm);
  //Wire.endTransmission();
}
else
{
  // digitalWrite(4,HIGH);//el carro se mueve hacia enfrente
  // digitalWrite(2,LOW);

  int salida=map(out,0,255,0,255);

  analogWrite(5, salida); //control the motor based on PID value
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);

  analogWrite(7, salida); //control the motor based on PID value
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,HIGH);

  // dtostrf ( salida , 1 , 0 , pwm );
  // Serial.println(pwm);
  //Wire.beginTransmission(23); // Comunicarse con esclavo #23
  //Wire.write(pwm);
  //Wire.endTransmission();
}

//*****
//Serial.println(out);
//dtostrf ( out , 1 , 2 , PID );

//Serial.println(PID);
//Serial.println("PWM");
//
// Wire.beginTransmission(23); // Comunicarse con esclavo #23

```

```
//Wire.write(PID);  
//Wire.endTransmission();  
//*****  
  
errorpasado = error; //actualizamos el error  
tiempoanterior = tiempoactual; //remember current time  
  
return out; //have function return the PID output  
}
```