



UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES

TRABAJO DE TITULACIÓN

Propuesta tecnológica, previo a la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

**“DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR LINEAL
CUADRÁTICO APLICADO A LA ESTABILIZACIÓN DE UN SISTEMA
ELECTROMECAÁNICO DENOMINADO ROBOT BALANCÍN”**

AUTOR

Lídice Mariana Pillasagua Ayala

TUTOR

Ing. Carlos Alberto Saldaña Enderica MSc.

LA LIBERTAD - ECUADOR

2024

AGRADECIMIENTO

Quiero expresar mi más sincero agradecimiento a mis padres, Félix Pillasagua e Ingrid Ayala, por su inquebrantable apoyo a lo largo de este viaje académico. Su amor, aliento y sacrificios han sido la fuerza impulsora detrás de cada logro que he alcanzado. Gracias por creer en mí y por ser mi mayor inspiración.

A mi tutor de tesis, Ing. Carlos Saldaña, le agradezco sinceramente por su paciencia, orientación y dedicación. Su experiencia y sabiduría han sido fundamentales para el desarrollo de este trabajo. Aprecio enormemente el tiempo y esfuerzo que ha invertido en mi formación académica.

A mi querida compañera de estudio, Yuri Torres, le agradezco por su colaboración, apoyo y amistad a lo largo de este desafiante trayecto. Sus aportes han enriquecido mi experiencia académica y personal, y estoy agradecida por la camaradería compartida durante estos años.

Asimismo, agradezco a la Universidad Península de Santa Elena por proporcionarme un entorno educativo excepcional. Los años de estudio han sido una travesía enriquecedora, llena de aprendizajes significativos que me han preparado para enfrentar los desafíos futuros.

Lídice Pillasagua

DEDICATORIA

A mis distinguidos padres, Félix Pillasagua e Ingrid Ayala,

Quiero agradecerles su incondicional apoyo a lo largo de mi trayectoria académica. Su constante respaldo ha sido fundamental en el logro de este trabajo de investigación.

La guía, sabiduría y valores que me han transmitido han sido una fuente inagotable de inspiración. Agradezco sinceramente la dedicación y el sacrificio que han invertido en mi educación, permitiéndome alcanzar este hito significativo en mi vida académica.

Esta tesis está dedicada a ustedes, con la gratitud más profunda por ser los pilares sólidos sobre los cuales he construido mi educación y aspiraciones profesionales.

Lídice Pillasagua

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación denominado: **“DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR LINEAL CUADRÁTICO APLICADO A LA ESTABILIZACIÓN DE UN SISTEMA ELECTROMECAÁNICO DENOMINADO ROBOT BALANCÍN”**, presentado por la estudiante **Lídice Mariana Pillasagua Ayala**, de la carrera de electrónica y telecomunicaciones de la Universidad Península de Santa Elena, me permito declarar que luego de haber orientado, estudiado y revisado, la apruebo en todas sus partes y autorizo al estudiante para que inicie los trámites legales correspondientes.

Se adjunta como anexos los reportes generados por la plataforma compilatio que reporto 2% de plagio del documento final de tesis.

La libertad, 22 de mayo del 2024



Firmado electrónicamente por:
CARLOS ALBERTO
SALDANA ENDERICA

ATT: Ing. Carlos Saldaña

TRIBUNAL DE GRADO

Ing. Washington Torres Guin Mgt.

Decano de la Facultad

Ing. José Sánchez Aquino, Mgt.

Director de la Carrera



Firmado electrónicamente por:
JUNIOR RAFAEL
FIGUEROA OLMEDO

Ing. Junior Figueroa Olmedo Mgt.

Docente del Área



Firmado electrónicamente por:
CARLOS ALBERTO
SALDANA ENDERICA

Ing. Carlos Saldaña Enderica Mgt.

Docente Tutor

Ab. María Rivera González Mgt.

Secretaria General.

DECLARACION DEL AUTOR

Lídice Mariana Pillasagua Ayala, egresada de la Universidad Estatal Península de Santa Elena, facultad Sistemas y Telecomunicaciones, de la carrera Electrónica y Telecomunicaciones, con libertad y en plena voluntad declaro que la responsabilidad del contenido presente Trabajo de Titulación me corresponde exclusivamente y la propiedad intelectual de la misma le pertenece a la Universidad Estatal Península de Santa Elena.



Lídice Mariana Pillasagua Ayala

RESUMEN

En este trabajo se exploran los antecedentes y el contexto actual de los sistemas de control, destacando la importancia del regulador lineal cuadrático (LQR) en el control de sistemas no lineales como los robots balancín. Se menciona la relevancia histórica de las ecuaciones desarrolladas en los años 60 para el diseño del LQR. El proyecto se divide en tres etapas centradas en implementar un LQR en un sistema electromecánico, específicamente un robot balancín. Los objetivos incluyen el diseño del modelo matemático del sistema, la implementación del controlador LQR, y la evaluación de su desempeño mediante simulaciones y pruebas físicas. Se justifica la investigación en base a la necesidad de aplicar controladores modernos en sistemas complejos para mejorar su estabilidad y eficiencia. Se mencionan diversos estudios comparativos que demuestran la superioridad del LQR frente a otros controladores clásicos en términos de respuesta y estabilidad. El alcance del proyecto se delimita desde la construcción del modelo matemático hasta la implementación del controlador en el robot balancín físico, utilizando herramientas de simulación y comunicación UART para el intercambio de datos. Se detalla la metodología de investigación, que abarca investigación descriptiva, documental y aplicada, orientada a explorar y aplicar los principios del control LQR en sistemas electromecánicos no lineales.

Se presenta el marco teórico necesario para entender y aplicar el control LQR en sistemas electromecánicos como el robot balancín. Se explican los fundamentos del control de realimentación de estados y se introduce la teoría del regulador lineal cuadrático (LQR), destacando su aplicación en sistemas lineales y no lineales. Se abordan conceptos clave como las matrices de ponderación Q y R , fundamentales para el diseño del LQR y la optimización del control. Se describe el modelo matemático del robot balancín, basado en el modelo del péndulo invertido, y se detallan los componentes electrónicos esenciales del sistema, como el microcontrolador, el puente H y el giroscopio. Se explica cómo se realiza la linealización del modelo no lineal del robot balancín y la identificación de parámetros para el diseño del controlador. Además, se analizan estudios previos y aplicaciones del control LQR en diversas áreas como la robótica, vehículos autónomos y sistemas industriales, destacando la eficacia del LQR para mejorar la estabilidad y respuesta de estos sistemas. Se mencionan módulos didácticos y aplicaciones prácticas del robot balancín en la investigación científica y el desarrollo tecnológico.

También se detalla el proceso de diseño e implementación del controlador lineal cuadrático (LQR) para estabilizar el sistema electromecánico del robot balancín. Se inicia con la formulación del modelo matemático del sistema en espacio de estado, utilizando ecuaciones que describen la dinámica del robot bajo la influencia de variables como la posición, velocidad y aceleración. Se destacan los pasos para la linealización de las ecuaciones no lineales y la obtención de las matrices dinámicas del sistema. Posteriormente, se discute la aplicación del controlador LQR, que se basa en la minimización de una función de costo cuadrática que pondera el error de estado y la magnitud de la señal de control. Se explican los criterios para la selección de las matrices Q y R del LQR, las cuales influyen en la respuesta del sistema y su estabilidad. Además, se describe la metodología para la implementación del controlador en un entorno de simulación utilizando el software Simulink, lo que permite validar el comportamiento teórico del controlador antes de la implementación física.

Se presentan las evaluaciones numéricas y experimentales del controlador LQR en el robot balancín. Se detallan los parámetros de evaluación utilizados, como el tiempo de estabilización, máximo sobreimpulso, señal de control máxima y mínima, error de estado estacionario y ángulo de inclinación máximo. Se realizan simulaciones en Simulink con diferentes condiciones iniciales y señales de referencia, obteniendo datos específicos que muestran el desempeño del controlador. Se comparan estos resultados simulados con pruebas físicas realizadas en el sistema real del robot balancín, destacando similitudes y diferencias clave entre ambos entornos. Finalmente, se presentan conclusiones basadas en los resultados obtenidos, resaltando la efectividad del controlador LQR para mantener la estabilidad del sistema bajo diversas condiciones. También se ofrecen recomendaciones para ajustar y mejorar los parámetros del controlador, así como explorar estrategias alternativas para manejar perturbaciones externas.

ABSTRACT

This work explores the background and current context of control systems, highlighting the importance of the linear quadratic regulator (LQR) in the control of non-linear systems such as seesaw robots. The historical relevance of the equations developed in the 1960s for the design of the LQR is mentioned. The project is divided into three stages focused on implementing an LQR in an electromechanical system, specifically a seesaw robot. The objectives include the design of the mathematical model of the system, the implementation of the LQR controller, and the evaluation of its performance through simulations and physical tests. The research is justified based on the need to apply modern controllers in complex systems to improve their stability and efficiency. Various comparative studies are mentioned that demonstrate the superiority of the LQR compared to other classic controllers in terms of response and stability. The scope of the project is delimited from the construction of the mathematical model to the implementation of the controller in the physical seesaw robot, using simulation tools and UART communication for data exchange. The research methodology is detailed, which encompasses descriptive, documentary and applied research, aimed at exploring and applying the principles of LQR control in nonlinear electromechanical systems.

The theoretical framework necessary to understand and apply LQR control in electromechanical systems such as the seesaw robot is presented. The fundamentals of state feedback control are explained and the theory of the linear quadratic regulator (LQR) is introduced, highlighting its application in linear and non-linear systems. Key concepts such as the Q and R weighting matrices, fundamental for LQR design and control optimization, are addressed. The mathematical model of the rocking robot is described, based on the inverted pendulum model, and the essential electronic components of the system are detailed, such as the microcontroller, the H-bridge and the gyroscope. It explains how the linearization of the nonlinear model of the rocking robot is carried out and the identification of parameters for the design of the controller. In addition, previous studies and applications of LQR control in various areas such as robotics, autonomous vehicles and industrial systems are analyzed, highlighting the effectiveness of LQR to improve the stability and response of these systems. Didactic modules and practical applications of the seesaw robot in scientific research and technological development are mentioned.

The design and implementation process of the linear quadratic (LQR) controller to stabilize the electromechanical system of the seesaw robot is also detailed. It begins with the formulation of the mathematical model of the system in state space, using equations that describe the dynamics of the robot under the influence of variables such as position, speed and acceleration. The steps for the linearization of the nonlinear equations and obtaining the dynamic matrices of the system are highlighted. Subsequently, the application of the LQR controller is discussed, which is based on the minimization of a quadratic cost function that weights the state error and the magnitude of the control signal. The criteria for selecting the Q and R matrices of the LQR are explained, which influence the response of the system and its stability. Additionally, the methodology for implementing the controller in a simulation environment using the Simulink software is described, which allows validating the theoretical behavior of the controller before physical implementation.

Numerical and experimental evaluations of the LQR controller on the seesaw robot are presented. The evaluation parameters used are detailed, such as stabilization time, maximum overshoot, maximum and minimum control signal, steady state error and maximum tilt angle. Simulations are carried out in Simulink with different initial conditions and reference signals, obtaining specific data that shows the performance of the controller. These simulated results are compared with physical tests performed on the real seesaw robot system, highlighting key similarities and differences between both environments. Finally, conclusions are presented based on the results obtained, highlighting the effectiveness of the LQR controller in maintaining system stability under various conditions. Recommendations are also provided for tuning and improving controller parameters, as well as exploring alternative strategies for handling external disturbances.

INDICE GENERAL

AGRADECIMIENTO	i
DEDICATORIA	ii
APROBACIÓN DEL TUTOR	iii
TRIBUNAL DE GRADO	iv
DECLARACION DEL AUTOR	ii
RESUMEN	iii
ABSTRACT	v
INDICE GENERAL	vii
INTRODUCCIÓN	1
CAPÍTULO I	2
1.1. ANTECEDENTES	2
1.2. DESCRIPCIÓN DEL PROYECTO	3
1.3. OBJETIVOS DEL PROYECTO	4
1.3.1. OBJETIVO PRINCIPAL	4
1.3.2. OBJETIVOS ESPECÍFICOS	4
1.4. JUSTIFICACIÓN	4
1.5. Alcance del proyecto	6
1.6. METODOLOGÍA DE LA INVESTIGACIÓN	7
1.6.1. INVESTIGACIÓN DESCRIPTIVA	7
1.6.2. INVESTIGACIÓN DOCUMENTAL	7
1.6.3. INVESTIGACIÓN APLICADA	7
1.7. RESULTADOS ESPERADOS	7
CAPÍTULO II	9
2.1. MARCO CONTEXTUAL	9
2.2. MARCO CONCEPTUAL	9
2.2.1. Sistemas de control automático	9

2.2.2. Señal.....	13
2.2.3. Proceso.....	13
2.2.4. Péndulo invertido.....	13
2.2.5. Estado	14
2.2.6. Variables de estado.....	14
2.2.7. Vector de estado.....	15
2.2.8. Espacio de estado.....	16
2.2.9. Ecuaciones en espacio de estado.....	16
2.2.10. Puntos de equilibrio.....	18
2.2.11. Controlabilidad y observabilidad	19
2.2.12. Modelo matemático	24
2.2.13. Diseño del servosistema de control.....	26
2.2.13. Control óptimo.....	27
2.2.14. Variable aleatoria y procesos estocásticos.....	31
2.2.15. Procesos estocásticos.....	34
2.2.16. Cinemática de robots balancín.....	35
2.2.17. Elementos del robot balancín	37
2.3. MARCO TEORICO.....	45
CAPÍTULO III	47
3.1. Componentes de la propuesta.....	47
Componentes físicos	47
Componentes lógicos	49
3.2. Diseño de la propuesta	49
3.2.1. Sistema electromecánico e identificación de variables.....	49
3.2.2. Identificación de la planta.....	50
3.2.3. Modelado matemático del sistema	54
3.2.4. Análisis de diseño del control LQR.....	63

3.2.5. Diseño del control LQR.....	68
3.2.6. Diseño de control LQR en Simulink	70
3.2.7. Diseño de control LQR en Arduino	71
CAPÍTULO IV.....	73
4.1. Parámetros de evaluación.....	73
4.2. Evaluaciones.....	73
4.2.2. Respuesta función escalón	74
4.2.6. Comparación del controlador LQR.....	78
CONCLUSIONES	81
RECOMENDACIONES	83
REFERENCIAS	85
ANEXOS	91

ÍNDICE DE FIGURAS

Figura 1. Representación del sistema de lazo abierto [31].	10
Figura 2. Representación del sistema de lazo cerrado [33].	11
Figura 3. Diseño del observador.	21
Figura 4. Sistema de control en lazo cerrado	22
Figura 5. Esquema del proceso de modelado matemático [50].	24
Figura 6. Servosistema de control [53].	27
Figura 7. Regulador cuadrático lineal de lazo cerrado [55].	30
Figura 8. Grafica de un proceso estocástico.	35
Figura 9. Diagrama en lazo abierto del sistema.	51
Figura 10. Estimaciones del mejor modelado matemático del sistema.	51
Figura 11. Circuito interno de un motor.	57
Figura 12. Respuesta de sistema en lazo abierto.	66
Figura 13. Sistema de control LQR en Simulink.	71
Figura 14. Resultado de la simulación de la posición con entrada escalón de 4 cm.	74
Figura 15. Resultado de la simulación del ángulo de inclinación con entrada escalón de 4 cm.	75
Figura 16. Resultado de la simulación de la señal de control con entrada escalón de 4 cm.	75
Figura 17. Resultado de la prueba física de la posición con entrada escalón de 4 cm.	76
Figura 18. Resultado de la prueba física de la señal de control con entrada escalón de 4 cm.	77
Figura 19. Resultado de la prueba física del ángulo de inclinación con entrada escalón de 4 cm.	77
Figura 20. Comparación del sobre impulso del sistema simulado vs el real.	78
Figura 21. Comparación de la señal de control del sistema simulado vs el real.	79
Figura 22. Comparación del ángulo de inclinación del sistema simulado vs el real.	80

ÍNDICE DE TABLAS

Tabla 1. Representación de las variables a utilizar	50
Tabla 2. Respuesta de los estados evaluados.....	52
Tabla 3. Valores de la matriz A	52
Tabla 4. Valores de la matriz B.	53
Tabla 5. Valores de la matriz C.	53
Tabla 6. Valores de la matriz D.	53
Tabla 7. Valores en espacio de estado continuo de la matriz A.	64
Tabla 8. Valores en espacio de estado continuo de la matriz B.	64
Tabla 9. Valores en espacio de estado continuo de la matriz C.	64
Tabla 10. Valores en espacio de estado continuo de la matriz D	64
Tabla 11. Valores en espacio de estado continuo de la matriz A.	65
Tabla 12. Valores en espacio de estado continuo de la matriz B.	65
Tabla 13. Valores en espacio de estado continuo de la matriz C.	65
Tabla 14. Valores en espacio de estado continuo de la matriz D	65
Tabla 15. Polos del sistema	66
Tabla 16. Matriz de controlabilidad.....	67
Tabla 17. Matriz de observabilidad	68
Tabla 18. Especificaciones del control LQR.	73
Tabla 19. Resultados de las simulaciones para una entrada escalón	75
Tabla 20. Resultados de las pruebas para una entrada escalón.....	77
Tabla 21. Comparación del máximo sobreimpulso.	79
Tabla 22. Comparación del tiempo de estabilización.....	79
Tabla 23. Comparación de la señal de control.....	79
Tabla 24. Comparación del ángulo de inclinación.	80
Tabla 25. Comparación del máximo sobreimpulso.	80

ÍNDICE DE ANEXOS

Anexo 1. Reporte de plagio.....	91
Anexo 2. Proceso de linealización de las ecuaciones.....	91
Anexo 3. Código en matlab para la obtención de matrices de estado mediante jacobiano.....	95
Anexo 4. Código en Matlab para la inestabilidad.	96
Anexo 5. Código en Matlab para la controlabilidad.	97
Anexo 6. Código Matlab para la Observabilidad.....	98
Anexo 7. Código en Matlab del controlador LQR.....	99
Anexo 8. Código en Arduino del control LQR.....	102
Anexo 9. Extensión del Código en Arduino para la obtención del ángulo.	103
Anexo 10. Extensión del Código en Arduino para la obtención del ángulo. ...	104
Anexo 11. Código en Arduino del LQR para estabilización.....	105
Anexo 12. Clases y funciones necesarias para implementar el control LQR. .	109
Anexo 13. Funcionalidad de control de motores del robot balancín.....	111
Anexo 14. Funciones para controlar motores y medir la velocidad de los encoders.....	113
Anexo 15. Configuración de interrupciones para contar los pulsos de los encoders.....	114
Anexo 16. Herramienta System Identification.....	115
Anexo 17. Control del robot balancín.	115

INTRODUCCIÓN

En un mundo donde la automatización y la robótica están en constante evolución, surge la necesidad imperante de abordar el diseño y control eficiente de sistemas electromecánicos. La problemática fundamental radica en la complejidad de desarrollar sistemas adaptables que puedan mantener un equilibrio dinámico óptimo ante entornos variables y demandas cambiantes, sin comprometer la estabilidad ni la eficiencia operativa. En este sentido, la presente investigación se enfoca en explorar el potencial del control lineal cuadrático (LQR) aplicado a un sistema específico: el robot balancín. Este dispositivo, reconocido por su capacidad para mantener el equilibrio dinámico en condiciones desafiantes, representa un reto técnico y científico que, si se aborda de manera efectiva, podría tener un impacto significativo en diversos campos, incluyendo la robótica, la ingeniería de control y la automatización.

El primer capítulo de esta tesis sienta las bases al ofrecer una visión integral del proyecto. En él, se delimitan los objetivos de la investigación, se justifica la relevancia del estudio y se expone la metodología empleada para abordar los desafíos propuestos. A medida que avanzamos en la lectura, nos adentramos en el segundo capítulo, donde se teje el marco contextual, conceptual y teórico que sustenta el trabajo. Este análisis crítico de antecedentes, conceptos clave y teoría asociada proporciona un fundamento sólido para la comprensión profunda del proyecto.

El tercer capítulo, núcleo central de esta investigación, desentraña los componentes esenciales de la propuesta. Desde el diseño meticuloso del robot balancín hasta la creación de modelos teóricos y la implementación del control LQR mediante simulaciones en MATLAB, cada paso se documenta con rigor científico y técnico. Este capítulo representa el puente entre la teoría y la aplicación práctica, mostrando cómo la investigación se traduce en avances tangibles en el campo del control de sistemas electromecánicos.

Finalmente, el cuarto capítulo revela los resultados obtenidos durante la implementación del controlador LQR en el sistema. Aquí, se analizan críticamente los logros alcanzados y se abren puertas hacia posibles mejoras o ajustes futuros. Con esta tesis, se busca no solo contribuir al entendimiento de la estabilización de robots balancines, sino también establecer un punto de partida para futuras investigaciones en el vasto y emocionante dominio del control de sistemas dinámicos.

CAPÍTULO I

1.1. ANTECEDENTES

En la actualidad, los sistemas de control son utilizados por sus múltiples aplicaciones, especialmente sobre los sistemas no lineales variantes en el tiempo, en el área de la robótica los sistemas de control moderno se han convertido en uno de los puntos claves para las investigaciones científicas y desarrollo de tecnologías avanzadas [1].

En la década de los 60 se publicaron las ecuaciones para el diseño de un regulador lineal cuadrático (LQR) por Rudolph E. Kalman y otros, gracias al conocimiento del álgebra lineal, y matrices junto con la teoría de control previa permitió el tratamiento de múltiples entradas y salidas para desarrollar la teoría de control del controlador LQR [2].

Los robots balancín son elementos electrónicos que están basados en el modelo del péndulo invertido, el cual busca solucionar los problemas que se originan de este modelo [3]. El oportuno diseño electrónico y estructural que tienen los robots balancín han permitido que estos se conviertan en caso de estudio perfecto para el desarrollo de nuevos controladores, algunos de ellos son el control PID(Proporcional, Integral y Derivativo) [4], LGR(Lugar Geométrico de las Raíces) [5], LQG(lineal cuadrático gaussiano) [6], entre otros, estos controladores permiten que los sistemas electrónicos como el robot balancín permanecer estables frente a perturbaciones externas.

Debido a la versatilidad de estos robots es posible aplicar diferentes controladores que permitan lograr la estabilidad de estos sistemas electrónicos, uno de estos controles ha sido el control PID, pero este control no es tan eficiente como el controlador LQR, como se menciona en un estudio realizado en [7] el cual indica que a través del enfoque LQR se realiza un nuevo criterio de selección de matrices Q y R para llegar a la frecuencia natural deseada y el amortiguamiento del sistema de lazo cerrado. En otros trabajos de investigación realizaron diferentes comparativas de los controladores clásicos frente al controlador LQR, como es el caso de [8] donde se comparan los controladores PID y LQR en un sistema robótico con dos grados de libertad, los resultados demostraron que frente a perturbaciones el control PID regresa al punto de referencia en un tiempo de 6 segundos y el control LQR logra regresar en un tiempo de 3 segundos, de esta manera se demostró que para este tipo de sistemas el

control LQR es más óptimo.

Este resultado es similar en [9], donde se pone a prueba estos mismo controladores en un sistema de suspensión, los resultados de esta investigación demostró que el desplazamiento máximo de este sistema con controlador PID es de 60.1 mm y con un controlador LQR es de 44.9 mm.

Todo esto permite continuar en la investigación en el tema, evaluar el tipo de control sobre un sistema electrónico físico existente en el laboratorio de investigación de FACSISTEL el cual es un robot balancín, obtener propios resultados que conlleven al aporte en la investigación científica en esta área usando el conocimiento electrónico y tecnológico nacional para el desarrollo de nuevos productos electrónicos con tecnologías avanzadas.

1.2. DESCRIPCIÓN DEL PROYECTO

El proyecto consta de tres etapas.

El objetivo principal de este proyecto es poder implementar un LQR en un sistema electromecánico para lograr la estabilización de este, por lo que es fundamental el diseño del modelo matemático el cual se lograra mediante diferentes parámetros, restricciones y características para lograr dicho objetivo, el sistema de control de la planta del robot balancín tiene diferentes elementos electrónicos como un microcontrolador, un puente H para que los motores D/C para que estos puedan girar en ambos sentidos y un giroscopio, para poder lograr la estabilización de la planta se desarrollaran numerosas simulaciones por medios de software especializados para el control automático, el cual junto con la comunicación UART permitirá que se establezca la comunicación entre la planta de sistema electromecánico y el software de simulación. Se realiza la comparación entre el modelo no lineal del sistema contra el modelo linealizado y el modelo identificado de la planta y con estos resultados se procede al desarrollo del controlador, se evalúa el controlador y se lo implementa en la simulación, una vez con resultados deseados se lo implementa en la planta para obtener la respuesta del sistema. Con esto se procede a realizar el análisis de resultados del control diseñado para determinar si este cumple o no con los requerimientos planteados.

1.3. OBJETIVOS DEL PROYECTO

1.3.1. OBJETIVO PRINCIPAL

Diseñar e implementar un controlador lineal cuadrático aplicado a la estabilización de un sistema electromecánico en variable de estado denominado robot balancín.

1.3.2. OBJETIVOS ESPECÍFICOS

- Analizar los componentes electrónicos de la planta para el diseño del modelo matemático del sistema robótico balancín para evaluar el comportamiento de este, estableciendo los parámetros, restricciones y características propias del sistema.
- Establecer la comunicación entre el sistema electromecánico del robot balancín y el software de simulación mediante comunicación UART para el intercambio de datos.
- Implementar un controlador LQR para la regulación del sistema no lineal electromecánico del robot balancín mediante un software de simulación numérico.
- Analizar los resultados del comportamiento del sistema para verificar el funcionamiento del control LQR sobre el sistema electromecánico del robot balancín mediante diferentes simulaciones y pruebas.

1.4. JUSTIFICACIÓN

El control de realimentación por estados nace de la necesidad de controlar procesos en diferentes sistemas electrónicos en donde la teoría de control clásica no tiene efecto significativo, o no cumple alguna exigencia, en la actualidad la necesidad de calidad, velocidad, y versatilidad en la producción, o en el desarrollo de sistemas electrónicos complejos ha llevado al desarrollo de teorías de control modernas para mejorar los procesos, el LQR puede aplicarse a sistemas electromecánicos no lineales para diferentes fines según las áreas a las que se vaya a emplear. Este control ya existe en equipos industriales, sistemas electromecánicos como UAV (vehículo aéreo no tripulado), segway para el transporte de personas, vehículos autónomos como las sillas de dos ruedas para personas con capacidades especiales y en equipo armamentístico avanzado en el área militar, por lo que es urgente el estudio, la investigación y el desarrollo de aplicaciones con este tipo de controladores sobre los sistemas electrónicos como estudiantes de ingeniería electrónica.

El logaritmo de un sistema LQR permite reducir la cantidad de trabajo para optimizar el control de retroalimentación de estado de un sistema no lineal [10]. Debido a esto en Latinoamérica en diferentes universidades se han realizado guías para las prácticas de laboratorio donde se aplican este tipo de controlador en diferentes sistemas para realizar los estudios respectivos [11].

Los robots balancín en los últimos años se han vuelto muy populares debido a sus aplicaciones en el transporte, como es el caso de la plataforma de movilidad robótica segway la cual se enfoca en la movilidad de un individuo de un lugar a otro utilizando técnicas de estabilización como los controladores LQR [12]. Otra aplicación que tienen los robots balancín es la creación de una silla electrónica de dos ruedas para la movilidad de las personas con capacidades especiales, esta silla cumple con las características fundamentales de estos sistemas no lineales lo que permite la aplicación de controladores basados en la teoría de control para su estabilización [13].

Existen módulos didácticos para el estudio del robot balancín, diferentes configuraciones y modelos, pero todos guardan el mismo propósito, desarrollar control automático sobre el sistema [14], estos robots son usados para los trabajos investigativos como se demuestra en [15] el cual indica que se usó este tipo de sistema no lineal para el diseño de un control para la estabilización, en [16] se realizó un esquema de control para el equilibrio de un segway auto balanceado, y en [17] se eligió un sistema electromecánico de dos ruedas para el debido control de la estabilización de este. Estos estudios posteriores a esta investigación demuestran que debido a la infraestructura que tienen estos sistemas electrónicos aplicar los diferentes logaritmos de control basados en la teoría de control permiten el estudio adecuado de estos controladores.

Existen investigaciones en el Ecuador donde aplican el control LQR sobre variables de presión para la optimización del sistema donde llegan a concluir que la aplicación de este controlador a escalas industriales es viable debido a la robustez del control [18]. Además, un análisis comparativo entre un control LQR y un control clásico demostró que el controlador LQR es más rápido que los otros controladores [19].

En otros trabajos investigativos han realizado la construcción de robots balancín mediante componentes electrónicos los cuales son controlados por programación automática los cuales tienen como base la teoría de control para el desplazamiento libre

de este robot [20], o mediante el control por medio de dispositivos móviles [21] para poder realizar la construcción de la placa electrónica del robot fue importante el conocimiento previo de electrónica y de los componentes utilizados.

A nivel provincial en el repositorio de la Universidad Península de Santa Elena, se desarrolla proyectos orientados al control lógico programable, usando dispositivos robustos industriales como se muestran en [22] donde indica que usaron sistemas de control automático para el proceso de enfriamiento de un mosto mediante el uso de controladores PLC, pero no se ha encontrado investigaciones relacionadas con control óptimo, perteneciente a la teoría de control en sistemas electrónicos o electromecánicos. Por lo tanto, es un justificante para el desarrollo de este trabajo de investigación.

Una área muy importante en la ingeniería es el desarrollo de sistemas de control avanzado, para aplicaciones de procesos altamente no lineales y variantes en el tiempo, esta tesis es una oportunidad para explorar estos sistemas, en este caso el robot balancín, de igual manera la investigación está enmarcada dentro de experimentos sobre estos sistemas, y como es el caso la implementación de este algoritmo de control permite explorar de manera específica la técnica y la teoría de control moderno.

1.5. Alcance del proyecto

El presente proyecto de propuesta tecnológica tiene como finalidad realizar un diseño de control LQR y su implementación sobre el sistema electromecánico no lineal presentado para conseguir una mejor respuesta.

Mediante el diseño de un modelo matemático se podrá identificar los puntos de equilibrio del sistema no lineal electromecánico y en que rango existe inestabilidad, para esto se desarrolla la teoría de modelado basándose en artículos y tesis previas.

Una vez construido el modelo matemático se lo simula, para observar la dinámica del sistema y corregir detalles del modelado. Una vez validado el modelado se procede a realizar el controlador LQR basado en la teoría de control óptimo cuadrático lineal.

Al terminar la simulación del sistema se procede a la implementación del controlador sobre el sistema electromecánico denominado robot balancín (planta – físico) mediante comunicación UART, para la respectiva evaluación del controlador diseñado. Para culminar se procede a obtener los resultados para su respectivo análisis, conclusiones y recomendaciones para futuros proyectos.

1.6. METODOLOGÍA DE LA INVESTIGACIÓN

1.6.1. INVESTIGACIÓN DESCRIPTIVA

El proyecto actual se enfoca en una detallada descripción de los sistemas de control LQR (Linear Quadratic Regulator), abordando tanto sus aspectos teóricos como prácticos. Se pone énfasis en comprender los componentes clave de este método, incluyendo la formulación de criterios de costo cuadrático, la determinación de las matrices de ponderación Q y R, y la obtención de las leyes de control LQR. Además, se lleva a cabo un análisis exhaustivo del comportamiento de estos sistemas cuando se aplican a una planta con características no lineales. El proyecto busca proporcionar una comprensión integral de cómo los sistemas LQR interactúan con sistemas no lineales en aplicaciones prácticas, con el objetivo de optimizar el rendimiento en condiciones del mundo real y contribuir al desarrollo de soluciones avanzadas de control.

1.6.2. INVESTIGACIÓN DOCUMENTAL

El proyecto basado en la teoría del control LQR busca aplicar los principios de control óptimo en sistemas lineales a diferentes áreas profesionales. Esto implica una investigación profunda de métodos de control, seguida del diseño y desarrollo de controladores LQR específicos para aplicaciones relevantes, como la automatización industrial, la robótica, la aeroespacial y la economía. El proyecto se centra en la optimización del rendimiento de sistemas y procesos mediante la implementación de controladores LQR, y los resultados se documentan y comunican de manera efectiva para su aplicación práctica.

1.6.3. INVESTIGACIÓN APLICADA

El proyecto tiene como objetivo analizar la aplicación de los principios de sistemas de control en el ámbito científico, investigando cómo estos conceptos se utilizan en diversas disciplinas científicas. Se pretende identificar ejemplos concretos de aplicaciones de control en experimentos, mediciones y procesos científicos, evaluando sus beneficios y desafíos. El proyecto tiene como propósito destacar la relevancia y el impacto de la tecnología de control en la mejora de la precisión, la eficiencia y la resolución de problemas en la investigación científica, abarcando múltiples campos y promoviendo una comprensión más amplia de su aplicabilidad en la ciencia contemporánea.

1.7. RESULTADOS ESPERADOS

Los resultados previstos tras el desarrollo de este proyecto son los siguientes:

- Construir modelos matemáticos del sistema electromecánico del robot balancín.
- Evaluar el comportamiento del sistema estableciendo los parámetros, restricciones y características.
- Establecer la comunicación entre el sistema electromecánico con el software de simulación elegido.
- Implementar el algoritmo de control LQR para el sistema electromecánico no lineal.
- Complementar el aprendizaje de los sistemas de control moderno con la aplicación de control LQR a una planta con sistema no lineal.
- Observar los aspectos más resaltantes en la ejecución del proceso de control del sistema electromecánico.
- Analizar los resultados del algoritmo para establecer nuevos temas de investigación dentro de la facultad.
- Establecer nuevos conocimientos base para futuras investigaciones a fines con este proyecto.

CAPÍTULO II

2.1. MARCO CONTEXTUAL

En procesos industriales los sistemas de control automático representan una optimización de los procesos industriales por lo que se han convertido en una necesidad [23], estudios en América latina y el caribe demostraron que solo el 60% de las industrias cuentan con algún tipo de automatización en sus instalaciones [24]. En el Ecuador esta realidad no está muy distante debido a que menos del 50% de total de las industrias manufacturera cuentan con algún tipo de automatización en las instalaciones [25].

La industria robótica ha permitido crear nuevos dispositivos que permiten la movilidad de personas con diferentes dispositivos que se adaptan a las necesidades de cada individuo como es el caso del Segway [26], Hoverboard [27], Genny Mobility [28]. En Ecuador la robótica está en procesos de desarrollo con diferentes propuestas tecnológicas por parte de diferentes instituciones que investigan los sistemas de control moderno, la Universidad Estatal Península de Santa Elena es la institución educativa dentro de la provincia de Santa Elena que se dedica a desarrollar dichos sistemas mediante la investigación.

Se usa un sistema electromecánico de dos ruedas, conocido como robot balancín, el cual cuenta con una entrada y varias salidas, como planta a controlar, el modelo matemático que se realiza tiene como fin conocer el comportamiento de este sistema, al igual que el algoritmo de control LQR de manera teórica, se incorpora al sistema antes mencionado para el posterior análisis d resultados.

La implementación de un controlador LQR para un sistema electromecánico de dos ruedas en variable de estado, tiene la finalidad de incrementar el conocimiento sobre la teoría de control automático.

2.2. MARCO CONCEPTUAL

2.2.1. Sistemas de control automático

El control automático se refiere a la capacidad de un sistema para regular su propio comportamiento de forma autónoma, sin la intervención directa de un operador humano. En términos generales, se trata de un conjunto de técnicas y algoritmos diseñados para ajustar los parámetros de un sistema en función de las condiciones

ambientales y las entradas externas, de modo que el sistema pueda operar de manera eficiente y precisa [29].

Los sistemas de control automático se utilizan en una amplia variedad de aplicaciones, desde la regulación de la temperatura en un hogar o edificio, hasta el control de procesos industriales complejos en la fabricación de productos químicos o la producción de energía [29]. En el control de un proceso se deben aplicar diferentes configuraciones para la elaboración de sistemas automáticos, las cuales son sistemas de control de lazo abierto y sistema de control de lazo cerrado.

Sistema de control de lazo abierto

Un sistema de control de lazo abierto es un tipo de sistema de control en el que la acción de control se realiza sin tener en cuenta la retroalimentación o la medición de la salida del sistema. En otras palabras, en un sistema de lazo abierto, la señal de control se genera en función de una referencia o un conjunto de parámetros predefinidos y se aplica directamente al proceso o sistema que se está controlando, sin considerar cómo afecta o cambia la salida. En los sistemas de lazo abierto el valor de salida requerido dependerá de la calibración de diversos elementos, estos pueden ser actuadores y reguladores [30]. En la figura 2 se puede apreciar una representación de este sistema.



Figura 1. Representación del sistema de lazo abierto [31].

Sistema de control de lazo cerrado

Un sistema de control de lazo cerrado es un tipo de sistema de control en el que se utiliza retroalimentación para controlar y mantener una variable de interés (la salida del sistema) dentro de los límites deseados o en seguimiento de una referencia. En contraste con el sistema de control de lazo abierto, en el que no se utiliza retroalimentación, el sistema de lazo cerrado mide la salida real del sistema y ajusta la señal de control en consecuencia para mantener la salida en la posición deseada [32]. En la figura 3 se puede apreciar una representación de este sistema.

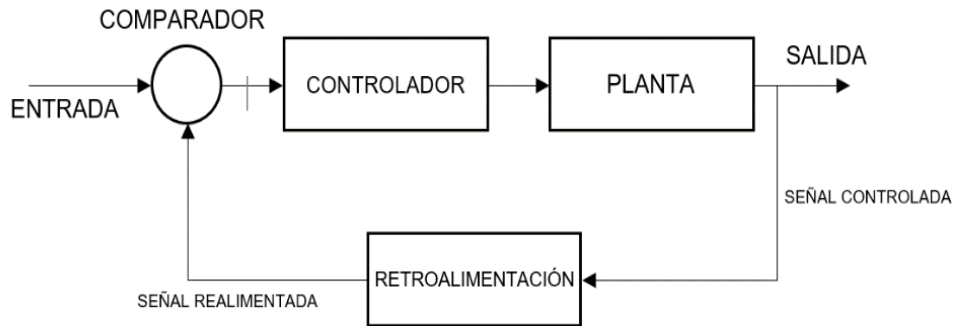


Figura 2. Representación del sistema de lazo cerrado [33].

Cada sistema de control tiene diferentes características que se acoplan al requerimiento del sistema y el fin que se tenga para cada proceso de control automatizado, en los cuales se suministran múltiples algoritmos de control para poder perfeccionar una determinada función según el requerimiento del sistema.

Teoría de control

La teoría de control constituye un campo multidisciplinario que emplea tanto herramientas matemáticas como ingenieriles para examinar y concebir sistemas dinámicos. Su influencia se extiende al comportamiento de estos sistemas, permitiendo la optimización de variables con el fin de alcanzar resultados específicos y deseables. Este ámbito abarca una amplia gama de aplicaciones y se enfoca en comprender cómo las señales de entrada impactan en las variables de estado y salida de un sistema dado. Asimismo, proporciona metodologías para el desarrollo de controladores que garanticen la estabilidad, precisión y respuestas eficientes ante diversas condiciones operativas y requerimientos de rendimiento [33].

Elementos de un sistema de control

Todo sistema de control está determinado por diferentes elementos que intervienen simultáneamente para poner a disposición una respuesta determinada según el objetivo establecido. Estos sistemas tienen la particularidad de estar conformados por los siguientes componentes:

Variables: se refiere a una cantidad física o medida que se utiliza para controlar y ajustar automáticamente el comportamiento del sistema. Las variables pueden ser de entrada o de salida, y se miden y utilizan para permitir que el sistema tome decisiones y realice ajustes en función de las condiciones del entorno y las entradas recibidas. En resumen, una variable en un sistema de control automatizado es un elemento esencial para el correcto funcionamiento del sistema, ya que permite el control y ajuste automático en respuesta a las condiciones cambiantes del entorno [34].

Sensores: es un dispositivo que mide una o varias variables físicas, como la temperatura, la presión, la posición o la velocidad, y convierte esa medida en una señal eléctrica o electrónica que puede ser procesada por el sistema de control. Los sensores son elementos críticos en los sistemas de control automatizado, ya que proporcionan información en tiempo real sobre el comportamiento del sistema y su entorno [35], [36].

Actuadores: un actuador es un dispositivo que convierte una señal eléctrica o electrónica en una acción física, como el movimiento de un motor o el movimiento de un brazo robótico. Los actuadores son componentes críticos en los sistemas de control automatizado, ya que permiten que el sistema realice ajustes y cambios en su entorno físico en función de las decisiones tomadas por el sistema de control [36], [37].

Unidad de control: la unidad de control es el componente central que recopila y procesa la información de los sensores del sistema y toma decisiones en función de esa información. La unidad de control utiliza algoritmos y software especializado para interpretar los datos de los sensores y enviar señales de control a los actuadores del sistema para ajustar su comportamiento [38].

Planta: la planta se refiere al proceso o sistema físico que está siendo controlado. La planta es el objeto de control en el sistema, que puede ser un motor, una máquina de producción, un brazo robótico o cualquier otro dispositivo o sistema físico que se esté automatizando. La planta recibe las entradas y produce salidas que son medidas y controladas por el sistema de control automatizado. La eficacia del sistema de control depende en gran medida de la comprensión de la planta, ya que se requiere un conocimiento profundo de sus características y comportamiento para lograr un control óptimo [38].

Para este trabajo de titulación la planta donde se incorporará el control compete a un sistema electromecánico de dos ruedas conocido también como robot balancín, es un tipo de robot que utiliza un principio de control de retroalimentación para mantener su equilibrio en una posición vertical. Los elementos principales de este sistema son: un microcontrolador para realizar el respectivo control del sistema, un MPU el cual proporcionará los datos de posición del robot balancín, dos motores DC para corregir la posición de este.

El proceso que tiene un sistema electromagnético de dos ruedas se representa en la figura 4 con un diagrama de bloques, los cuales permiten describir mediante bloques los componentes de un sistema y las respectivas conexiones de este, donde se pueden

visualizar los actuadores y los sensores, conocer el comportamiento de estos es importante porque de ellos depende la estabilización de la planta.

2.2.2. Señal

Una señal $x(t)$ se define como una función cuyo valor está asociado de manera real o escalar a la variable de tiempo t . La expresión "con valor real" implica que, para cualquier valor específico de la variable temporal t , la señal en ese instante corresponde a un número real [39].

Señal continua

Una señal continua es una representación matemática de un fenómeno que varía de manera suave y continua en el tiempo, lo que significa que puede tomar cualquier valor en un intervalo continuo, la variable independiente suele ser el tiempo (t), y la señal está definida para todos los valores de tiempo dentro de un intervalo continuo. Las señales continuas pueden describir fenómenos físicos como el movimiento de un objeto, la temperatura en un lugar específico a lo largo del tiempo, o cualquier otra cantidad que cambie de manera continua y sin discontinuidades [39].

Señal discreta

Una señal discreta es una representación matemática de un fenómeno que solo toma valores en puntos específicos o instantes discretos en el tiempo, la variable independiente suele ser el tiempo (números enteros, por ejemplo), y la señal toma valores solo en esos puntos discretos. Las señales discretas son comúnmente utilizadas en el procesamiento digital de señales, donde la información se representa y manipula de manera discreta, como en sistemas de comunicación digital, procesamiento de imágenes y audio digital, entre otros [39].

2.2.3. Proceso

Un proceso se refiere a una secuencia de eventos o actividades que se llevan a cabo para lograr un objetivo específico. Puede aplicarse a una amplia variedad de áreas, desde procesos industriales y comerciales hasta procesos biológicos o de pensamiento [40].

2.2.4. Péndulo invertido

El péndulo invertido es un sistema mecánico que consiste en un péndulo común (una masa suspendida de un punto fijo por un hilo o barra) pero con la particularidad de que la masa se encuentra colgando hacia arriba, en lugar de estar colgando hacia abajo como en un péndulo tradicional.

Debido a su configuración y la influencia de la gravedad, es un sistema inestable desde el punto de vista del equilibrio. Esto significa que, si se deja en libertad sin control alguno, la masa del péndulo invertido caerá hacia un lado o hacia el otro y no se mantendrá en una posición estable en equilibrio [41].

El péndulo invertido es un ejemplo clásico de un sistema no lineal y altamente no lineal debido a las fuerzas gravitatorias y la cinemática involucrada. Controlar un péndulo invertido implica mantenerlo en posición vertical, a pesar de las perturbaciones externas.

Para controlar un péndulo invertido, se utilizan técnicas de control como el control proporcional-integral-derivativo (PID), controladores de retroalimentación lineal, control óptimo, redes neuronales, entre otros. Estos controladores permiten ajustar continuamente la posición del péndulo para mantenerlo en equilibrio [42].

Los péndulos invertidos también se utilizan en aplicaciones educativas y de investigación para estudiar conceptos de control automático y teoría de sistemas dinámicos. Además, los robots de péndulo invertido son populares en la robótica para desarrollar sistemas que pueden mantener el equilibrio de forma autónoma.

2.2.5. Estado

El estado de un sistema dinámico se refiere al conjunto más reducido de variables, conocidas como variables de estado, que en el instante inicial $t = t_0$, junto con la información sobre la entrada para $t > t_0$, son suficientes para predecir de manera completa y precisa el comportamiento del sistema en cualquier momento futuro $t \geq t_0$. Este concepto fundamental no se encuentra restringido exclusivamente a sistemas físicos, sino que también encuentra aplicación en diversos dominios como sistemas biológicos, económicos y sociales, así como en otras disciplinas [43].

2.2.6. Variables de estado

Las variables de estado son parámetros o magnitudes que caracterizan completamente el estado de un sistema dinámico en un momento dado y contienen la información esencial para prever su comportamiento futuro. Estas variables son fundamentales en el análisis y la modelización de sistemas dinámicos en ingeniería, física, biología, economía y otras disciplinas [44].

En un sistema dinámico, la elección de las variables de estado es crucial para describir de manera adecuada su evolución en el tiempo. Estas variables son seleccionadas de manera que, conocidas sus condiciones iniciales en un instante

particular, junto con la información sobre las entradas subsiguientes, se pueda determinar completamente el estado del sistema en cualquier momento futuro.

Por ejemplo, en el caso de un péndulo simple, las variables de estado podrían ser la posición y la velocidad angulares en un instante dado. Conocer estas dos variables en un momento inicial y tener información sobre las fuerzas o perturbaciones aplicadas al péndulo permitiría predecir su movimiento en cualquier momento futuro [44].

En sistemas más complejos, las variables de estado pueden ser más numerosas y representar aspectos específicos del sistema. Por ejemplo, en un sistema eléctrico, las variables de estado pueden incluir voltajes y corrientes en componentes clave. En biología, las variables de estado pueden abordar la concentración de ciertas sustancias químicas o el estado de activación de ciertos genes.

El concepto de variables de estado es central en la teoría de control y modelado de sistemas dinámicos. Ayuda a simplificar la descripción de sistemas complejos, facilita el diseño de estrategias de control y permite comprender mejor la dinámica y el comportamiento de los sistemas en estudio [44].

2.2.7. Vector de estado

El término "vector de estado" se refiere a un conjunto ordenado de variables de estado que caracterizan completamente el estado de un sistema dinámico en un instante específico. Este conjunto de variables se organiza en forma de vector para representar de manera compacta y estructurada la información esencial del sistema en un momento dado [45].

En el contexto de sistemas dinámicos y control, el vector de estado se utiliza para representar la situación actual del sistema y su evolución en el tiempo. La dimensión de este vector está determinada por la cantidad de variables de estado seleccionadas para describir el sistema [45]. Por ejemplo, si un sistema tiene tres variables de estado (x_1 , x_2 , x_3), el vector de estado se expresaría como un vector columna:

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_3(t) \end{bmatrix} \quad 1$$

Aquí, x es el vector de estado y representa completamente el estado del sistema en el instante considerado [45].

2.2.8. Espacio de estado

El "espacio de estado" es un concepto fundamental en la teoría de control y sistemas dinámicos. Se refiere al conjunto de todos los posibles estados que puede tomar un sistema dinámico en un momento dado. Este conjunto de estados se representa en un espacio multidimensional, donde cada dimensión corresponde a una variable de estado [45].

El espacio de estado es un enfoque matemático que permite representar y analizar sistemas dinámicos de manera integral. En este enfoque, las ecuaciones que describen la evolución temporal del sistema se expresan en forma matricial, relacionando las tasas de cambio de las variables de estado con las propias variables y posiblemente con las entradas al sistema.

Por ejemplo, consideremos un sistema lineal e invariante en el tiempo con tres variables de estado (x_1, x_2, x_3). El espacio de estado asociado a este sistema estaría representado por un espacio tridimensional, donde cada punto en ese espacio corresponde a un conjunto específico de valores para las variables de estado en un instante dado [45].

La representación matricial del sistema en espacio de estado sería:

$$\dot{x} = Ax + Bu \quad 2$$

Donde:

x es el vector de estado que describe la situación actual del sistema.

\dot{x} es la tasa de cambio de las variables de estado con respecto al tiempo.

A es una matriz que relaciona las variables de estado con sus tasas de cambio.

B es una matriz que relaciona las entradas externas (u) con las variables de estado.

El análisis en el espacio de estado proporciona una perspectiva completa y unificada de la dinámica del sistema, permitiendo la aplicación de herramientas matriciales y técnicas de álgebra lineal para comprender y controlar el comportamiento del sistema en diferentes condiciones y ante diversas entradas.

2.2.9. Ecuaciones en espacio de estado

Las ecuaciones en el espacio de estados son una representación matricial de un sistema dinámico. Estas ecuaciones describen cómo evolucionan las variables de estado de un sistema en función del tiempo y de las entradas externas [46]. La forma general de

las ecuaciones en el espacio de estados para un sistema lineal e invariante en el tiempo es:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}\tag{3}$$

Donde:

$x(t)$ es el vector de estado del sistema en el tiempo t .

$u(t)$ es el vector de entrada al sistema en el tiempo t .

$y(t)$ es el vector de salida del sistema en el tiempo t .

$A, B, C,$ y D son matrices que caracterizan las propiedades y las interconexiones del sistema.

La primera ecuación representa la dinámica del sistema, indicando cómo cambian las variables de estado con respecto al tiempo en función de su estado actual, las entradas y las matrices de estado (A y B). La segunda ecuación relaciona las salidas del sistema ($y(t)$) con las variables de estado y las entradas a través de las matrices C y D [46].

Cada elemento de las matrices $A, B, C,$ y D tiene un significado específico en términos de las propiedades y la interconexión de las variables de estado, las entradas y las salidas del sistema [46].

La representación en espacio de estados es particularmente útil para el análisis y el diseño de sistemas de control. Permite utilizar herramientas de álgebra lineal y teoría de matrices para comprender la estabilidad del sistema, diseñar controladores y analizar el comportamiento dinámico en diferentes condiciones de entrada [46].

Jacobiano

Para calcular los jacobianos, es necesario realizar derivadas parciales de cada expresión con respecto a cada una de las variables del sistema. Estas derivadas parciales se utilizan para construir las cuatro matrices requeridas, siguiendo las ecuaciones que se detallan a continuación. Este proceso implica una evaluación exhaustiva de cómo cada componente del sistema afecta a las otras variables, lo que nos permite comprender mejor la interacción entre los diferentes elementos y su influencia en el comportamiento global del sistema. Este enfoque analítico nos brinda una base sólida para abordar con precisión la dinámica y la estabilidad del sistema, lo que resulta fundamental para el diseño y la implementación efectivos del controlador LQR [47].

$$A = \left. \frac{\partial f(x, u)}{\partial x} \right|_{\substack{x=x_e \\ u=u_e}}$$

$$B = \left. \frac{\partial f(x, u)}{\partial u} \right|_{\substack{x=x_e \\ u=u_e}}$$

$$C = \left. \frac{\partial g(x, u)}{\partial x} \right|_{\substack{x=x_e \\ u=u_e}}$$

$$D = \left. \frac{\partial g(x, u)}{\partial u} \right|_{\substack{x=x_e \\ u=u_e}}$$

4

2.2.10. Puntos de equilibrio

Para comprender cómo se comporta el sistema en un entorno estable, se utilizan los puntos de equilibrio como puntos de referencia. El proceso implica calcular las derivadas parciales del sistema en relación con las variables cuando se encuentran en estos puntos de equilibrio, que generalmente tienen un ángulo cercano a cero. La matriz jacobiana se crea con estas derivadas, que proporciona información útil sobre la estabilidad y el comportamiento dinámico de la planta en esos momentos.

Matrices linealizadas

Las matrices que describen la planta del modelo linealizado en su configuración de lazo abierto son elementos fundamentales para entender y controlar el sistema. Estas matrices encapsulan las características dinámicas del sistema, permitiendo su análisis y diseño de estrategias de control efectivas. Representan las relaciones entre las variables de entrada y salida, brindando una representación matemática precisa de cómo evoluciona el sistema en respuesta a las entradas y perturbaciones. Su comprensión es esencial para el diseño y la implementación de algoritmos de control que garanticen un rendimiento óptimo y una estabilidad adecuada del sistema en diversas condiciones operativas.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \\ \frac{\partial f_3}{\partial u} \\ \frac{\partial f_4}{\partial u} \end{bmatrix} u$$

5

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} u$$

6

2.2.11. Controlabilidad y observabilidad

La controlabilidad y la observabilidad son conceptos fundamentales en la teoría de control de sistemas dinámicos. Estos conceptos se utilizan para evaluar la capacidad de un sistema para ser controlado y observado, respectivamente [48].

Controlabilidad

La controlabilidad se refiere a la capacidad de un sistema para ser llevado desde cualquier estado inicial a cualquier estado deseado en un tiempo finito mediante la aplicación adecuada de las entradas de control. En términos más simples, un sistema es controlable si es posible influir y dirigir su evolución para alcanzar cualquier configuración deseada de sus variables de estado.

Matemáticamente, un sistema es completamente controlable si la matriz de controlabilidad, formada por las columnas de la matriz de control B, junto con sus potencias sucesivas hasta llegar a la dimensión del sistema, tiene rango completo. Si el rango el sistema es igual a la dimensión del espacio de estado por lo tanto se vuelve completamente controlable [48].

La controlabilidad en el tiempo es una propiedad importante en el diseño de sistemas de control, ya que garantiza que el sistema pueda ser manejado de manera efectiva y pueda alcanzar cualquier estado deseado dentro de un periodo de tiempo limitado. Este concepto es fundamental para la teoría del control y tiene aplicaciones prácticas en diversos campos, como la ingeniería de control, la robótica y la automatización.

Para este análisis se tomará la ecuación 4 donde esta es lineal e invariante en el tiempo, representada en la siguiente ecuación

$$\dot{x}(t) = Ax(t) + Bu(t) \quad x(t_0) = x_0 \quad 7$$

Se dice que un estado $x \in \mathbb{R}^n$ es controlable si para un tiempo inicial t_0 hay un tiempo finito mayor que el tiempo inicial $t_f > t_0$, junto con una señal de entrada establecida en el rango t_0 hasta t_f con un estado inicial $x(t_0) = x_0$.

$$x(t_f) = e^{A(t_f-t_0)}x + \int_{t_0}^{t_f} e^{A(t_f-\tau)}Bu(\tau)d\tau \quad 8$$

$$= 0 \in \mathbb{R}^n$$

Donde la ecuación 4 será controlable si $x \in \mathbb{R}^n$ hasta el origen. Utilizar esta definición suele ser larga, para ello se realiza la comprobación de controlabilidad del sistema en Matlab, donde este verifica que el rango de las matrices y compara el rango con el número de filas de la matriz de controlabilidad, cuando el rango es igual al número de filas se considera controlable, el resultado de este procedimiento indica que el sistema es controlable

Observabilidad

La observabilidad se refiere a la capacidad de determinar el estado completo de un sistema a partir de sus salidas observadas. Un sistema es observable si, dado un tiempo suficiente, el estado actual del sistema puede ser determinado con precisión utilizando solo las salidas observadas.

La observabilidad está relacionada con la matriz de observabilidad, que se forma mediante las filas de la matriz de observación C, junto con sus potencias sucesivas hasta llegar a la dimensión del sistema. Si la matriz de observabilidad tiene rango completo y es igual a la dimensión del espacio de estado, por lo tanto el sistema es completamente observable [48].

Utilizando la ecuación 3 en espacio de estados se toma la señal de entrada $u(t)$ y de salida $y(t)$ y suponiendo que estas se pueden medir en un intervalo de tiempo finito, buscando el estado inicial $x(t_0) = x_0$ y conociendo la señal de entrada se puede determinar la trayectoria de estado mediante la siguiente ecuación

$$x(t) = e^{A(t-t_0)}x_0 + \int_{t_0}^t e^{A(t-\tau)}Bu(\tau)d\tau \quad \text{para } t \geq t_0 \quad 9$$

Como $u(t)$ es un valor conocido la respuesta de estado cero se puede obtener de la respuesta $y(t)$, la cual también tiene un valor conocido, para poder obtener la respuesta de entrada cero con la siguiente ecuación

$$y(t) - \left[\int_{t_0}^t Ce^{A(t-\tau)}Bu(\tau)d\tau + Du(t) \right] = Ce^{A(t-t_0)}x_0 \quad 10$$

La comprobación de observabilidad implica determinar si es factible estimar los estados internos de un sistema basándose únicamente en las salidas observadas. Este proceso se realiza mediante la construcción de una matriz de observabilidad a partir de las matrices del sistema y la evaluación de su rango.

Observador de estado

Un componente dentro del sistema de control, llamado observador de estados o estimador de estados, desempeña la función de estimar las variables de estado utilizando las mediciones tanto de las variables de salida como de las variables de control. En este contexto, la noción de observabilidad desempeña un papel crucial.

Para el diseño del observador de estado se van a utilizar las ecuaciones dinámicas para definir un sistema de espacio de estado.

$$\dot{x}(t) = Ax(t) + Bu(t) \quad x(t_0) = x_0 \quad 11$$

$$y(t) = Cx(t) \quad 12$$

A estas se define un observador lineal que acepte $u(t)$ e $y(t)$ como entradas, donde la estimación de $x(t)$ represente el estado. De esta manera el observador tendrá la siguiente forma

$$\dot{\hat{x}} = A\hat{x}(t) + bu(t) + L[y(t) - \hat{y}(t)] \quad \hat{x}(0) = \hat{x}(t) \quad 13$$

Donde L es la ganancia del observador, este observador se puede apreciar en la siguiente figura

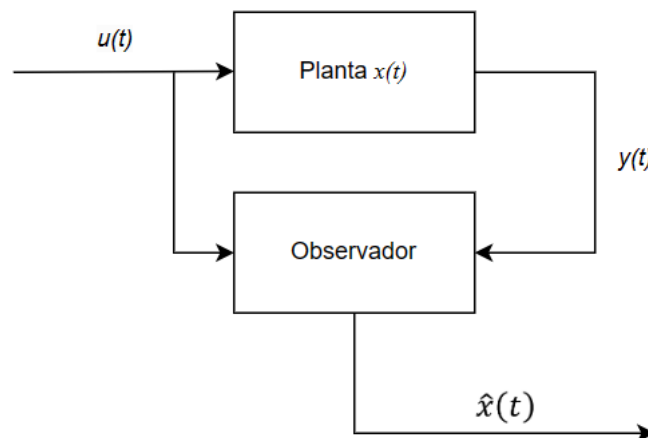


Figura 3. Diseño del observador.

Diseño mediante ubicación de polos

Para este diseño se emplearán todos los polos de lazo cerrado y no solo los polos dominantes como se emplea en el diseño habitual.

Sea el sistema de control:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}\tag{3}$$

La señal de control será

$$u = -kx\tag{14}$$

Esto implica que la señal de control u se calcula utilizando la información de un estado particular. Este enfoque se conoce como realimentación de estado. La matriz K , que tiene dimensiones $1 \times n$, es referida como la matriz de ganancia de la realimentación de estado. Se asume que las variables de estado estarán a disposición para la realimentación.

En este sistema de bucle cerrado, no hay entradas directas, y su propósito es mantener la salida en cero. A pesar de la posible presencia de perturbaciones, la salida se apartará de cero. No obstante, gracias al esquema de realimentación de estado implementado en el sistema, la salida eventualmente regresará a cero en respuesta a la entrada de referencia, que siempre se mantiene en cero. Un sistema de esta índole, en el cual la entrada de referencia es constante en cero, se clasifica como un sistema regulador, el cual se puede apreciar en la siguiente figura.

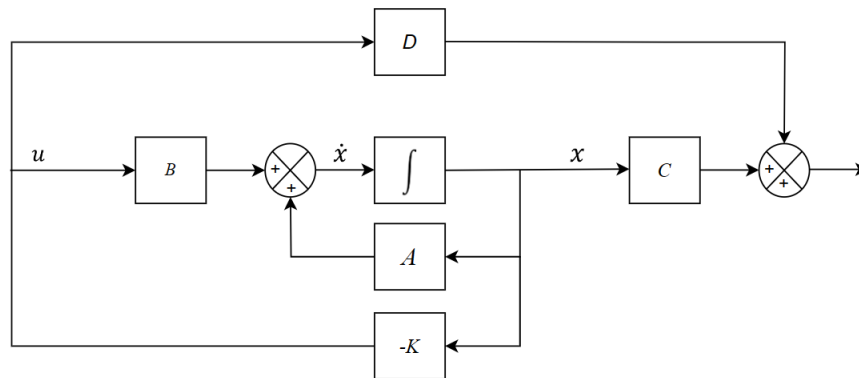


Figura 4. Sistema de control en lazo cerrado

Verificación de los eigenvalores de la matriz $A-BK$

Para determinar la estabilidad del sistema, es necesario verificar los eigenvalores de la matriz $A-BK$. Estos valores de K no indican directamente si el sistema es estable o no. La verificación de los eigenvalores de la matriz $A-BK$ es importante en el contexto del control LQR porque proporciona información sobre la estabilidad del sistema de control resultante.

La solución de esta ecuación está dada por:

$$x(t) = e^{(A-BK)t}x(0) \quad 15$$

Donde $x(0)$ es el estado inicial originado por influencias externas. La estabilidad y las propiedades de la respuesta temporal se establecen mediante los valores característicos de la matriz resultante de restar la matriz B multiplicada por la matriz K de la matriz A. Si se selecciona la matriz K de manera apropiada, la matriz resultante se vuelve asintóticamente estable, permitiendo que para cualquier condición inicial diferente de cero ($x(0) \neq 0$), la variable de estado $x(t)$ tienda a cero a medida que el tiempo se acerca a infinito.

Los eigenvalores de la matriz A-BK son conocidos como los polos del controlador. Si estos son ubicados en la región del semiplano izquierdo en el plano s , entonces la variable de estado $x(t)$ tiende a cero conforme el tiempo tiende a infinito.

Cálculo de la ganancia de control K

El cálculo de la ganancia de control K es una parte fundamental en el diseño de sistemas de control retroalimentado. La ganancia de control K se utiliza para determinar cómo las señales de control se aplican al sistema para lograr un comportamiento deseado.

Una de las razones más importantes para calcular la ganancia de control K es estabilizar el sistema. El objetivo principal del control es mantener el sistema en un estado deseado o en un punto de referencia. El controlador ajusta las señales de control de manera que el sistema converja y se establezca en este estado deseado, incluso si el sistema es inherentemente inestable o tiene dinámicas no deseadas.

La matriz de ganancias de retroalimentación K, que asegura que los eigenvalores de $A - BK$ sean los valores deseados k_1, k_2, \dots, k_n , se determina a través de los siguientes pasos. Si k_i es un eigenvalor complejo el conjugado de este también será un eigenvalor de $A - BK$.

Verificar la condición de controlabilidad del sistema. Si el sistema es completamente controlable en estado, continúe con los pasos siguientes.

Utilizar el polinomio que se encuentra en la matriz A como punto de partida.

$$|sI - A| = s^n + a_1s^{n-1} + a_{n-1}s + a_n \quad 16$$

Identificar la matriz de transformación T que transforma la ecuación de estado del sistema a su forma controlable canónica. No es crucial expresar la ecuación de estado en la forma canónica controlable; simplemente, se requiere determinar la matriz

T . La ecuación para la matriz de transformación T se presenta mediante la siguiente expresión:

$$T = MW \quad 17$$

Donde M es la matriz de controlabilidad y W está representada por:

$$W = \begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \cdots & 1 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_1 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Donde las a_i son los términos que conforman el polinomio característico.

Con los eigenvalores deseados (los polos deseados en bucle cerrado), elabore el polinomio característico deseado y determine los valores de cada α .

$$(s - \mu_2)(s - \mu_2) \dots (s - \mu_n) = s^n + \alpha_1 s^{n-1} + \alpha_{n-1} s + \alpha_n \quad 18$$

Se obtiene la matriz de ganancia de retroalimentación de estado K necesaria a partir de la siguiente ecuación.

$$K = [\alpha_n - a_n \quad \alpha_{n-1} - a_{n-1} \quad \dots \quad \alpha_2 - a_2 \quad \alpha_1 - a_1] T^{-1} \quad 19$$

2.2.12. Modelo matemático

Un modelo matemático es una representación abstracta de un sistema o fenómeno del mundo real mediante ecuaciones u otras herramientas matemáticas. Estos modelos se utilizan en una amplia variedad de campos, como la física, la ingeniería, la biología, la economía y muchas otras disciplinas, para comprender, analizar y predecir el comportamiento del sistema o fenómeno real en diferentes situaciones [49].

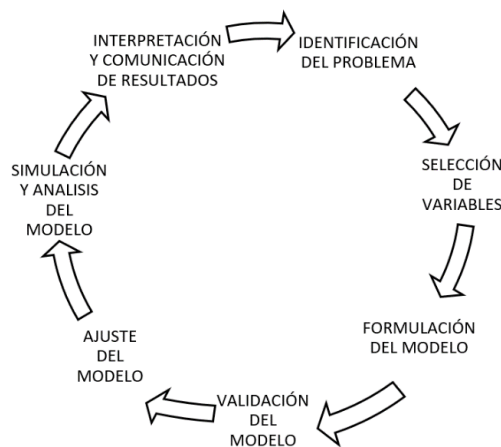


Figura 5. Esquema del proceso de modelado matemático [50].

Las etapas para la construcción de un modelo matemático (ver Figura) pueden variar según la aplicación específica y la complejidad del sistema que se está

modelando. Sin embargo, a grandes rasgos, se pueden identificar algunas etapas generales comunes en la construcción de un modelo matemático:

Identificación del problema: en esta etapa se define claramente el sistema o fenómeno que se desea estudiar. Esto implica identificar las variables relevantes y los parámetros que influyen en el sistema [50].

Formulación de las ecuaciones: Una vez que se comprende el sistema, se formulan las ecuaciones matemáticas que describen las relaciones entre las variables y parámetros del sistema. Dependiendo de la naturaleza del sistema y el problema que está tratando, estas ecuaciones pueden ser ecuaciones diferenciales, ecuaciones algebraicas, ecuaciones integrales, etc [50].

Simplificaciones y suposiciones: En muchos casos, los modelos matemáticos involucran simplificaciones y suposiciones para hacer que las ecuaciones sean más manejables. Estas simplificaciones pueden incluir la eliminación de términos que tienen un impacto menor en el comportamiento general del sistema o la asunción de relaciones lineales en lugar de no lineales [50].

Solución de las ecuaciones: Una vez formuladas las ecuaciones, se procede a resolverlas matemáticamente. Esto puede implicar encontrar soluciones analíticas o utilizar métodos numéricos, dependiendo de la complejidad de las ecuaciones [50].

Validación y ajuste: Los modelos matemáticos deben validarse utilizando datos experimentales o evidencia empírica. Si el modelo no se ajusta bien a los datos observados, es posible que se requieran ajustes en las ecuaciones o suposiciones del modelo [50].

Utilización del modelo: El modelo se puede usar para predecir el comportamiento del sistema en diversas condiciones una vez validado o para analizar el impacto de cambios en los parámetros del sistema [50].

Interpretación y conclusiones: Los resultados obtenidos a través del modelo matemático se interpretan y se utilizan para sacar conclusiones sobre el sistema o fenómeno estudiado [50].

Técnicas de representación

Para realizar el modelado matemático de uno o diversos procedimientos reales se deben utilizar diferentes técnicas, las cuales van a depender de las particularidades internas y externas de la planta a modelar, estas técnicas se dividen en:

Técnica de representación externa: Esta técnica tiene como fin representar un determinado sistema por medio de la función de transferencia, se caracterizan por ser la técnica adecuada para los sistemas SISO debido a que estos están conformados por una variable de entrada y de salida, esta técnica no es adecuada para los sistemas MIMO debido a que estos están conformados por múltiples variables de entrada y de salida. [51]

Técnica de representación interna: Esta técnica permite representar los sistemas MIMO de manera adecuada, como también a los sistemas SISO, esta técnica es conocida como el modelado de sistemas en espacios de estado porque permite representar el comportamiento de la planta en forma matricial, las cuales dependerán de las variables que conforman el sistema [51].

2.2.13. Diseño del servosistema de control

En el diseño de sistemas de control, el desarrollo de servosistemas efectivos desempeña un papel crucial en la ingeniería de control. Un aspecto fundamental de este proceso se encuentra en la capacidad del sistema para seguir de manera precisa y estable las referencias deseadas. En el caso de plantas de tipo 0, que carecen de un integrador en su función de transferencia, surge la necesidad de estrategias específicas para garantizar un seguimiento de estado estacionario sin error para las entradas de referencia [52].

Cuando nos enfrentamos a una planta de tipo 0, caracterizada por su incapacidad para seguir referencias constantes sin un error de estado estacionario, el enfoque clásico sugiere la inserción de un integrador en el camino directo entre la planta y el comparador de error. Este paso es esencial para convertir el sistema en un servomecanismo de tipo 1 y superar los desafíos asociados con la falta de capacidad integral en la planta.

La metodología propuesta fusiona el enfoque clásico, que implica la adición de un término integral de error, con modernos métodos de diseño de retroalimentación de estado. Este enfoque combina la capacidad de corregir el error de estado estacionario con la estabilidad del circuito cerrado y las cualidades deseables de respuesta transitoria [53].

La capacidad de resistir la incertidumbre en la ecuación de estado de bucle abierto es un componente importante de esta estrategia de diseño. La estabilidad del bucle cerrado garantiza el rendimiento de seguimiento de estado estacionario, lo que

significa que el sistema puede adaptarse a variaciones y perturbaciones en la descripción del sistema [53].

Para realizar este diseño el sistema tendrá una entrada y una salida donde se cumplirá que la ecuación de estado de lazo abierto es controlable.

La fórmula de ley de control será de la siguiente forma:

$$\begin{aligned} \dot{\xi} &= r(t) - y(t) \\ u(t) &= -kx(t) + k_1\xi(t) \end{aligned} \quad 20$$

Donde $r(t)$ es la referencia de paso que debe seguir la salida $y(t)$. Cuando se establece la derivada temporal $\dot{\xi}(t)$ para igualar el error de seguimiento $r(t) - y(t)$, se aprecia que $\xi(t)$ es la integral del error de seguimiento [52].

La combinación entre la ley de control de retroalimentación y la característica de la planta en su configuración de lazo abierto da como resultado la formulación de la ecuación de estado correspondiente al sistema de lazo cerrado en la figura 4 se aprecia el diseño del servosistema de control.

$$\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \dot{\xi}(t) \end{bmatrix} &= \begin{bmatrix} A - BK & Bk_1 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \xi(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(t) \\ y(t) &= [C \quad 0] \begin{bmatrix} x(t) \\ \xi(t) \end{bmatrix} \end{aligned} \quad 21$$

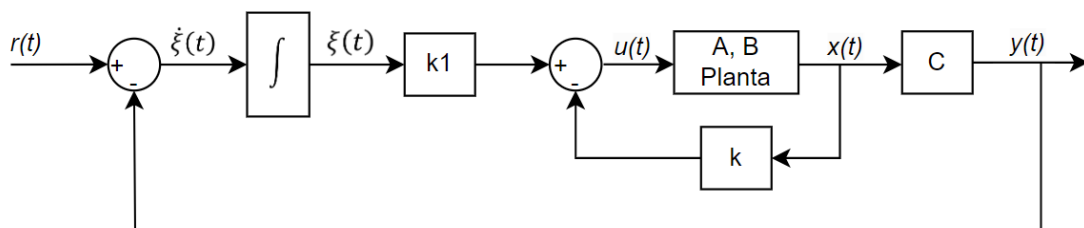


Figura 6. Servosistema de control [53].

2.2.13. Control óptimo

El control óptimo es un área de la teoría de control que se centra en encontrar estrategias de control que optimizan ciertos criterios de rendimiento. En este enfoque, se busca determinar la secuencia óptima de acciones de control para un sistema dinámico dado, de modo que se minimice o maximice una función de costo o rendimiento específica [54].

En el proceso de concebir un sistema de control o un regulador óptimos, surge la necesidad de establecer una directriz para la toma de decisiones de control actual, respetando ciertas limitaciones, con el fin de minimizar una medida específica que

evalúa la desviación del comportamiento real respecto al comportamiento ideal. Esta medida, comúnmente representada por el índice de desempeño elegido, constituye una función cuyo valor se interpreta como un indicador de qué tan cercano se encuentra el rendimiento del sistema real al rendimiento deseado [44]. En esencia, el índice de desempeño sirve como una herramienta fundamental para cuantificar y evaluar la eficacia del sistema de control en la consecución de sus objetivos, proporcionando una guía valiosa para ajustar y perfeccionar el rendimiento del sistema.

Regulador óptimo cuadrático lineal

Un Regulador Óptimo Cuadrático Lineal (LQR, por sus siglas en inglés Linear Quadratic Regulator) es un enfoque dentro de la teoría de control óptimo que se centra en diseñar un controlador lineal que minimice una función cuadrática de costo. Este método es especialmente aplicable a sistemas dinámicos lineales y, en términos generales, implica la minimización de la suma ponderada de los errores de estado y las entradas de control, a lo largo de un horizonte de tiempo finito. *gf*

Este regulador está expresado por la ecuación de estado

$$\dot{x} = A(t)x(t) + B(t)u(t) \quad 22$$

Es necesario que el controlador reduzca al mínimo un índice de rendimiento con la siguiente estructura:

$$J(u) = \frac{1}{2}x^T(t_1)S(t_1)x(t_1) + \frac{1}{2} \int_{t_0}^t (x^T(t)R_1x(t) + u(t)^T R_2 u(t)) dt \quad 23$$

Las matrices de ponderación de las señales de estado y control están semidefinidas y definidas de manera positiva, respectivamente:

$$R_1(t) \geq 0 \text{ y } R_2(t) \geq 0, \quad \forall t \quad 24$$

Se puede describir también que:

$$\lambda(t) = P(t)x(t) \quad 25$$

Por lo que la señal de control estará dada por la siguiente ecuación

$$u(t) = -R_2^{-1}(t)B(t)P(t)x(t) \quad 26$$

La ecuación 9 indica que el vector de control se obtiene derivando el vector de estado, es decir, se ha implementado un sistema de control en bucle cerrado, lo cual es altamente beneficioso en términos de aplicaciones. La cuestión pendiente radica en determinar la matriz $P(t)$. Esta matriz debe cumplir con una ecuación diferencial que resulta de la derivación con respecto al tiempo de la expresión 8.

$$\dot{\lambda}(t) = \dot{P}(t)x(t) + P(t) \dot{x}(t) \quad 27$$

Esta ecuación es resoluble para cualquier $x(t)$ siempre y cuando la matriz $P(t)$ cumpla con la ecuación diferencial que se deriva a partir de la misma. En otras palabras, la existencia de una solución general para $x(t)$ está condicionada a la satisfacción de la ecuación diferencial por parte de la matriz $P(t)$. Esta ecuación diferencial es aquella que se la conoce como ecuación de Riccati, expresada en la en la ecuación 11.

$$-\dot{P}(t) = P(t)A(t) + A^T(t)P(t) - P(t)B(t)R_2^{-1}(t)B(t)P(t) + R_1(t) \quad 28$$

La ecuación de Riccati se manifiesta como un conjunto interrelacionado de n^2 ecuaciones diferenciales de primer orden no lineales. Estas ecuaciones se encuentran en el intervalo temporal $t_0 \leq t \leq t_1$, y tiempo final t_1 , están sujetas a las condiciones de contorno n^2 . La conclusión extraída de esta compleja estructura es que, bajo condiciones suficientemente flexibles, la ecuación de Riccati poseerá una solución real. A partir de esta solución, se abre la posibilidad de diseñar un controlador que optimice el índice de rendimiento cuadrático del sistema. Este enfoque brinda una herramienta poderosa para la optimización y mejora del desempeño en sistemas dinámicos lineales, al permitir la determinación de estrategias de control que minimizan de manera efectiva los costos asociados con el comportamiento del sistema [55].

Considerando que, en términos generales, la ecuación de Riccati posee una solución única, se puede inferir que el problema del regulador cuadrático lineal tiene una solución única para el controlador óptimo. Esta solución adopta la forma de una señal de control que actúa como un controlador de retroalimentación de estado.

$$u(t) = -K(t)x(t) \quad 29$$

La ganancia de retroalimentación que varía con el tiempo, representada por $K(t)$, es conocida como la ganancia LQR (Regulador Cuadrático Lineal) o la ganancia óptima del regulador. Puede derivarse de la ecuación 9 de la siguiente manera:

$$K(t) = R_2^{-1}(t)B^T(t)P(t) \quad 30$$

El diagrama de bloques de retroalimentación de estado para el LQR se muestra en la figura 7. De manera general, la ganancia LQR mostrará dependencia temporal, incluso cuando el sistema sea Invariante en el Tiempo Lineal (LTI) y la función de costo incorpore matrices de peso constantes. El valor óptimo, es decir, el mínimo del índice de rendimiento se puede determinar cómo:

$$J_{min} = \frac{1}{2}x^T(t_0)P(t_0)x(t_0) \quad 31$$

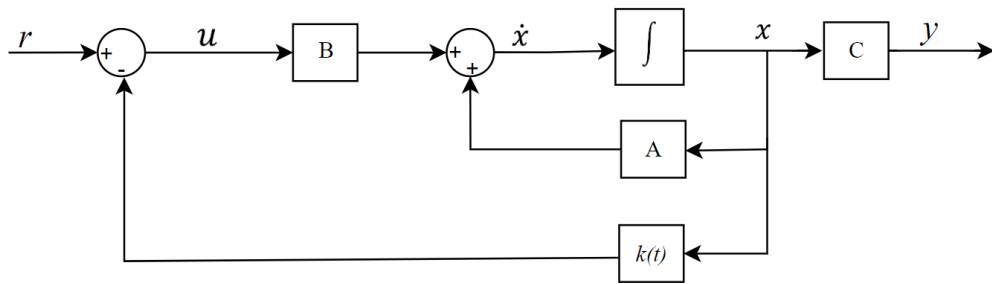


Figura 7. Regulador cuadrático lineal de lazo cerrado [55].

Regulador óptimo cuadrático en estado estacionario.

El controlador regulador lineal cuadrático conocido por sus siglas en inglés LQR es un método de control óptimo para sistemas dinámicos lineales, donde minimiza una función de costo cuadrática [56]. El objetivo del control LQR es encontrar una ley de control lineal (una relación entre la señal de control y el estado del sistema) que minimice una función de costo cuadrática dada por:

$$J(u) = \lim_{t \rightarrow \infty} \left[\int_{t_0}^t (x^T R_1 x(t) + u(t) R_2 u(t)) dt \right] \quad 32$$

Donde:

J : es la función de costo total.

x : es el vector de estado del sistema.

u : es la señal de control.

Q : es una matriz de ponderación de estado, que penaliza el error en el estado.

R : es una matriz de ponderación de control, que penaliza la magnitud del control.

Se debe suponer que el sistema es invariante en el tiempo por lo que las matrices son constantes y $R_1 > 0$ y $R_2 > 0$.

Dado que la meta de los controladores LQR convencionales es llevar los estados incrementales al estado cero de la manera óptima posible, el vector de estado $x(t)$ tenderá a aproximarse al vector cero a medida que t_1 tienda a infinito, siempre y cuando el sistema de lazo cerrado sea estable [56]. Por lo tanto, no es significativo incluir un término de estado final en este contexto. Esta consideración es equivalente a establecer $S(t_1) = 0$ en la ecuación 7.

Al ser un sistema invariante en el tiempo, el valor de J_{min} deberá ser independiente del tiempo t_0 , haciendo que la matriz P sea constante, provocando que $\dot{P} = 0$ y la ecuación de Riccati se reduzca a la ecuación 16.

$$0 = A^T P + PA + R_1 - PBR_2^{-1}B^T P \quad 33$$

La ecuación 16 puede presentar diversas soluciones, sin embargo, se puede demostrar que únicamente una de ellas es semidefinida positiva, siempre y cuando el sistema sea estabilizable. Esta solución específica es la que conduce al valor mínimo del índice de desempeño expresada en la siguiente formula:

$$J_{\min} = \frac{1}{2} x_0^T P_{\infty} x_0 \quad 34$$

La matriz de estado estacionario quedaría reestructurada de la siguiente manera:

$$K_{\infty} = R_2^{-1} B^T P_{\infty} \quad 35$$

El objetivo del LQR es encontrar la ley de control $u(t)$ que minimiza esta función de costo, la cual queda expresada:

$$u(t) = -K_{\infty} x(t) \quad 36$$

Teorema de LQR continuo

Se debe considerar la ecuación de estado invariante en el tiempo, descrita a continuación:

$$\dot{x} = Ax(t) + Bu(t) \quad 37$$

El cual debe estar sujeto el índice de rendimiento expresado en la siguiente ecuación:

$$J(u) = \int_0^{t_1} (x^T(t)R_1x(t) + u^T(t)R_2u(t))dt \quad 38$$

Por lo tanto, se verifica lo siguiente:

Si el sistema definido por la ecuación 19 es estabilizable y la pareja de matrices $A, \sqrt{R_1}$ es detectable, entonces la ecuación algebraica de Riccati tiene una única solución que es definida positiva. Esta solución, representada por P_{∞} , conduce al valor mínimo del índice de rendimiento. Por lo tanto, la ley de retroalimentación de estado resultante es:

$$u(t) = -K_{\infty} x(t) = -R_2^{-1} B^T P_{\infty} \quad 39$$

Proporcionando de esta manera un lazo cerrado que es asintóticamente estable.

2.2.14. Variable aleatoria y procesos estocásticos

En ingeniería, las variables aleatorias y los procesos estocásticos son herramientas fundamentales para modelar y analizar sistemas que están sujetos a la variabilidad e incertidumbre inherentes. Las variables aleatorias se utilizan para representar magnitudes cuyos valores no pueden preverse con certeza, como la

resistencia de un material, el tiempo de vida de un componente o la llegada de señales en sistemas de comunicación [57].

Los procesos estocásticos, por otro lado, son esenciales para describir la evolución temporal de sistemas dinámicos en presencia de aleatoriedad. En ingeniería de señales, los procesos estocásticos se aplican para modelar fenómenos como el ruido en comunicaciones o las fluctuaciones en señales biomédicas [57]. Además, en control de sistemas, los procesos estocásticos son empleados para analizar y diseñar estrategias de control robustas teniendo en cuenta la variabilidad y perturbaciones del sistema.

Variable aleatoria

Las variables aleatorias son conceptos fundamentales en teoría de probabilidad y estadísticas. Representan cantidades numéricas cuyos valores no pueden preverse con certeza, ya que están sujetos a la variabilidad o aleatoriedad. Las variables aleatorias pueden ser discretas o continuas, dependiendo de si toman valores distintos o forman un conjunto infinito no numerable [58].

- **Discretas:** Estas variables toman un conjunto finito o contablemente infinito de valores posibles. Por ejemplo, el número de caras al lanzar un dado es una variable aleatoria discreta [58].
- **Continuas:** Estas variables pueden tomar cualquier valor en un intervalo continuo. Por ejemplo, la altura de una persona o la duración de un evento pueden modelarse como variables aleatorias continuas [58].

Cada variable aleatoria tiene asociada una función de distribución de probabilidad que describe la probabilidad de que la variable tome valores específicos. En el caso discreto, esto se llama función de masa de probabilidad, y en el continuo se llama función de densidad de probabilidad [58]. Las variables aleatorias son esenciales en el modelado y análisis de fenómenos inciertos en diversas disciplinas, incluyendo estadísticas, ingeniería, economía y ciencias de la computación.

Función de densidad de probabilidad.

La función de densidad de probabilidad (PDF, por sus siglas en inglés) es una herramienta fundamental en la teoría de probabilidad y estadísticas, especialmente cuando se trabaja con variables aleatorias continuas [59]. La función de densidad de probabilidad describe la distribución de probabilidades de una variable aleatoria continua a lo largo de su rango de valores.

Para una variable aleatoria continua X , la función de densidad de probabilidad $f(x)$ tiene las siguientes propiedades:

1. $f(x) \geq 0$ para todos los valores de x .
2. El área bajo la curva de la función de densidad de probabilidad en todo su rango es igual a $1 \int_{-\infty}^{\infty} f(x)dx = 1$
3. La probabilidad de que la variable aleatoria X caiga en un intervalo específico $[a, b]$ se calcula como la integral de $f(x)$ desde a hasta b : $\int_a^b f(x)dx$.

$$f(x) \geq 0$$

$$\int_{-\infty}^{\infty} f(x)dx = 1 \quad 40$$

$$\int_a^b f(x)dx = P(a \leq x \leq b)$$

Características de la distribución de probabilidad

Las distribuciones de probabilidad son modelos matemáticos que describen cómo se distribuyen las probabilidades de los posibles resultados de una variable aleatoria. Cada distribución de probabilidad tiene características distintivas que la definen y la hacen adecuada para modelar diferentes tipos de fenómenos [60].

Valor esperado

La media, también conocida como valor esperado, es una medida fundamental en estadísticas que representa el promedio ponderado de los posibles valores de una variable aleatoria. Matemáticamente, se calcula sumando cada valor posible multiplicado por su probabilidad asociada.

$$E(X) = \int_{-\infty}^{\infty} xf(x)dx \quad 41$$

La media es esencial para entender la tendencia central de una distribución y proporciona un punto de referencia que indica dónde, en promedio, se encuentran los valores.

Varianza

La varianza y la desviación estándar son medidas cruciales que cuantifican la dispersión de los valores en torno a la media en una distribución de probabilidad. La varianza se calcula como la media de los cuadrados de las diferencias entre cada valor y la media, proporcionando una medida de la extensión de los datos. Una varianza mayor indica que los valores tienden a alejarse más de la media, reflejando una mayor

dispersión en la distribución. La desviación estándar, simplemente la raíz cuadrada de la varianza tiene la misma interpretación, pero en la misma unidad que los datos originales.

$$var(X) = \sigma_X^2 = E(X - \mu)^2 \quad 42$$

Ambas medidas son esenciales para comprender la variabilidad y la consistencia de los datos en relación con la media, permitiendo evaluar la amplitud de la distribución y su grado de concentración alrededor del valor central.

La varianza se calcula de la siguiente manera, siempre que X sea una variable aleatoria continua:

$$var(X) = \int_{-\infty}^{\infty} (X - \mu)^2 f(x) dx \quad 43$$

Covarianza

La covarianza es una medida estadística que cuantifica la relación lineal entre dos variables aleatorias. Específicamente, mide cómo las variaciones conjuntas de estas dos variables se comparan con sus medias respectivas. La fórmula para la covarianza ($Cov(X, Y)$) entre dos variables aleatorias X e Y es:

$$Cov(X, Y) = E[(X - \mu_X)(Y - \mu_Y)] = E(XY) - \mu_X\mu_Y \quad 44$$

Para calcular la covarianza se debe utilizar la siguiente formula siempre que X y Y sean variables aleatorias continuas:

$$Cov(X, Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (X - \mu_X)(Y - \mu_Y) f(x, y) dx dy \quad 45$$

$$Cov(X, Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (XY) f(x, y) dx dy - \mu_X\mu_Y$$

2.2.15. Procesos estocásticos

Los procesos estocásticos son modelos matemáticos que describen la evolución temporal de sistemas en los cuales la aleatoriedad o la incertidumbre desempeñan un papel significativo. En lugar de seguir trayectorias determinísticas fijas, los procesos estocásticos incorporan elementos aleatorios que influyen en la evolución del sistema con el tiempo, estos procesos pueden ser discretos o continuos.

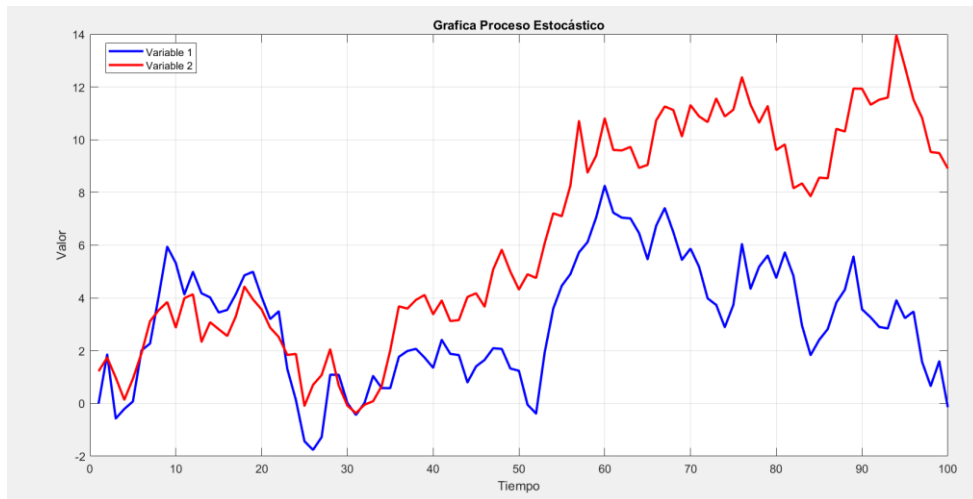


Figura 8. Gráfica de un proceso estocástico.

Un proceso estocástico se caracteriza por su espacio de estados, que representa todos los posibles estados que puede tomar el sistema, y por su ley de evolución, que describe cómo las probabilidades de transición entre estados cambian con el tiempo.

La utilidad de los procesos estocásticos radica en su capacidad para modelar fenómenos que no pueden preverse con certeza, como las fluctuaciones en los precios de las acciones, la variabilidad en la señal de comunicación o la dinámica de sistemas biológicos. Además, los procesos estocásticos son fundamentales en la teoría de control, la simulación, y la toma de decisiones bajo incertidumbre. Su estudio proporciona herramientas poderosas para entender y abordar sistemas complejos en presencia de aleatoriedad.

2.2.16. Cinemática de robots balancín

La cinemática de robots balancín se refiere al estudio del movimiento y la posición de robots que se mueven sobre una base móvil o un punto de pivote. Estos robots utilizan un sistema de balanceo para mantenerse estables mientras se mueven, y su cinemática se basa en las leyes de la física que gobiernan el movimiento y la rotación [61].

En los robots balancín, la cinemática directa se utiliza para calcular la posición y la orientación del robot en función de la posición y la velocidad de su base móvil o punto de pivote. La cinemática inversa, por otro lado, se utiliza para determinar las posiciones y velocidades necesarias de los actuadores del robot para lograr una posición y orientación específicas [61].

La cinemática también se utiliza para controlar el equilibrio del robot mientras se mueve, y para diseñar trayectorias de movimiento para el robot que mantengan su

estabilidad. La dinámica del robot y su interacción con la base móvil o el punto de pivote también se tienen en cuenta en la cinemática de robots balancín [61].

Para poder entender la cinemática de un robot balancín se deben conocer ciertos conceptos claves como:

- **Coordenadas y Referencias:** La descripción de la cinemática de un robot de dos ruedas generalmente implica el uso de coordenadas para representar su posición y orientación. Un sistema de coordenadas puede estar ubicado en el centro del robot, y las coordenadas (x, y) representarían la posición del robot en un plano bidimensional [61].
- **Variables de Estado:** Las variables de estado clave en la cinemática de un robot de dos ruedas incluyen la posición y orientación del cuerpo del robot (generalmente representadas como la inclinación hacia adelante o hacia atrás y la inclinación lateral) y las velocidades de las ruedas izquierda y derecha [61].
- **Cinemática Directa:** La cinemática directa se utiliza para calcular la posición y la orientación del robot en función de las velocidades de las ruedas y su inclinación. Este cálculo se basa en relaciones geométricas y cinemáticas que dependen de la geometría del robot y el radio de las ruedas [61].
- **Cinemática Inversa:** La cinemática inversa se utiliza para determinar las velocidades de las ruedas requeridas para lograr una cierta posición y orientación deseada del robot. Esto es crucial para el control del robot, ya que permite calcular cómo deben ajustarse las velocidades de las ruedas para mantener el equilibrio y lograr una trayectoria deseada [61].
- **Control de Equilibrio:** El control de equilibrio es fundamental en un robot de dos ruedas. Esto implica ajustar continuamente las velocidades de las ruedas para mantener el robot en posición vertical. Se utilizan sensores como giroscopios y acelerómetros para medir la inclinación y los algoritmos de control para generar las señales de control adecuadas [61].
- **Planificación de Trayectorias:** Para mover el robot de un lugar a otro, se debe realizar una planificación de trayectorias que tenga en cuenta las restricciones cinemáticas y las limitaciones de velocidad del robot. Esto

implica calcular cómo debe cambiar la velocidad de las ruedas para seguir una trayectoria deseada [61].

2.2.17. Elementos del robot balancín

El presente trabajo tiene como base un sistema electromecánico de dos ruedas, el cual permitirá el desarrollo del trabajo propuesto, comprender cuales son las propiedades teóricas de los elementos que se utilizaran para ejecutar el proceso propuesto.

Componentes físicos

Un sistema físico electromecánico que controla el equilibrio está formado por varios elementos electrónicos y electromecánicos que ejecutan determinadas funciones dentro del sistema a controlar, los elementos son los siguientes.

Giroscopio

Un giroscopio es un dispositivo utilizado para medir y mantener la orientación angular de un objeto. Puede ser mecánico, con un disco giratorio que resiste cambios en la orientación, o electrónico, basado en principios físicos para medir la velocidad angular sin partes móviles. Los giroscopios son esenciales en aplicaciones como la navegación aérea, la estabilización de cámaras y la detección de movimiento en dispositivos electrónicos como teléfonos inteligentes. Su capacidad para detectar cambios en la orientación los hace cruciales en una variedad de campos, desde la aviación hasta la robótica y la navegación espacial [62].

Los giroscopios se utilizan en una amplia variedad de aplicaciones, incluyendo:

- Navegación: En aviones, barcos, submarinos y vehículos espaciales para medir la orientación y la velocidad angular [62].
- Estabilización de Imagen: En cámaras y videocámaras para mantener las imágenes estables y evitar la vibración y el movimiento no deseado [62].
- Electrónica de Consumo: En dispositivos móviles como teléfonos inteligentes y tabletas para detectar la orientación y el movimiento del dispositivo [62].
- Robótica: En robots y drones para mejorar la capacidad de navegación y estabilización [62].
- Realidad Virtual y Videojuegos: En sistemas de realidad virtual y controladores de videojuegos para detectar el movimiento y la orientación de los dispositivos [62].

- Instrumentación Científica: En laboratorios y experimentos científicos que requieren mediciones precisas de la velocidad angular y la orientación [62].

Encoders

Los encoders son dispositivos que se utilizan para medir la rotación, la velocidad y la dirección de un eje en un sistema mecánico o electromecánico. Algunas de las propiedades de los encoders son: alta precisión, resolución, velocidad de respuesta rápida, lo que los hace ideales para una amplia variedad de aplicaciones que requieren mediciones precisas y en tiempo real del eje en su posición, velocidad y dirección de rotación en un sistema mecánico o electromecánico [63].

Aplicaciones comunes de los encoders incluyen:

- Control de motores: Para medir la velocidad y la posición de los ejes de los motores en sistemas de automatización industrial, impresoras, CNC, etc [63].
- Robótica: Para proporcionar retroalimentación de posición a robots industriales y robots móviles [63].
- Dispositivos de medición y posicionamiento: En aplicaciones de medición de precisión y sistemas de posicionamiento [63].
- Electrónica de consumo: En dispositivos como mandos a distancia, controles de videojuegos y pantallas táctiles [63].
- Navegación: En vehículos autónomos y sistemas de navegación para medir la dirección y el desplazamiento [63].
- Instrumentación: En equipos de laboratorio y sistemas de medición para obtener datos precisos [63].

Puente H

El puente H es un circuito electrónico utilizado para controlar la dirección y la velocidad de los motores eléctricos en sistemas de control de movimiento. Algunas de las propiedades del puente H son: control bidireccional, control de velocidad, alta eficiencia, protección del circuito, configuraciones flexibles [64].

Un puente H permite las siguientes operaciones:

- Control de dirección del motor: Puede invertir la dirección del motor al cambiar la polaridad de la corriente que fluye a través de él. Cuando los interruptores están dispuestos de cierta manera, el motor gira en una

dirección, y cuando se cambian a otra configuración, el motor gira en la dirección opuesta [64].

- Control de velocidad: Al modular la anchura de pulso (PWM) de la señal de control enviada a los interruptores del puente H, es posible controlar la velocidad del motor. Cuanto mayor sea el ciclo de trabajo del PWM, mayor será la velocidad del motor, y viceversa [64].

El puente H se utiliza en una variedad de aplicaciones, como en vehículos eléctricos, robots, sistemas de posicionamiento, y en cualquier situación en la que se requiera un control preciso y bidireccional de un motor DC. También se utiliza en circuitos integrados dedicados llamados "controladores de puente H" que simplifican la implementación del circuito y proporcionan características de protección adicionales para el motor y el circuito de control.

Microcontrolador

Un microcontrolador es un dispositivo integrado que incluye un microprocesador, memoria y periféricos de entrada/salida en un solo chip, son de pequeño tamaño y están diseñados para ser utilizados en dispositivos y sistemas integrados, están diseñados para funcionar con bajo consumo de energía haciéndolos ideales para sistemas que requieren largas duraciones de batería, tienen integrados periféricos de entrada/salida, como puertos seriales, puertos de entrada analógica y puertos de salida digital, lo que los hace ideales para controlar dispositivos electrónicos [65].

Las características y funciones clave de un microcontrolador se detallan a continuación:

- Procesador Central (CPU): El núcleo de un microcontrolador es su unidad central de procesamiento (CPU). La CPU ejecuta instrucciones de software y controla todas las operaciones del microcontrolador.
- Memoria: Un microcontrolador típicamente incluye dos tipos de memoria:
 - Memoria de programa (Flash o ROM): Almacena el código de programa que se ejecuta en el microcontrolador. El firmware o software del sistema se almacena aquí y se utiliza para controlar las operaciones del dispositivo.

- Memoria de datos (RAM): Se utiliza para almacenar datos temporales y variables utilizadas durante la ejecución del programa. Es volátil y se borra cuando se apaga el microcontrolador.
- Periféricos de Entrada/Salida (I/O): Los microcontroladores tienen pines o puertos de entrada/salida que permiten la comunicación con el mundo exterior. Estos periféricos pueden incluir pines digitales, pines analógicos, UART, SPI, I2C y otros protocolos de comunicación.
- Temporizadores y Contadores: Los microcontroladores incluyen temporizadores y contadores que permiten medir el tiempo, generar señales de temporización y controlar eventos en tiempo real.
- Convertidores Analógico-Digital (ADC) y Digital-Analógico (DAC): Algunos microcontroladores incluyen ADC para convertir señales analógicas en señales digitales y DAC para realizar la conversión inversa.
- Puertos de Comunicación: Algunos microcontroladores tienen puertos de comunicación integrados, como UART (Universal Asynchronous Receiver/Transmitter), SPI (Serial Peripheral Interface) e I2C (Inter-Integrated Circuit), que permiten la comunicación con otros dispositivos.
- Oscilador y Reloj: Los microcontroladores tienen un oscilador o generador de reloj interno que establece la velocidad de ejecución del programa.
- Interfaces de Energía: Incluyen reguladores de voltaje y circuitos de administración de energía que permiten que el microcontrolador funcione con una fuente de alimentación específica.

Motores DC

Los motores DC (corriente continua) son dispositivos electromecánicos que producen movimiento mecánico a partir de energía eléctrica. Algunas de las propiedades de los motores DC son: control de velocidad, control de dirección, alta eficiencia, Amplia gama de tamaños y potencias [64].

A continuación, se describen las características y componentes principales de los motores DC:

Componentes Principales de un Motor DC:

Rotor: El rotor es la parte giratoria del motor. Puede estar hecho de un núcleo magnético y bobinas enrolladas (rotor de devanado) o un imán permanente (rotor de imán permanente). La interacción entre el rotor y el campo magnético crea el movimiento rotativo [64].

Estator: El estator es la parte estacionaria del motor y contiene el campo magnético principal. Puede consistir en imanes permanentes o bobinas de alambre enrolladas que se magnetizan cuando se les suministra corriente [64].

Conmutador (en motores de escobillas): En los motores DC con escobillas, el conmutador es un interruptor rotativo que invierte la dirección de la corriente en las bobinas del rotor a medida que gira, lo que permite que el motor gire en una dirección continua [64].

Escobillas: En los motores DC con escobillas, las escobillas son piezas de material conductor (a menudo carbón) que hacen contacto eléctrico con el conmutador y permiten que la corriente fluya hacia el rotor. Las escobillas también son responsables de invertir la dirección de la corriente en el conmutador para cambiar la dirección de rotación del motor [64].

Características y Funcionamiento:

Control de Velocidad: Los motores DC son conocidos por su capacidad de control de velocidad preciso. Al variar la tensión o la corriente suministrada al motor, se puede controlar la velocidad de giro. Esto los hace ideales para aplicaciones que requieren ajustes de velocidad, como ventiladores, cintas transportadoras y controladores de movimiento [64].

Sentido de Giro Reversible: Cambiar la polaridad de la corriente a través del motor DC permite invertir su dirección de giro. Esto es útil en aplicaciones donde se requiere movimiento en ambas direcciones, como en vehículos eléctricos y robots [64].

Arranque y Paro Suaves: Los motores DC pueden arrancar y detenerse suavemente, lo que reduce el desgaste mecánico y la posibilidad de golpes bruscos al iniciar o detener el motor [64].

Alto Par de Arranque: Los motores DC pueden proporcionar un alto par de arranque, lo que los hace adecuados para aplicaciones que requieren superar la resistencia inicial, como en ascensores y grúas [64].

Escobillas y Mantenimiento: Los motores DC con escobillas requieren mantenimiento periódico debido al desgaste natural de las escobillas. Los motores sin escobillas (BLDC) o motores de corriente continua sin escobillas son una alternativa que elimina esta necesidad de mantenimiento [64].

Componentes lógicos

Los componentes lógicos, en el contexto de la electrónica y la informática, son dispositivos o elementos que realizan funciones lógicas en circuitos digitales. Estos componentes son fundamentales en la construcción de sistemas digitales, como computadoras, calculadoras, sistemas de control, y otros dispositivos electrónicos que procesan información en forma binaria (0 y 1) o con niveles de voltaje distintos que representan estados lógicos (alto y bajo) [66].

Matlab - Simulink

La interfaz gráfica de Simulink se compone de bloques funcionales y se utiliza para construir modelos de sistemas [67]. Cada bloque representa una función específica y se conectan entre sí mediante líneas que indican la transferencia de señales o datos.

Características Clave:

- Ofrece una amplia variedad de funciones y herramientas para cálculos numéricos y simbólicos [67].
- Proporciona un entorno interactivo para escribir y ejecutar scripts y funciones [67].
- Es ampliamente utilizado en campos como matemáticas, física, ingeniería, ciencias de la computación, economía y más [67].
- Permite la manipulación de matrices y vectores de manera eficiente.
- Ofrece bibliotecas y herramientas para realizar análisis de datos y visualización de resultados [67].

La simulación en Simulink se basa en el método del dominio del tiempo, lo que significa que se pueden simular sistemas en tiempo real. Los modelos se construyen utilizando ecuaciones matemáticas y algoritmos, y se pueden simular para analizar su comportamiento en diferentes escenarios [68].

Características Clave:

- Utiliza un enfoque basado en diagramas de bloques para modelar sistemas dinámicos [68].
- Permite la simulación de sistemas en tiempo continuo y discreto [68].

- Se utiliza ampliamente en la industria para el diseño y la verificación de controladores y sistemas embebidos [68].
- Ofrece una amplia biblioteca de bloques predefinidos que representan componentes de sistemas físicos y de control [68].
- Es compatible con la generación de código automático para implementar algoritmos de control en hardware real [68].

Comunicación UART

UART (Universal Asynchronous Receiver/Transmitter) es un protocolo de comunicación serie utilizado para la transmisión de datos entre dispositivos electrónicos. (“Microcontroladores: Tipos Y Aplicaciones En La Automatización.”) Algunas de las propiedades de la comunicación UART son: asincronía, simpleza, velocidad de transmisión y tolerancia al ruido [69].

Las características principales de la comunicación UART son las siguientes:

- **Asincronía:** En la comunicación UART, los datos se transmiten de forma asincrónica, lo que significa que no se utiliza un reloj compartido entre el transmisor y el receptor para sincronizar la comunicación. En su lugar, se utiliza una velocidad de baudios (baud rate) predefinida para indicar la velocidad a la que se transmiten los datos. Tanto el transmisor como el receptor deben configurarse con la misma velocidad de baudios para una comunicación exitosa.
- **Dos Hilos (Cables):** La comunicación UART requiere al menos dos hilos (cables) para transmitir datos:
 - **Tx (Transmit):** Este hilo se utiliza para enviar datos desde el dispositivo transmisor al dispositivo receptor.
 - **Rx (Receive):** Este hilo se utiliza para recibir datos en el dispositivo receptor.
- **Estructura de Marco de Datos:** Cada paquete de datos transmitido a través de UART sigue una estructura de marco de datos, que suele incluir los siguientes elementos:
 - **Bit de Inicio (Start Bit):** Indica el inicio del marco de datos.
 - **Bits de Datos (Data Bits):** Los datos que se están transmitiendo (generalmente de 8 bits, pero puede variar).

- Bit de Paridad (Parity Bit) (opcional): Se utiliza para verificar la integridad de los datos.
- Bits de Parada (Stop Bits): Indican el final del marco de datos.
- Full-Duplex: La comunicación UART es full-duplex, lo que significa que permite la transmisión y la recepción de datos al mismo tiempo. Esto se logra utilizando dos hilos separados para transmitir y recibir datos.
- Configuración de Velocidad de Baudios: La velocidad de baudios (baud rate) se configura para determinar la velocidad de transferencia de datos. Las velocidades de baudios comunes incluyen 9600, 115200, 57600, etc. Es importante que tanto el transmisor como el receptor estén configurados con la misma velocidad de baudios para que la comunicación sea efectiva.

Arduino

Una plataforma de hardware y software de código abierto llamada Arduino permite a los usuarios crear proyectos electrónicos y de robótica de manera accesible y amigable para principiantes y profesionales por igual. El software de Arduino, conocido como "Arduino IDE" (Entorno de Desarrollo Integrado de Arduino), es una herramienta esencial para programar y cargar código en las placas Arduino [70].

A continuación, se presentan las características clave del software de Arduino:

Características Principales:

- Editor de Código: Arduino IDE proporciona un editor de código en el que puedes escribir y editar programas en el lenguaje de programación de Arduino. Utiliza una sintaxis similar a C/C++ [70].
- Compilador: El software incluye un compilador que convierte tu código escrito en lenguaje de programación de Arduino en un lenguaje de máquina que puede entender la placa Arduino [70].
- Librerías: Arduino IDE incluye una serie de librerías predefinidas que simplifican la programación de tareas comunes, como la comunicación serial, la lectura de sensores y el control de actuadores. Puedes utilizar estas librerías para ahorrar tiempo en la escritura de código [70].
- Monitor Serial: Arduino IDE tiene una ventana de monitor serial que te permite comunicarte con la placa Arduino a través de la conexión USB.

Puedes enviar datos al Arduino y ver las respuestas que envía de vuelta [70].

- Cargar Código: Puedes cargar tu programa en la placa Arduino conectándola a tu computadora a través de un cable USB y utilizando la función "Cargar" en el IDE [70].
- Depuración: Aunque el soporte de depuración en Arduino IDE es limitado en comparación con algunas otras herramientas de desarrollo, puedes usar declaraciones de impresión (por ejemplo, `Serial.print()`) para rastrear el comportamiento de tu programa [70].
- Compatibilidad con Diversos Modelos de Arduino: El IDE es compatible con una variedad de placas Arduino, incluyendo Arduino Uno, Arduino Nano, Arduino Mega, Arduino Due, entre otras [70].

2.3. MARCO TEORICO

Para esta sección del documento se analizaron diferentes documentos de investigación científica que son bases para la elaboración de la propuesta tecnológica y permitieron el desarrollo de este tema en particular.

El tema “diseño de controlador lqr para helicópteros de cuatro rotores” realizado por Emmanuel Okyere, Amar Bousbaine, Gwangtim T. Poyi, Ajay K. Joseph, Jose M. Andrade en la revista El diario de ingeniería en 2019, muestra el análisis y desempeño que obtuvo del algoritmo de control LQR en un helicóptero quadrotor pudo cumplir con los parámetros de rendimiento basándose en el modelo analítico para dicho sistema con las respectivas pruebas, el modelo presento inconvenientes al tener valor bajos en la matriz de ganancia [71].

El tema “control de velocidad y aceleración para un robot de dos ruedas y patas basado en modelo dinámico distribuido y control de cuerpo completo” realizado por Yaxian Xin, Hui Chai, Yibin Li, Xuwen Rong, Bin Li; Yueyang Li en IEEE 2019, utilizaron un método de modelado dinámico de cuerpo completo para un robot de dos ruedas y patas, con el cual llegaron a realizar un controlador que mantiene la altura y ángulo deseado, permitiendo una locomoción compatible y robusta. Los errores en este trabajo son muy pequeños con lo cual demuestra la excelente calidad del controlador [72].

En el artículo científico de tema “implementación y control de un robot bípedo con ruedas utilizando un enfoque de logic difusa” realizado por Chun Fei Hsu, Bo Rui Chen y Zi-Ling Lin de la revista científica ACTUATORS, diseñaron e implementaron un robot bípedo con ruedas en el que combinaron las dos destrezas de los robots, en el que se utilizó un control inteligente de movimiento y equilibrio debido a la naturaleza de este robot el cual es altamente no lineal, dinámica e inestable. Los parámetros de control se ajustaron utilizando aspectos cualitativos de los conocimientos humanos [73].

En el artículo científico de la revista IEEE SPECTRUM con el título “un robot para el peor trabajo en el almacén: stretch de boston dynamics puede mover 800 cajas pesadas por hora” realizado por Evan Ackermann, destaca la importancia y necesidad que tienen las industrias al mantenerse a la vanguardia y con la pandemia de Covid-19 esta necesidad aumento, por ende, adquirir elementos robóticos que ayuden con las operaciones de logística de estas empresas son necesarios. Las nuevas generaciones de robots son flexibles e inteligentes para manejar diferentes entornos industriales [74].

CAPÍTULO III

En este capítulo se trata de las técnicas y elementos empleados en el proyecto de titulación, se realiza la explicación y funcionamiento de cada uno de ellos y la importancia que tienen en el trabajo.

3.1. Componentes de la propuesta

Se hace alusión a los componentes físicos y lógicos utilizados para el desarrollo del proyecto, indicando las características relevantes y el funcionamiento de estos.

Componentes físicos

Arduino Uno

El Arduino Uno es una placa de desarrollo altamente popular que se ha convertido en una herramienta fundamental para entusiastas, estudiantes y profesionales de la electrónica y la programación. Equipado con un microcontrolador ATmega328P, el Arduino Uno ofrece 14 pines digitales y 6 pines analógicos que permiten la interacción con una amplia variedad de componentes y sensores. Su facilidad de programación a través del entorno de desarrollo de Arduino, que se basa en un lenguaje de programación simplificado, lo hace accesible incluso para principiantes. Además, su comunidad activa y la abundancia de recursos, incluyendo librerías y ejemplos de código, fomentan la creación de proyectos desde simples experimentos hasta complejas aplicaciones de robótica, automatización y IoT [75].

El Arduino Uno es conocido por su versatilidad y su capacidad para democratizar el mundo de la electrónica y la automatización, proporcionando una plataforma abierta y asequible que fomenta la creatividad y la innovación en el ámbito de la electrónica y la programación [75].

MPU 6050

El MPU-6050 es un sensor versátil de seis ejes que combina un giroscopio de tres ejes y un acelerómetro de tres ejes en un solo dispositivo compacto. Este sensor es utilizado en una amplia variedad de aplicaciones, desde robótica y drones hasta dispositivos portátiles y sistemas de control de movimiento. Su capacidad para medir la velocidad angular y la aceleración en tres direcciones permite determinar con precisión la orientación y el movimiento de un objeto en el espacio tridimensional. El MPU-6050 se comunica a través del protocolo I2C y proporciona datos digitales, lo que facilita su

integración en sistemas electrónicos. Además, su bajo consumo de energía lo hace adecuado para aplicaciones alimentadas por batería [76].

Este sensor es especialmente útil en proyectos que requieren la detección de movimiento y la orientación, y su versatilidad lo convierte en una herramienta valiosa para diseñadores y desarrolladores que buscan agregar capacidades de control y navegación a sus aplicaciones y dispositivos. La capacidad de programar interrupciones y su capacidad de calibración lo hacen aún más atractivo para aplicaciones que exigen mediciones precisas en tiempo real [76].

Puente H TB6612FNG

El TB6612FNG es un controlador de motor dual conocido como puente H utilizado en proyectos de electrónica y robótica para controlar motores de corriente continua (DC) o motores paso a paso. Este dispositivo ofrece la capacidad de controlar dos motores de manera independiente, permitiendo la inversión de la polaridad de la corriente para controlar la dirección de giro de cada motor. Además, proporciona protección contra sobre corriente y sobrecalentamiento, lo que garantiza la seguridad y la integridad del controlador y los motores en condiciones adversas [77].

El TB6612FNG es ampliamente utilizado en una variedad de aplicaciones, desde robots móviles y vehículos controlados por radio hasta impresoras 3D y sistemas de automatización. Su diseño compacto, amplio rango de tensiones de alimentación y facilidad de integración lo hacen una elección versátil para proyectos que requieren control preciso y bidireccional de motores de corriente continua [77].

Motor "Hytech GA37-520

El motor "Hytech GA37-520" es una pieza esencial en proyectos de electrónica y robótica que requieren precisión en la velocidad y dirección del motor. Funciona con un voltaje de 12 voltios y tiene una velocidad nominal de 330 RPM, lo que lo hace adecuado para aplicaciones que necesitan un control suave y constante de la velocidad de giro. Su caja de engranajes de metal y su eje del motor de 6 mm lo convierten en una opción robusta y duradera, capaz de soportar cargas de trabajo considerables. Además, la inclusión de un codificador de velocidad agrega una capa de precisión al permitir la medición y el control exactos de la velocidad, lo que es particularmente valioso en proyectos de vehículos robóticos DIY basados en Arduino.

El motor "Hytech GA37-520" se integra fácilmente en kits de chasis de robots inteligentes y proyectos de bricolaje, donde su calidad de construcción y su potencia

nominal de aproximadamente 7.2W destacan. Con dimensiones compactas, una longitud de 58 mm y un diámetro máximo de 37 mm, este motor ofrece un equilibrio entre rendimiento y tamaño, lo que lo hace ideal para una amplia gama de aplicaciones de automatización y robótica. Su facilidad de uso y capacidad de control de velocidad lo convierten en una elección popular para entusiastas y diseñadores que buscan crear vehículos robóticos personalizados y otros dispositivos controlados electrónicamente.

Componentes lógicos

Simulink- Matlab

La utilización de Simulink-Matlab para desarrollar el modelo matemático de un robot balancín en espacio de estados es una elección poderosa y altamente efectiva. Simulink es una herramienta de modelado y simulación que ofrece un entorno gráfico intuitivo para la creación de modelos de sistemas dinámicos complejos, como robots. Al aprovechar la capacidad de Simulink para representar sistemas en espacio de estados, se describe las dinámicas del robot balancín de una manera más precisa y comprensible. Esto implica definir las ecuaciones que describen cómo las variables de estado evolucionan en el tiempo, lo que facilita el análisis y la simulación de su comportamiento bajo diferentes condiciones y controladores.

Además, la integración de Matlab permite aprovechar su potencia computacional y herramientas avanzadas de análisis numérico para validar y optimizar el modelo. Esto es especialmente valioso al diseñar controladores y estrategias de control para el robot balancín, debido a que se puede realizar pruebas virtuales antes de la implementación física.

3.2. Diseño de la propuesta

En este ítem se abordan temas como la identificación de la planta, identificación de variables, modelo matemático, métodos de linealización y otros que permitirán resolver de manera adecuada el tema propuesto para este trabajo de titulación.

3.2.1. Sistema electromecánico e identificación de variables

Un robot de la marca Elegoo Tumbler Self- Balancing, es el sistema donde se desarrolla y aplica el controlador LQR, este sistema está conformado por diferentes parámetros que actúan sobre él, las cuales son representadas por ecuaciones que satisfagan las propiedades de cada una de ellas, para ello es necesario representar las variables que ayudaran a comprenderlas, como se muestra en la tabla 1.

Tabla 1. Representación de las variables a utilizar

Variable	Nombre
E_{c_1}	Energía cinética de traslación en la rueda
E_{c_2}	Energía cinética de rotación en la rueda
E_{c_3}	Energía cinética de traslación en el péndulo
E_{c_4}	Energía cinética de rotación en el péndulo
E_p	Energía potencial en el péndulo
m_r	Masa de las ruedas
m_p	Masa del péndulo
J_r	Momento de inercia en las ruedas
J_p	Momento de inercia en el péndulo
l_{cg}	Distancia entre el eje de la rueda y el centro de gravedad del péndulo
x_{cm}	Distancia entre el centro de masa del péndulo y la horizontal
y_{cm}	Distancia entre el centro de masa del péndulo y la vertical.
G	Gravedad
φ	Angulo de la rueda
$\dot{\varphi}$	Velocidad angular de la rueda
θ	Angulo del péndulo respecto a la vertical
$\dot{\theta}$	Velocidad angular del péndulo en relación con la vertical
$\ddot{\theta}$	Aceleración angular del péndulo en relación con la vertical
x	Posición del eje
\dot{x}	Velocidad del eje
\ddot{x}	Aceleración del eje

3.2.2. Identificación de la planta

Una vez que los valores son ingresados con intervalos de tiempo de 0.005 segundos, se producen un total de 551 muestras, las cuales serán empleadas en la identificación de la planta, en la figura 7 se muestra la recopilación de datos de las señales de entrada y salida del sistema.

Se procede a emplear la opción de “System Identification” la cual se encuentra dentro de la herramienta de “APPS” del entorno gráfico de Matlab, para poder determinar el diseño en espacio de estados del sistema. En la figura 8 se observa la

entrada y salida del sistema, μ_1 ser la señal de entrada del sistema y y_1 es la señal de salida del sistema.

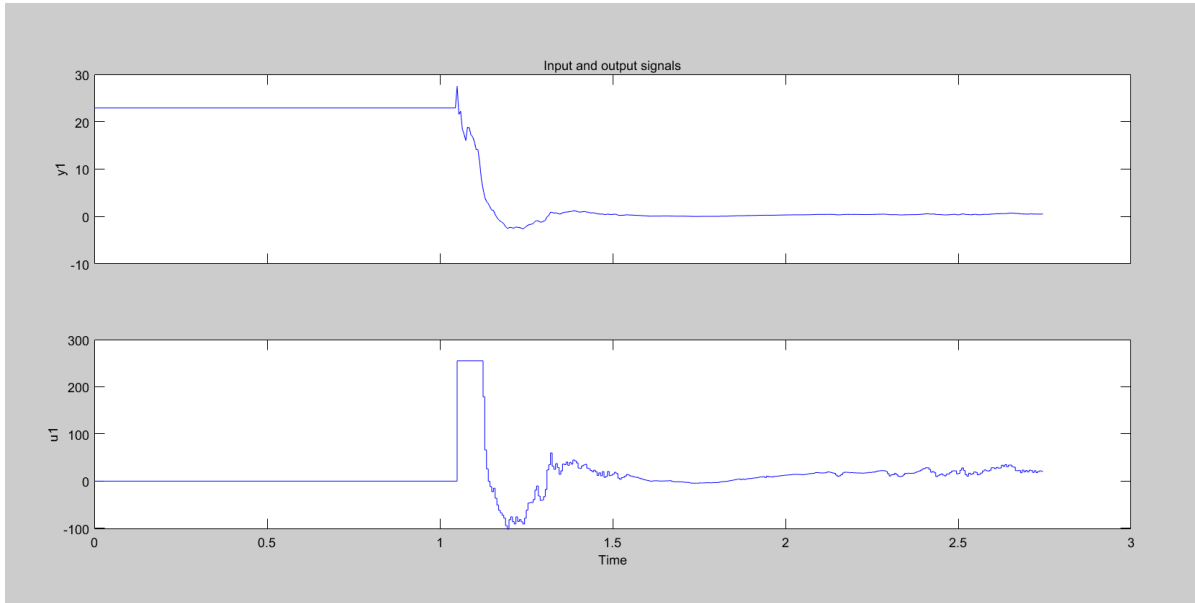


Figura 9. Diagrama en lazo abierto del sistema.

Se procede a realizar múltiples simulaciones con diferentes aproximaciones del modelo en espacio de estado del sistema, se realizaron un total de 6 simulaciones de las cuales se selecciona la señal con la respuesta más alta de todas, esta señal se la tomará para la realización del algoritmo de control en cuestión, en la figura 9 se aprecia las simulaciones realizadas con la estimación más alta.

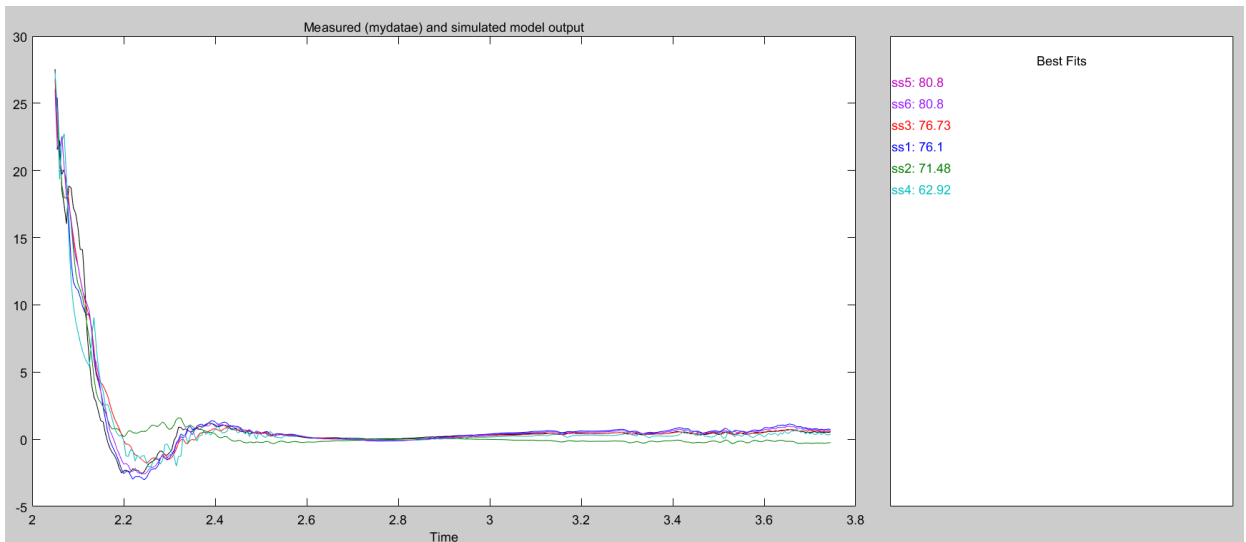


Figura 10. Estimaciones del mejor modelado matemático del sistema.

Para poder observar las aproximaciones de una manera más clara se procede a realizar la tabla 2, donde se muestran las respuestas de los estados evaluados en la

planta del robot balancín, definiendo de esta manera cuál de estas señales es la óptima para desarrollar el control.

Tabla 2. Respuesta de los estados evaluados.

Modelo	Estimación	Tipo	Orden	Aproximación
SS1	State Space Model	Espacio de estado en tiempo continuo	Cuarto orden	76.1%
SS2	State Space Model	Espacio de estado en tiempo continuo	Cuarto orden	71.48%
SS3	State Space Model	Espacio de estado en tiempo continuo	Cuarto orden	76.73%
SS4	State Space Model	Espacio de estado en tiempo continuo	Cuarto orden	62.92%
SS5	State Space Model	Espacio de estado en tiempo continuo	Cuarto orden	80.8%
SS6	State Space Model	Espacio de estado en tiempo discreto	Cuarto orden	80.8%

En esta tabla se puede apreciar seis expresiones del modelo matemático, las cuales fueron generadas por medio de estimaciones aplicadas a contextos continuos y discretos, se consideró que los modelos fueran de orden 4 debido a que este se acerca más a la salida requerida, el modelo seleccionado es “SS6” el cual corresponde a un modelo en espacio de estado en tiempo discreto de cuarto orden el cual tiene una aproximación de 80.8% como función del sistema, el modelo seleccionado se utilizara para el desarrollo de los algoritmos de control del LQR.

Tomando como referencia las ecuaciones en espacio de estado se reemplaza las matrices obtenidas de la simulación

Tabla 3. Valores de la matriz A

	x_1	x_2	x_3	x_4
x_1	0.8183	-0.1712	0.004428	-0.003131
x_2	0.09972	0.1822	-0.7157	0.6702
x_3	-0.02655	-0.162	0.5857	0.5184
x_4	0.08992	-0.5339	-0.3448	-0.2894

$$A = \begin{bmatrix} 0.8183 & -0.1712 & 0.004428 & -0.003131 \\ 0.09972 & 0.1822 & -0.7157 & 0.6702 \\ -0.02655 & -0.162 & 0.5857 & 0.5184 \\ 0.08992 & -0.5339 & -0.3448 & -0.2894 \end{bmatrix}$$

Tabla 4. Valores de la matriz B.

	u_1
x_1	0.002859
x_2	0.01812
x_3	0.008265
x_4	0.0006789

$$B = \begin{bmatrix} 0.002859 \\ 0.01812 \\ 0.008265 \\ 0.0006789 \end{bmatrix}$$

Tabla 5. Valores de la matriz C.

	x_1	x_2	x_3	x_4
y_1	5.286	-0.1187	0.1383	-0.2147

$$C = [5.286 \quad -0.1187 \quad 0.1383 \quad -0.2147]$$

Tabla 6. Valores de la matriz D.

	u_1
y_1	0

$$D = [0]$$

Dejando representadas las matrices en espacio de estado quedan de la siguiente manera

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0.8183 & -0.1712 & 0.004428 & -0.003131 \\ 0.09972 & 0.1822 & -0.7157 & 0.6702 \\ -0.02655 & -0.162 & 0.5857 & 0.5184 \\ 0.08992 & -0.5339 & -0.3448 & -0.2894 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0.002859 \\ 0.01812 \\ 0.008265 \\ 0.0006789 \end{bmatrix} u \quad 46$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = [5.286 \quad -0.1187 \quad 0.1383 \quad -0.2147] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + [0]u$$

3.2.3. Modelado matemático del sistema

Al modelar un sistema se debe definir la dinámica de la planta mediante ecuaciones matemáticas que permitan describir de forma teórica el comportamiento de esta, teniendo en cuenta los elementos electrónicos y mecánicos utilizados en su fabricación.

Para el modelado matemático del robot balancín se deben considerar diferentes fuerzas que interactúan sobre este sistema, una de estas fuerzas analiza el motor con el propósito de obtener el torque, viscosidad y rozamiento los cuales servirán para análisis posteriores.

Dinámica del sistema

Dentro del sistema electromecánico actúan dos energías, la cinética y la potencial, la fórmula de energía cinética está representada en la ecuación 30 y la energía potencial en la ecuación 31.

$$E_c = \frac{1}{2} m v \quad 47$$

$$E_p = mgh \quad 48$$

Se debe tener en cuenta que cada elemento en cuestión tiene masa e inercia, las cuales se representan con la letra m para la masa y J para la inercia, tanto para las ruedas como para el péndulo. Al aplicar las fórmulas de las energías antes mencionadas sobre las ruedas, estas quedarían expresadas de la siguiente manera:

$$E_{c_1} = \frac{1}{2} m_r \dot{x}^2 \quad 49$$

$$E_{c_2} = \frac{1}{2} J_r \dot{\varphi}^2 \quad 50$$

La ecuación 33 debe quedar en términos de \dot{x} , para ello es necesario hacer una equivalencia de $\varphi = v/r$, despejar v y dejar en términos de x , quedando de la siguiente manera:

$$E_{c_2} = \frac{1}{2} \frac{J_r}{r^2} \dot{x}^2 \quad 51$$

Para el péndulo las fórmulas mencionadas anteriormente quedarían de la siguiente manera:

$$E_{c_3} = \frac{1}{2} m_p \dot{x}^2 \quad 52$$

$$E_{c_4} = \frac{1}{2} J_p \dot{\theta}^2 \quad 53$$

Al momento que el robot experimenta alguna inclinación, la posición de este pasa a estar en un plano bidimensional generando una posición en x y una posición en y , las cuales serán representadas mediante una función trigonométrica, quedando de la siguiente manera:

$$x_p = x + l_{cg} \text{sen}(\theta) \quad 54$$

$$y_p = l_{cg} \text{cos}(\theta) \quad 55$$

Es por este motivo que la ecuación 35 sufrirá una ligera modificación en \dot{x}^2 , al sumar las dos posiciones representadas en las ecuaciones 37 y 38 y posterior a esto derivar el resultado se obtiene:

$$E_{c_3} = \frac{1}{2} m_p (\dot{x}^2 + \dot{\theta}^2 l^2) + m_p \dot{x} \dot{\theta} l \cos(\theta) \quad 56$$

La fórmula de energía potencial al ser adaptada al sistema utilizado también sufrirá una ligera modificación, la cual se muestra en la fórmula 40, cabe recalcar que esta fórmula solo será aplicada al péndulo, en las ruedas no es aplicable debido que en estas no existe algún tipo de inclinación, por lo tanto, la formula queda expresada de la siguiente manera:

$$E_p = m_p g l (\cos(\theta) - 1) \quad 57$$

Para poder describir el comportamiento de la planta utilizando las ecuaciones obtenidas se empleará el método Lagrangiano, este se basa en el principio de mínima acción obtenido a partir de la energía cinética y potencial del sistema quedando de la siguiente manera:

$$L = E_{c_1} + E_{c_2} + E_{c_3} + E_{c_4} - E_p \quad 58$$

Al reemplazar las ecuaciones 32, 34, 35, 39 y 40 en la ecuación 41 se obtiene el siguiente resultado:

$$L = \frac{1}{2} m_r \dot{x}^2 + \frac{1}{2} J_r \dot{x}^2 + \left(\frac{1}{2} m_p (\dot{x}^2 + \dot{\theta}^2 l^2) + m_p \dot{x} \dot{\theta} l \cos(\theta) \right) + \frac{1}{2} J_p \dot{\theta}^2 - m_p g l (\cos(\theta) - 1) \quad 59$$

Fuerzas generalizadas

En el sistema interviene una fuerza general denominada trabajo expresada en la ecuación 43, la cual será empleada en diferentes partes, las cuales permitirán obtener fricción, viscosidad y torque, las cuales son necesarias para el análisis del modelo matemático de la planta.

$$W = F \cdot d \quad 60$$

Para hallar la fuerza del torque se realiza el diferencial de este, en términos de distancia con términos de posición, obteniendo la siguiente ecuación:

$$W_{\tau_m} = \left(\Gamma_m \frac{x}{r} \right) - (\Gamma_m \theta) \quad 61$$

Al derivar la ecuación 44 en términos de x y de θ , se obtienen las siguientes ecuaciones respectivamente:

$$\frac{\partial W_{\tau_m}}{\partial x} = \Gamma_m \frac{1}{r} \quad 62$$

$$\frac{\partial W_{\tau_m}}{\partial \theta} = -\Gamma_m \quad 63$$

Para hallar la fuerza de fricción se emplea la ecuación 43 en donde $F = \mu_s * \dot{x}$ obteniendo de este modo la siguiente ecuación:

$$W_{F_s} = (\mu_s * \dot{x})(x) \quad 64$$

Se deriva esta ecuación en términos de x y de \dot{x} , se obtienen las siguientes ecuaciones respectivamente:

$$\frac{\partial W_{F_s}}{\partial x} = -\mu_s \dot{x} \quad 65$$

$$\frac{\partial W_{F_s}}{\partial \dot{x}} = -\mu_s x \quad 66$$

Para hallar la fuerza de viscosidad en las ruedas se emplea la ecuación 43 en donde $F_v = -\mu_e * v$, $v = \dot{\phi} - \dot{\theta}$ y $d = \phi - \theta$, se obtiene de este modo la siguiente ecuación:

$$W_{F_v} = -\mu_e (\dot{\phi} - \dot{\theta})(\phi - \theta) \quad 67$$

Se deriva esta ecuación en términos de x y de θ , se obtienen las siguientes ecuaciones respectivamente:

$$\frac{\partial W_{F_v}}{\partial x} = -\frac{\mu_e}{r} (\dot{\phi} - \dot{\theta}) \quad 68$$

$$\frac{\partial W_{F_v}}{\partial \theta} = \mu_e (\dot{\phi} - \dot{\theta}) \quad 69$$

Modelo del motor

Como la planta es de un sistema electromecánico se analiza la parte interna del motor para hallar el torque en este, para ello es necesario conocer el circuito interno del motor el cual se aprecia en la figura 4.

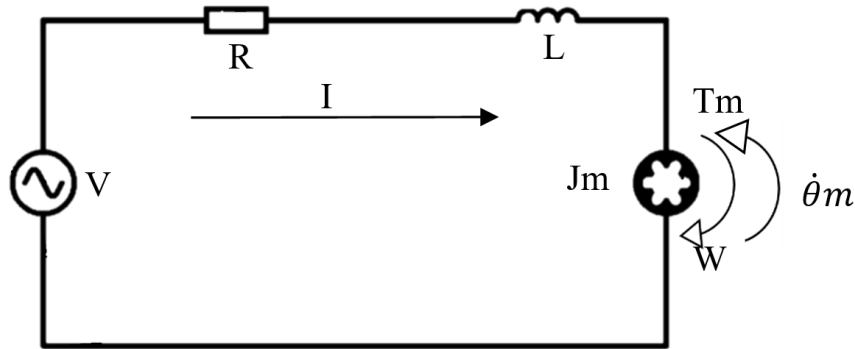


Figura 11. Circuito interno de un motor.

A este circuito conformado por resistencia, inductor, motor y un voltaje, se le aplica LVK para hallar la corriente, quedando la siguiente ecuación:

$$-v + Ri + L \frac{di}{dt} + E_a(t) = 0 \quad 70$$

Donde:

v : voltaje del motor

R : resistencia del motor

i : corriente del motor

L : inductancia del motor

E_a : voltaje interno generado del motor

Existen dos ecuaciones electromotrices ya establecidas que permitirán relacionar E_a y Γ_m con las ecuaciones anteriores, estas son:

$$E_a = k_e \dot{\theta}(t) \quad 71$$

$$\Gamma_m = k_t i(t) \quad 72$$

Reemplazando las ecuaciones 54 y 55 en 53 y despejando la corriente se obtiene:

$$i = \frac{v - k_e \dot{\theta}(t)}{R} \quad 73$$

Igualando la ecuación 56 en la ecuación 55 se obtiene la siguiente ecuación:

$$\Gamma_m(t) = \frac{vk_t - k_e k_t \dot{\theta}(t)}{R} \quad 74$$

De la ecuación 57 de torque se tiene una variable v la cual puede ser reemplazada con $k_s u$ y $\dot{\theta}(t)$ será reemplazada con $\dot{\varphi} - \dot{\theta}$, $\dot{\varphi} = \dot{x}/r$ para dejar todo expresado en términos de las variables correspondientes, obteniendo de esta forma la nueva ecuación de torque

$$\Gamma_m(t) = \frac{k_s k_t}{R} u - \frac{-k_e k_t}{R} \left(\frac{\dot{x}}{r} - \dot{\theta} \right) \quad 75$$

Euler-Lagrange

Para poder hallar las ecuaciones de movimiento del sistema físico que se está analizando es fundamental aplicar la ecuación de Euler-Lagrange expresada matemáticamente en la siguiente ecuación:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \sum Q_i \quad 76$$

Esta ecuación indica que se debe realizar una derivada con respecto al tiempo a la derivada parcial de Lagrange con respecto a la segunda derivada de q_i , menos la derivada de Lagrange con respecto a q_i , todo esto igualado a las fuerzas que actúan sobre el objeto de estudio Q_i [78].

Las fuerzas que actúan sobre este objeto están dadas en dos direcciones perpendiculares entre sí que son x y θ , los cuales son componentes del principio fundamental de la dinámica. Al modificar la ecuación 59 con las variables que se han definido para este trabajo queda de la siguiente manera:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial L}{\partial x} = \sum Q_x \quad 77$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = \sum Q_\theta \quad 78$$

Para realizar un análisis menos complejo, se realiza las derivadas parciales de Lagrange para las variables \dot{x} y x respectivamente con la ecuación 42

$$\frac{\partial}{\partial \dot{x}} \left(\frac{1}{2} m_r \dot{x}^2 + \frac{1}{2} J_r \dot{\theta}^2 + \left(\frac{1}{2} m_p (\dot{x}^2 + \dot{\theta}^2 l^2) + m_p \dot{x} \dot{\theta} l \cos(\theta) \right) + \frac{1}{2} J_p \dot{\theta}^2 - m_p g l (\cos(\theta) - 1) \right) \quad 79$$

$$\frac{\partial}{\partial x} \left(\frac{1}{2} m_r \dot{x}^2 + \frac{1}{2} J_r \dot{\theta}^2 + \left(\frac{1}{2} m_p (\dot{x}^2 + \dot{\theta}^2 l^2) + m_p \dot{x} \dot{\theta} l \cos(\theta) \right) + \frac{1}{2} J_p \dot{\theta}^2 - m_p g l (\cos(\theta) - 1) \right) \quad 80$$

Al derivar las ecuaciones 62 y 63 respectivamente, quedarían de la siguiente manera:

$$m_r \dot{x} + \frac{J_r}{r^2} \dot{x} + m_p \dot{\theta} l \cos(\theta) \quad 81$$

$$0 \quad 82$$

De la misma manera se realiza las derivadas parciales de Lagrange para las variables $\dot{\theta}$ y θ respectivamente con la ecuación 42

$$\frac{\partial}{\partial \theta} \left(\frac{1}{2} m_r \dot{x}^2 + \frac{1}{2} J_r \dot{\theta}^2 + \left(\frac{1}{2} m_p (\dot{x}^2 + \dot{\theta}^2 l^2) + m_p \dot{x} \dot{\theta} l \cos(\theta) \right) + \frac{1}{2} J_p \dot{\theta}^2 - m_p g l (\cos(\theta) - 1) \right) \quad 83$$

$$\frac{\partial}{\partial \theta} \left(\frac{1}{2} m_r \dot{x}^2 + \frac{1}{2} J_r \dot{\theta}^2 + \left(\frac{1}{2} m_p (\dot{x}^2 + \dot{\theta}^2 l^2) + m_p \dot{x} \dot{\theta} l \cos(\theta) \right) + \frac{1}{2} J_p \dot{\theta}^2 - m_p g l (\cos(\theta) - 1) \right) \quad 84$$

Al derivar las ecuaciones 66 y 67 respectivamente, el resultado será el siguiente:

$$m_p \dot{x} l \cos(\theta) + m_p \dot{\theta} l^2 + J_p \dot{\theta} \quad 85$$

$$m_p g l \sin(\theta) \quad 86$$

Una vez obtenidas las derivadas parciales para cada variable se procede a derivar las ecuaciones 64 y 68 con respecto al tiempo como se aprecia en las siguientes ecuaciones:

$$\frac{\partial}{\partial t} \left(m_r \dot{x} + \frac{J_r}{r^2} \dot{x} + m_p \dot{\theta} l \cos(\theta) \right) \quad 87$$

$$\frac{\partial}{\partial t} \left(m_p \dot{x} l \cos(\theta) + m_p \dot{\theta} l^2 + J_p \dot{\theta} \right) \quad 88$$

Los resultados de estas derivadas son las siguientes

$$\ddot{x} \left(m_r + \frac{J_r}{r^2} + m_p \right) + m_p l \ddot{\theta} \cos \theta \quad 89$$

$$\ddot{\theta} (m_p l^2 + J_p) + m_p l \dot{x} \cos \theta \quad 90$$

Una vez obtenidas las ecuaciones respectivas se realiza el reemplazo de estas en las ecuaciones 60 y 61 teniendo en cuenta que la sumatoria de fuerzas satisfagan los puntos de referencia establecidos anteriormente. Para el reemplazo de la ecuación 60 se deben tomar las ecuaciones 72, 65, 51, 48 y 45.

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial L}{\partial x} = \sum Q_x$$

$$\left(\ddot{x} \left(m_r + \frac{J_r}{r^2} + m_p \right) + m_p l \ddot{\theta} \cos \theta \right) - 0 = \Gamma_m \frac{1}{r} - \mu_s \dot{x} - \frac{\mu_e \dot{x}}{r^2} - \frac{\mu_e \theta}{r} \quad 91$$

Reorganizando la ecuación 74 se obtiene la ecuación 75, despejando \ddot{x} se halla la aceleración de las ruedas reflejada en la ecuación 77.

$$\ddot{x} \left(m_r + \frac{J_r}{r^2} + m_p \right) + m_p l \ddot{\theta} \cos \theta = \frac{\Gamma_m}{r} - \dot{x} \left(\mu_s + \frac{\mu_e}{r^2} \right) + \frac{\mu_e \theta}{r} \quad 92$$

$$I = m_r + \frac{J_r}{r^2} + m_p \quad 93$$

$$\ddot{x} = \frac{\Gamma_m}{I r} - \frac{\left(\mu_s + \frac{\mu_e}{r^2} \right)}{I} \dot{x} + \frac{\mu_e}{I r} \theta - \frac{m_p l}{I} (\ddot{\theta} \cos \theta) \quad 94$$

Para el reemplazo de la ecuación 61 se deben tomar las ecuaciones 73, 69, 52 y 46.

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial L}{\partial x} = \sum Q_x$$

$$\ddot{\theta}(m_p l^2 + J_p) + m_p l \ddot{x} \cos \theta - m_p g l \text{sen}(\theta) = -\Gamma_m - \dot{x} \frac{\mu_e}{r} - \mu_e \dot{\theta} \quad 95$$

Reorganizando la ecuación 78 se obtiene la ecuación 79, despejando $\ddot{\theta}$ se halla la aceleración angular del robot reflejada en la ecuación 81.

$$\ddot{\theta}(m_p l^2 + J_p) + m_p l (\ddot{x} \cos \theta + (g) \text{sen}(\theta)) = -\Gamma_m + \mu_e \left(\frac{\dot{x}}{r} - \dot{\theta} \right) \quad 96$$

$$J = m_p l^2 + J_p \quad 97$$

$$\ddot{\theta} = \frac{-\Gamma_m}{J} + \frac{\mu_e}{J} \left(\frac{\dot{x}}{r} - \dot{\theta} \right) - \frac{m_p l}{J} (\ddot{x} \cos \theta + (g) \text{sen}(\theta)) \quad 98$$

Reemplazando la ecuación de torque 58 en las ecuaciones 77 y 81 se obtienen las ecuaciones características del sistema

$$\ddot{x} = \frac{\frac{k_s k_t}{R} u - \frac{k_e k_t}{R} \left(\frac{\dot{x}}{r} - \dot{\theta} \right)}{I r} - \frac{(\mu_s + \frac{\mu_e}{r^2})}{I} \dot{x} + \frac{\mu_e}{I r} \dot{\theta} - \frac{m_p l}{I} \cos \theta \ddot{\theta} \quad 99$$

$$\ddot{\theta} = \frac{- \left(\frac{k_s k_t}{R} u - \frac{k_e k_t}{R} \left(\frac{\dot{x}}{r} - \dot{\theta} \right) \right)}{J} + \frac{\mu_e}{J} \left(\frac{\dot{x}}{r} - \dot{\theta} \right) - \frac{m_p l}{J} \ddot{x} \cos \theta + \frac{m_p g l}{J} \text{sen}(\theta) \quad 100$$

Mediante operaciones matemáticas aplicadas a las ecuaciones anteriores quedan nuevamente reestructuradas de la siguiente manera con las que se demuestra el sistema dinámico de la planta

$$\ddot{x} = -\frac{m_p l}{I} \ddot{\theta} \cos \theta + \frac{k_s k_t}{I r R} u - \left(\frac{k_e k_t}{r^2 R} - \frac{(\mu_s r^2 \mu_e)}{r^2 I} \right) \dot{x} + \left(\frac{k_e k_t}{I r R} + \frac{\mu_e}{I r} \right) \dot{\theta} \quad 101$$

$$\ddot{\theta} = -\frac{m_p l}{J} \ddot{x} \cos \theta + \frac{m_p g l}{J} \text{sen}(\theta) - \frac{k_s k_t}{J R} u + \left(\frac{k_e k_t}{J R r} + \frac{\mu_e}{J r} \right) \dot{x} - \left(\frac{k_e k_t}{J R r} + \frac{\mu_e}{J} \right) \dot{\theta} \quad 102$$

Para poder simplificar las ecuaciones obtenidas se va a reemplazar las variables por constantes de la siguiente manera:

$$C_1 = \frac{m_p l}{I}; C_2 = \frac{k_s k_t}{I r R}; C_3 = \frac{k_e k_t}{r^2 R} - \frac{(\mu_s r^2 \mu_e)}{r^2 I}; C_4 = \frac{k_e k_t}{I r R} + \frac{\mu_e}{I r}$$

$$C_5 = \frac{m_p l}{J}; C_6 = \frac{m_p g l}{J}; C_7 = \frac{k_s k_t}{J R}; C_8 = \frac{k_e k_t}{J R r} + \frac{\mu_e}{J r}; C_9 = \frac{k_e k_t}{J R r} + \frac{\mu_e}{J}$$

Reemplazando las constantes en las ecuaciones 99 y 100 las ecuaciones quedan planteadas de la siguiente manera:

$$\ddot{x} = -C_1\ddot{\theta} \cos \theta + C_2u - C_3\dot{x} + C_4\dot{\theta} \quad 103$$

$$\ddot{\theta} = -C_5\ddot{x} \cos \theta + C_5\text{sen}(\theta) - C_6u + C_6\dot{x} - C_7\dot{\theta} \quad 104$$

En el análisis de las ecuaciones presentadas, se observa una interdependencia entre las variables. La ecuación 86 incluye una segunda derivada con respecto al tiempo de la variable angular, denotada como $\ddot{\theta}$, mientras que la ecuación 87 involucra una derivada de la posición x , representada como \dot{x} . Esta interconexión plantea un desafío en la resolución del sistema de ecuaciones.

En consecuencia, se lleva a cabo un proceso de despeje de estas ecuaciones para lograr su independencia mutua, el cual se puede apreciar en el Anexo 2. Este procedimiento busca simplificar el análisis y facilitar la solución de las ecuaciones resultantes, permitiendo una comprensión más clara y efectiva del comportamiento del sistema bajo estudio. Las ecuaciones despejadas se expresan de la siguiente manera en las ecuaciones 88 y 89:

$$\ddot{x} = -\frac{C_1 \cdot C_6 \cdot \sin \theta \cdot \cos(\theta)}{(1 - C_5 \cdot C_1 (\cos \theta)^2)} + \left(\frac{C_1 \cdot C_7 \cdot \cos(\theta) + C_2}{(1 - C_5 \cdot C_1 (\cos \theta)^2)} \right) u - \left(\frac{C_1 \cdot C_8 \cdot \cos(\theta) + C_3}{(1 - C_5 \cdot C_1 (\cos \theta)^2)} \right) \dot{x} + \left(\frac{C_1 \cdot C_9 \cos(\theta) + C_4 - C_1 C_5 (\text{sen} \theta) (\cos \theta) \dot{x}}{(1 - C_5 \cdot C_1 (\cos \theta)^2)} \right) \dot{\theta} \quad 105$$

$$\ddot{\theta} = -\left(\frac{C_2 C_5 \cos(\theta) + C_7}{(1 - C_1 C_5 (\cos \theta)^2)} \right) u + \left(\frac{C_3 C_5 \cos(\theta) + C_8 + C_5 (\text{sen} \theta) \dot{\theta}}{(1 - C_1 C_5 (\cos \theta)^2)} \right) \dot{x} - \left(\frac{C_4 C_5 \cos(\theta) + C_9}{(1 - C_1 C_5 (\cos \theta)^2)} \right) \dot{\theta} + \frac{C_6 \sin \theta}{(1 - C_1 C_5 (\cos \theta)^2)} \quad 106$$

Linealización por medio de espacio de estados.

Para comenzar con el modelado lineal del sistema mediante la técnica de espacios de estados con Jacobiano se debe considerar las ecuaciones en variable de estado, estas ecuaciones son un conjunto de ecuaciones matemáticas que se utilizan para describir el comportamiento de sistemas dinámicos, como sistemas mecánicos, eléctricos, electrónicos, entre otros. En términos generales, un sistema dinámico puede ser representado mediante un conjunto de ecuaciones diferenciales de primer orden llamadas ecuaciones de estado. Estas ecuaciones están escritas en función de las variables de estado del sistema, que son un conjunto mínimo de variables que definen completamente el estado del sistema en un instante dado.

Para sistemas lineales, estas ecuaciones se pueden expresar de manera más específica como:

$$\frac{dx}{dt} = Ax + Bu \quad 107$$

$$y(t) = Cx \quad 108$$

Para el sistema que es objeto de estudio en este trabajo las variables de estado que servirán en el proceso de linealización del sistema son los siguientes:

$$x_1 = x; x_2 = \dot{x}; x_3 = \theta; x_4 = \dot{\theta} \quad 109$$

Al derivar estas variables se obtiene lo siguiente

$$\dot{x}_1 = \dot{x}; \dot{x}_2 = \ddot{x}; \dot{x}_3 = \dot{\theta}; \dot{x}_4 = \ddot{\theta} \quad 110$$

Estas variables de estados serán reemplazadas en las ecuaciones 88 y 89 las cuales quedaran reestructuradas como se muestran en las ecuaciones 94 y 95

$$\begin{aligned} \dot{x}_2 = & -\frac{C_1 \cdot C_6 \cdot \sin x_3 \cdot \cos(x_3)}{(1 - C_5 \cdot C_1 (\cos x_3)^2)} + \left(\frac{C_1 \cdot C_7 \cdot \cos(x_3) + C_2}{(1 - C_5 \cdot C_1 (\cos x_3)^2)} \right) u - \left(\frac{C_1 \cdot C_8 \cdot \cos(x_3) + C_3}{(1 - C_5 \cdot C_1 (\cos x_3)^2)} \right) x_2 \\ & + \left(\frac{C_1 \cdot C_9 \cos(x_3) + C_4 - C_1 C_5 \operatorname{sen}(x_3) \cos(x_3) x_2}{(1 - C_5 \cdot C_1 \cos(x_3)^2)} \right) x_4 \end{aligned} \quad 111$$

$$\begin{aligned} \dot{x}_4 = & -\left(\frac{C_2 C_5 \cos(x_3) + C_7}{(1 - C_1 C_5 (\cos x_3)^2)} \right) u + \left(\frac{C_3 C_5 \cos(x_3) + C_8 + C_5 \operatorname{sen}(x_3) x_4}{(1 - C_1 C_5 (\cos \theta)^2)} \right) x_2 \\ & - \left(\frac{C_4 C_5 \cos(x_3) + C_9}{(1 - C_1 C_5 \cos(x_3)^2)} \right) x_4 + \frac{C_6 \sin(x_3)}{(1 - C_1 C_5 \cos(x_3)^2)} \end{aligned} \quad 112$$

A las ecuaciones 94 y 95 se le aplicara el método jacobiano el proceso se puede apreciar en el Anexo 1. Una vez aplicado el método de jacobiano en las ecuaciones respectivas, las matrices jacobianas quedaran armadas como se aprecian en las siguientes ecuaciones 96 y 97.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{C_3 + C_1 * C_8}{C_1 * C_5 - 1} & \frac{C_1 * C_6}{C_1 * C_5 - 1} & -\frac{C_4 + C_1 * C_9}{C_1 * C_5 - 1} \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{C_8 + C_3 * C_5}{C_1 * C_5 - 1} & -\frac{C_6}{C_1 * C_5 - 1} & \frac{C_9 + C_4 * C_5}{C_1 * C_5 - 1} \end{bmatrix} \quad 113$$

$$B = \begin{bmatrix} 0 \\ \frac{C_2 + C_1 * C_7}{C_1 * C_5 - 1} \\ 0 \\ \frac{C_7 + C_2 * C_5}{C_1 * C_5 - 1} \end{bmatrix} \quad 114$$

Armando la matriz jacobiana C queda expresada de la siguiente manera:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 115$$

Armando la matriz jacobiana D queda expresada de la siguiente manera:

$$D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad 116$$

3.2.4. Análisis de diseño del control LQR

El Regulador Cuadrático Lineal (LQR) es una técnica de control óptimo que busca minimizar la función de costo cuadrática asociada al sistema lineal. En un robot balancín, el LQR se utiliza para diseñar un controlador que logre estabilizar el sistema alrededor de una posición de equilibrio deseada. Aprovechando la linealización del modelo dinámico del robot, el LQR asigna ponderaciones a las variables de estado y al esfuerzo de control, permitiendo una sintonización fina que mejora la estabilidad y la respuesta transitoria del robot. Este enfoque ofrece una solución eficiente para controlar sistemas dinámicos lineales, como el robot balancín, asegurando un comportamiento equilibrado y preciso. Una vez obtenidos los datos se procede a reemplazar los valores obtenidos de las simulaciones anteriores en las ecuaciones predeterminadas de espacio de estado, expresadas en la ecuación 3.

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad 117$$

Las matrices que se aprecian en la ecuación 3 son las que se muestran en las ecuaciones 96, 97, 98 y 99 respectivamente. Mediante la herramienta de Matlab se hallaron los respectivos valores de estas matrices en modelo continuo en espacio de estado, los cuales se encuentran en las tablas 3, 4, 5 y 6 respectivamente; el código que se utilizó se aprecia en el Anexo 2.

Tabla 7. Valores en espacio de estado continuo de la matriz A.

	x_1	x_2	x_3	x_4
x_1	0	1	0	0
x_2	0	27.15	-3.61	0.22
x_3	0	0	0	1
x_4	0	-162	24.76	1.08

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 27.15 & -3.61 & 0.22 \\ 0 & 0 & 0 & 1 \\ 0 & -162 & 24.76 & 1.08 \end{bmatrix}$$

Tabla 8. Valores en espacio de estado continuo de la matriz B.

	u_1
x_1	0
x_2	-0.016
x_3	0
x_4	-0.482

$$B = \begin{bmatrix} 0 \\ -0.016 \\ 0 \\ -0.482 \end{bmatrix}$$

Tabla 9. Valores en espacio de estado continuo de la matriz C.

	x_1	x_2	x_3	x_4
y_1	1	0	0	0
y_1	0	0	1	0

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Tabla 10. Valores en espacio de estado continuo de la matriz D

	u_1
y_1	0
y_1	0

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Las matrices en modelo discreto en espacio de estado están representadas en las tablas 7, 8, 9, 10 respectivamente; el código que se utilizó se aprecia en el Anexo 2.

Tabla 11. Valores en espacio de estado continuo de la matriz A.

	X1	X2	X3	X4
X1	1	-0.296	0.0525	-0.0007
X2	0	-6.92	1.052	-0.01417
X3	0	2.712	0.6582	0.1196
X4	0	54.24	-6.837	1.392

Tabla 12. Valores en espacio de estado continuo de la matriz B.

	U1
X1	0.0002679
X2	0.05359
X3	-0.005184
X4	-0.1037

Tabla 13. Valores en espacio de estado continuo de la matriz C.

	X1	X2	X3	X4
Y1	1	-0.148	0.0263	-0.0003543
Y2	0	1.356	0.8291	0.0598

Tabla 14. Valores en espacio de estado continuo de la matriz D

	U1
Y1	0.000134
Y2	-0.002592

Las variables de estado representan las variables internas del sistema, las cuales determinan el comportamiento dinámico a través del tiempo, las variables de estado en este modelo son mencionadas en la ecuación 92, donde x_1 es la posición de la rueda, x_2 es la velocidad del eje de la rueda, x_3 es el ángulo del péndulo en relación con la vertical y x_4 es la velocidad angular del péndulo en relación con la vertical.

3.2.4. Análisis de observabilidad, controlabilidad y estabilidad.

Este análisis proporciona información pertinente sobre el sistema electromecánico que se está estudiando, con el fin de ajustar parámetros del sistema.

El sistema dinámico del robot balancín es propicio para el estudio de inestabilidad, controlabilidad y observabilidad, al dejar al robot sin un control apropiado este sistema tiende a caerse por lo que lo vuelve un sistema inestable, la controlabilidad de este se debe a la capacidad que tiene el sistema para el diseño y la aplicación de un control que permita la estabilización de este, la observabilidad de este se da mediante las mediciones del sensor de posición y velocidad angular.

Para poder comprobar los parámetros antes mencionados se debe realizar una serie de procesos matemáticos para identificar qué tipo de sistema es, para ello se tomarán las matrices A, B, C y D, y se desarrollarán códigos en el software de Matlab donde los valores arrojados determinarán las características del sistema.

Para determinar que el sistema es estable o no, se procede a realizar el código en Matlab el cual se aprecia en el Anexo (3), el cual permite verificar mediante las matrices antes mencionadas si el sistema es estable, la respuesta del cálculo realizado determino que el sistema es inestable, donde los polos del sistema están en la tabla 15:

Tabla 15. Polos del sistema

Polos del sistema
0
26.5712
2.8161
-1.1615

Al someter el sistema a señales de entrada básicas en el modelo de espacio de estado para obtener la respuesta de forma gráfica, comprobando de esta manera si el sistema es estable o no. Para tiempo continuo y discreto el comportamiento del sistema se puede observar en la siguiente figura.

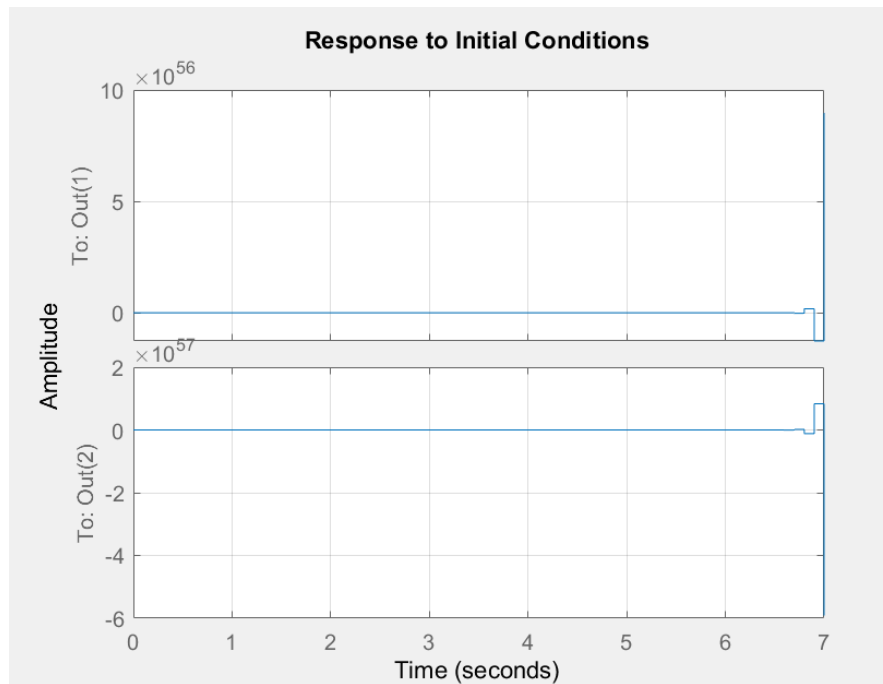


Figura 12. Respuesta de sistema en lazo abierto.

Un sistema dinámico se considera controlable en el tiempo si, a través de la aplicación de un vector de control adecuado, es posible transferir el sistema desde cualquier estado inicial a cualquier otro estado en un intervalo de tiempo finito, sin restricciones significativas.

La controlabilidad en el tiempo es una propiedad importante en el diseño de sistemas de control, ya que garantiza que el sistema pueda ser manejado de manera efectiva y pueda alcanzar cualquier estado deseado dentro de un periodo de tiempo limitado. Este concepto es fundamental para la teoría del control y tiene aplicaciones prácticas en diversos campos, como la ingeniería de control, la robótica y la automatización.

Se realizará la obtención de la matriz de controlabilidad mediante el cálculo de estas en el software de Matlab, los valores encontrados están elevados a $1e^3$ y se observan en la tabla 12.

Tabla 16. Matriz de controlabilidad

Matriz de controlabilidad			
0	-0.0000	-0.0006	-0.0132
-0.0000	-0.0006	-0.0132	-0.3472
0	-0.0005	0.0022	0.0824
-0.0005	0.0022	0.0824	2.2779

Solo si la dimensión del espacio generado por la cantidad de variables cumple con la condición siguiente, el sistema será controlable:

$$\text{rank}[B \ AB \ A^2B \ \dots \ A^{n-1}B] = n \quad 118$$

El valor del rango es igual a 4 por lo cual satisface la ecuación 101 permitiendo que este sistema sea controlable, asegurando de esta manera el diseño de un control para la estabilización.

Para que un sistema sea observable se debe cumplir que $x(t_0)$ sea determinado mediante la observabilidad de $y(t)$ en un intervalo de tiempo finito $t_0 \leq t \leq t_1$, esto indica que las transiciones del estado afectan a todos los vectores de salida eventualmente. La comprobación de observabilidad implica determinar si es factible estimar los estados internos de un sistema basándose únicamente en las salidas observadas. Este proceso se realiza mediante la construcción de una matriz de observabilidad a partir de las matrices del sistema y la evaluación de su rango. Los

valores encontrados matriz de observabilidad están elevados a $1e^3$ y se detallan a continuación en la tabla 13.

Tabla 17. Matriz de observabilidad

Matriz de observabilidad			
0.0010	0	0	0
0	0	0.0010	0
0	0.0010	0	0
0	0	0	0.0010
0	0.0271	-0.0036	0.0002
0	-0.1620	0.0248	0.0011
0	0.7007	-0.0926	0.0027
0	-4.5728	0.6121	-0.0102

Si la dimensión del espacio creado por la matriz de controlabilidad es igual al número de variables de estado, el sistema será observable solo si cumple con la condición siguiente:

$$\text{rank} \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} = n \quad 119$$

El valor del rango es igual a 4 por lo cual satisface la ecuación 102 permitiendo que este sistema sea observable, asegurando de esta manera el diseño de un observador de estado.

3.2.5. Diseño del control LQR

El objetivo principal que tiene el controlador LQR es hallar una ley de control $u = -Kx$ que busca minimizar un índice de desempeño cuadrático, es decir, el propósito del controlador es reducir al mínimo la energía necesaria para cumplir con las especificaciones de diseño. Esto implica tomar decisiones equilibradas y óptimas sobre cuánta energía $u(t)$ suministrar al sistema y cuán rápidamente responde el sistema (estados del sistema, $x(t)$), para lograr un valor ideal que cumpla con las especificaciones de diseño utilizando la menor cantidad de energía posible.

Estas nociones, al ser formuladas en términos matemáticos y tratarse de un problema de optimización que emplea el control óptimo LQR de horizonte infinito (de estado estacionario), minimizar la función de costo cuadrática queda expresada de la siguiente manera:

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad 120$$

La cual está sujeta a la siguiente dinámica del sistema:

$$\dot{x} = Ax + Bu \quad x(0) = x_0 \quad 121$$

En esta formulación, donde $Q \geq 0$ y $R \geq 0$, respectivamente, representan las matrices de ponderación de estados y energía de control. La ley de control $u = -Kx$ da solución a este problema de optimización, donde K se define como:

$$K = -R^{-1}B^T P \quad 122$$

Para encontrar la matriz P , se debe resolver la ecuación algebraica conocida como Ecuación de Riccati:

$$A^T P + PA + Q - PBR^{-1}B^T P = 0 \quad 123$$

Es necesario seleccionar los valores de las matrices Q y R , que representan el peso o la importancia de la rapidez del sistema y la energía de control, respectivamente. Después de realizar varias pruebas, considerando que las dimensiones de las matrices para el sistema aumentado serán de 5×5 y 1×1 respectivamente, los valores que proporcionaron los mejores resultados, cumpliendo con las especificaciones de diseño, son los siguientes:

$$Q = \begin{bmatrix} 20 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 50 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 15 \end{bmatrix}$$

$$R = [0.5]$$

Mediante el software de Matlab se empleó el comando LQR el cual permite calcular la matriz K de retroalimentación de estados óptima para el sistema, el código utilizado se encuentra en el Anexo (6).

La ejecución del código permitió obtener los siguientes datos, la matriz k_{lqi} que es la matriz de ganancias del controlador LQR, la matriz S es la solución a la ecuación de Riccati asociada al LQR, e son los valores propios del sistema en lazo cerrado.

$$k_{lqi} = [0.0354 \quad -3.6264 \quad 0.5048 \quad 0.0023 \quad -0.0055]$$

$$S = \begin{bmatrix} 0.0175 & -0.0218 & 0.0048 & 0.0004 & -0.0030 \\ -0.0218 & 1.1873 & -0.1709 & -0.0041 & 0.0034 \\ 0.0048 & -0.1709 & 0.0252 & 0.0008 & -0.0008 \\ 0.0004 & -0.0041 & 0.0008 & 0.0001 & -0.0001 \\ -0.0030 & 0.0034 & -0.0008 & -0.0001 & 0.0010 \end{bmatrix}$$

$$e = \begin{bmatrix} -0.1842 & +0.1779i \\ -0.1842 & -0.1779i \\ -2.6063 & +0.1274i \\ -2.6063 & -0.1274i \\ -26.5433 & +0.0000i \end{bmatrix}$$

Los vectores de control optimo quedan expresados de la siguiente manera:

$$k_{lqr} = [0.0354 \quad -3.6264 \quad 0.5048 \quad 0.0023]$$

$$k_i = [-5.4772]$$

Después de encontrar el valor del vector de control ideal para el sistema, aplicamos la realimentación del sistema en lazo cerrado y calculamos los autovalores resultantes. Los resultados son:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0.6 & 34.32 & 4.942 & 0.2622 & -0.09284 \\ 0 & 0 & 0 & 1 & 0 \\ 17.8 & 1912 & 268.3 & 2.196 & -2.643 \\ -1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ -0.01695 \\ 0 \\ -0.4825 \\ 0 \end{bmatrix}$$

$$C = [1 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$D = [0]$$

Se determinaron los autovalores del sistema con realimentación utilizando el comando “eig” de MATLAB, y se obtuvo:

$$e = \begin{bmatrix} -26.5433 & +0.0000i \\ -2.6063 & +0.1274i \\ -2.6063 & -0.1274i \\ -0.1842 & +0.1779i \\ -0.1842 & -0.1779i \end{bmatrix}$$

Al analizar el conjunto de autovalores del sistema en lazo cerrado, se llega a la conclusión de que el sistema, cuando se realimenta para el seguimiento de referencia, exhibe estabilidad.

3.2.6. Diseño de control LQR en Simulink

El diseño de control LQR en simulink comprende de la implementación y simulación del controlador, el cual implica la configuración de bloques que representan el sistema a controlar, la formulación del modelo de espacio de estados del sistema, la definición de las matrices de ponderación Q y R para la función de costo cuadrática, y la implementación del controlador LQR. Esta simulación va acompañada del código

realizado para la obtención de las diferentes matrices y valores necesarios para llevar a cabo la simulación del sistema.

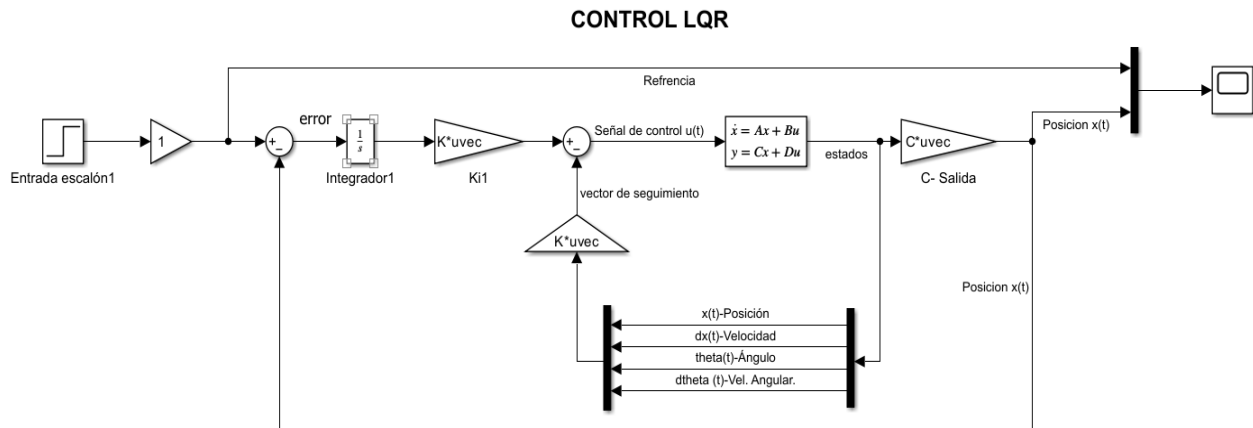


Figura 13. Sistema de control LQR en Simulink.

Una vez que se han encontrado todos los valores que son necesarios para la simulación del sistema electromecánico se procede a desarrollarlo en el software de Simulink de Matlab el cual se muestra en la siguiente figura 14, en este se aprecia una representación visual del modelo del sistema físico desarrollado en espacio de estados. Este diseño integra los vectores de control obtenidos en las secciones anterior, permitiendo una simulación completa muy cercana al sistema físico estudiado.

Se puede apreciar en esta figura como cada una de las señales de entrada y salida interactúan entre si mediante una retroalimentación continua, esta retroalimentación se debe a diferentes elementos como las ganancias encontradas anteriormente, el bloque de espacio de estados respectivamente el cual comprende de las ecuaciones antes mencionadas, el integrador para mejorar el seguimiento de la referencia, la ganancia de valor inicial del ángulo para lograr que el sistema se mantenga dentro de las referencias de estabilidad, los respectivos osciloscopios para visualizar las diferentes señales del sistema electromecánico.

3.2.7. Diseño de control LQR en Arduino

En esta etapa se sintetiza la ejecución del código en el entorno de desarrollo Arduino IDE, con el propósito de cargar el microcontrolador Arduino Uno y establecer la comunicación con el sistema robótico. Los complementos del código se encuentran en el anexo (7).

```

#include "Motor.h"
#include "PinChangeInt.h"
static void ContadorEncoderDerechaA();
static void ContadorEncoderIzquierdaA();
void Motor::IniciarEncoder()
{
    attachInterrupt(digitalPinToInterrupt(PIN_ENCODER_IZQUIERDA_A),
ContadorEncoderIzquierdaA, CHANGE);
    attachInterrupt(digitalPinToInterrupt(PIN_ENCODER_DERECHA_A),
ContadorEncoderDerechaA, CHANGE);
}
unsigned long Motor::contador_encoder_derecha_a;
//Conseguir velocidad en la rueda derecha.
static void ContadorEncoderDerechaA()
{
    Motor::contador_encoder_derecha_a++;
}
unsigned long Motor::contador_encoder_izquierda_a;
//Conseguir velocidad en la rueda izquierda.
static void ContadorEncoderIzquierdaA()
{
    Motor::contador_encoder_izquierda_a++;
}

```


CAPÍTULO IV

En este ítem se exponen las evaluaciones numéricas del controlador LQR en diversas condiciones iniciales.

4.1. Parámetros de evaluación

El fin de realizar una comparación del LQR en un ambiente controlado o simulado con el modelo real es determinar en qué condiciones es más efectivo este controlador, para ello se deben tomar en consideración algunos parámetros específicos como:

- Tiempo de estabilización (t_{ss}).
- Máximo sobre impulso (M_p).
- Señal de control máxima (u_{max}) y mínima (u_{min}).
- Error de estado estacionario (e_{ss}).
- Angulo de inclinación (θ_{max}).

4.2. Evaluaciones

El objetivo de esta evaluación es poder comprender el rendimiento y evaluar el funcionamiento del control LQR sobre el sistema electromecánico del robot balancín, para ello se realizará simulaciones en el software de simulink para el análisis posterior de los datos obtenidos de los diferentes parámetros, se toma como referencia el tiempo de simulación de 150 segundos. Los valores reales de referencia del sistema electromecánico se detallan en la tabla 18.

Tabla 18. Especificaciones del control LQR.

Especificaciones	Parámetros reales del sistema
Máximo sobre impulso	$M_p = 9,97\%$
Tiempo de estabilización (criterio 2%)	$t_{ss} = 5,35 \text{ s}$
Valores máximos de la señal de control	$u_{max} = 9.3 \text{ V}$
Valores mínimos de la señal de control	$u_{min} = -0.58 \text{ V}$
Rango de valores de la señal de control	$-0.1134 \leq u(t) \leq 0.8551$
Error de estado estacionario	$e_{ss}=0,3370$
Ángulo de inclinación máximo	$\theta_{max} = 9.52^\circ$
Ángulo de inclinación mínimo	$\theta_{min} = -9.54^\circ$

4.2.2. Respuesta función escalón

Para llevar a cabo el análisis, se utilizará una señal de entrada de 4 cm, la cual representará el desplazamiento que experimentará el robot balancín. Esta medida servirá como referencia fundamental para evaluar el rendimiento y comportamiento del controlador LQR sobre el robot durante su funcionamiento.

Además, se tomará en cuenta los parámetros de evaluación como el tiempo de estabilización, señal de control mínima y máxima, así como también el error estacionario. Estas señales estarán sometidas al criterio de tolerancia del 2% y se calculará el promedio de error absoluto para ello se hará uso de la siguiente ecuación.

$$E.A_{prom} = \frac{|x_{set} - x_i|}{x_{set}} \quad 124$$

Las Figura 15, 16 y 17 proporciona una visualización detallada del comportamiento simulado de las señales de posición y de control bajo la influencia de una señal de referencia con una amplitud de 4 cm, esta indica el desplazamiento que tiene el robot balancín. Estos resultados se presentan de manera más exhaustiva en la Tabla 19, la cual ofrece un desglose detallado de las mediciones y los valores obtenidos. Este análisis es crucial para comprender el impacto y la eficacia del controlador LQR, proporcionando información para futuras optimizaciones y ajustes.

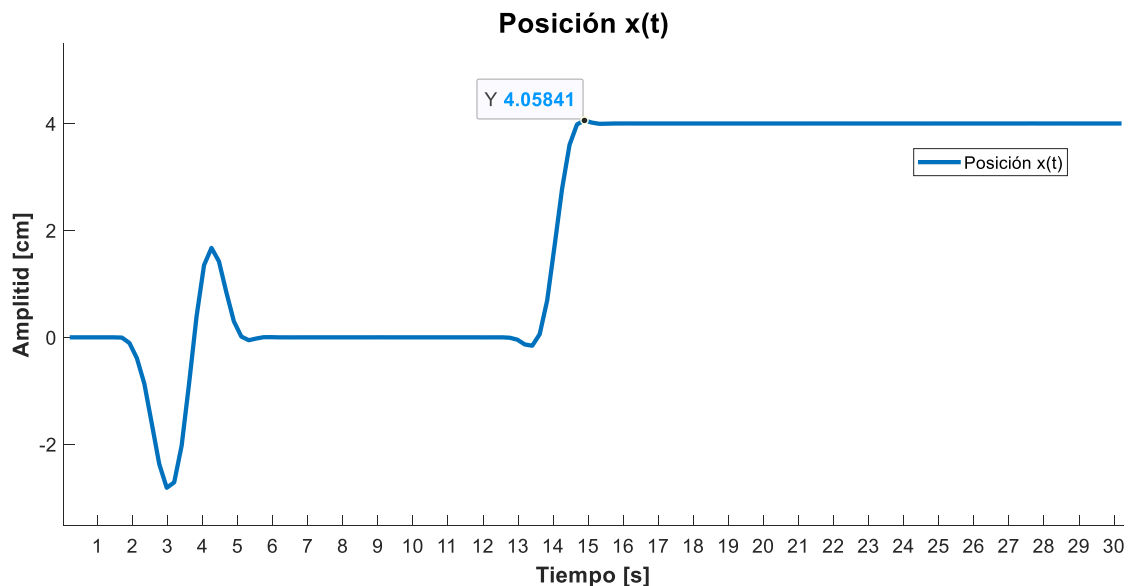


Figura 14. Resultado de la simulación de la posición con entrada escalón de 4 cm.

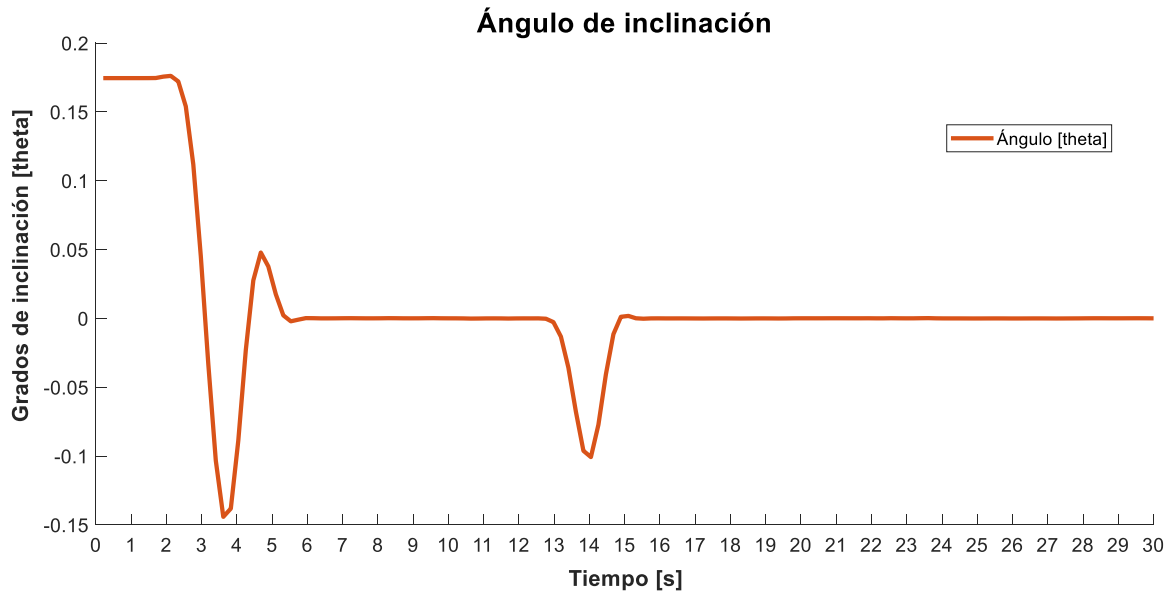


Figura 15. Resultado de la simulación del ángulo de inclinación con entrada escalón de 4 cm.

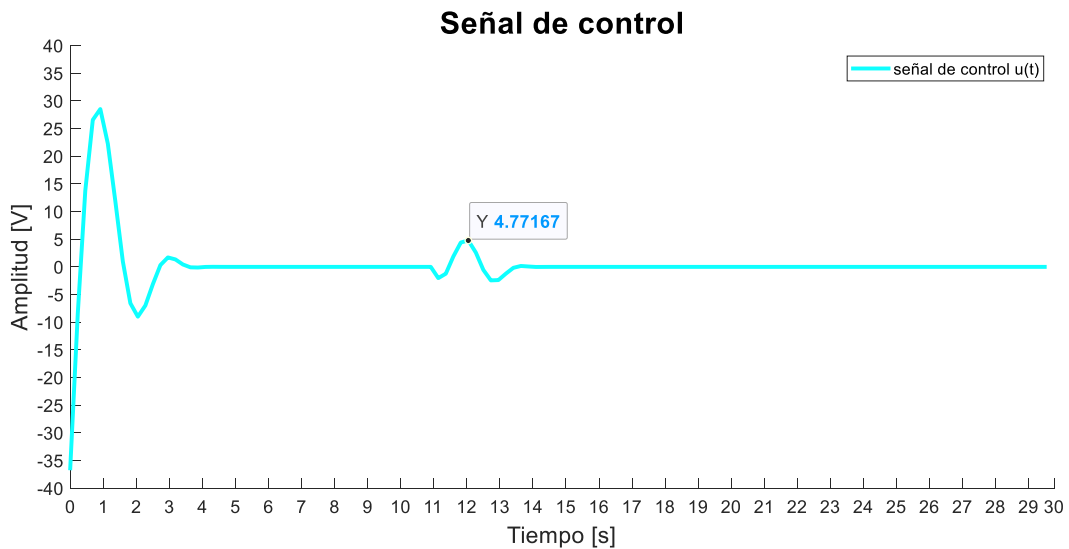


Figura 16. Resultado de la simulación de la señal de control con entrada escalón de 4 cm.

Tabla 19. Resultados de las simulaciones para una entrada escalón

Amplitud (cm)	t_{ss} (s)	M_p (%)	$u(t)$	e_{ss} (%)	θ_{max}
4	0,66	3,5	3,89	0,55	0,13

Los datos de la tabla 19 proporcionan información sobre el desempeño del control LQR en respuesta a una entrada de referencia tipo escalón, se observa que al desplazar 4 cm al robot, se logra un tiempo de estabilización es de 0,66 segundos, lo que indica una respuesta rápida del sistema para alcanzar el estado deseado. El máximo sobreimpulso del sistema alcanza un valor de 3,5%, lo que sugiere una respuesta controlada y bien ajustada que minimiza las oscilaciones.

El error de estado estacionario es de 0,55%, demostrando la capacidad del sistema para seguir de cerca la referencia deseada, lo que es fundamental para aplicaciones que requieren una precisión extrema. Por otro lado, el ángulo máximo de inclinación es de $0,13^{\circ}$, indicando que el robot experimenta una inclinación moderada en relación con la referencia, esta cantidad relativamente baja de inclinación indica que el sistema mantiene una estabilidad adecuada. La señal de control tiene un valor de 3,89 v, el cual está en un rango de funcionamiento optimo.

La figura 16 muestra los resultados de la implementación del controlador LQR en la planta física del sistema, en el cual se detalla el comportamiento de las señales de posición y de control al introducir una señal de tipo escalón con amplitud de 4cm. Los resultados se presentan en la tabla 20, la cual ofrece un desglose detallado de las mediciones y los valores obtenidos. Este análisis es crucial para comprender el impacto y la eficacia del controlador LQR, proporcionando información para futuras optimizaciones y ajustes.

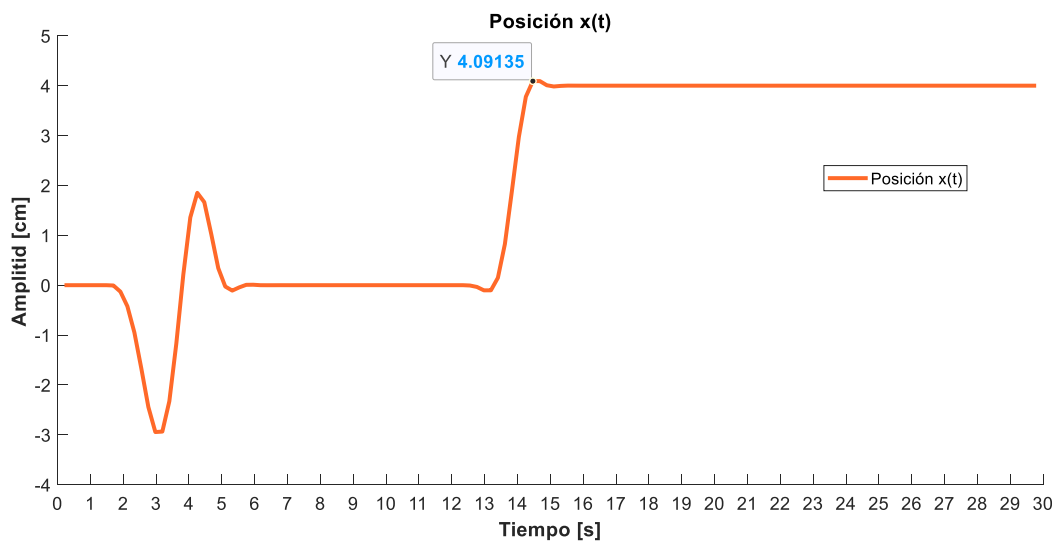


Figura 17. Resultado de la prueba física de la posición con entrada escalón de 4 cm.

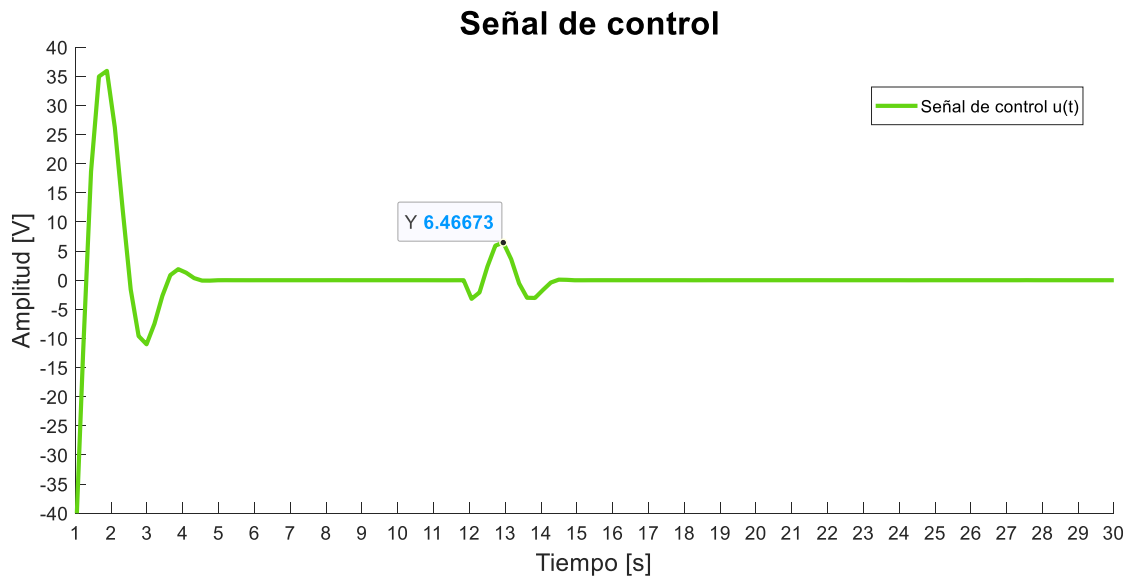


Figura 18. Resultado de la prueba física de la señal de control con entrada escalón de 4 cm.

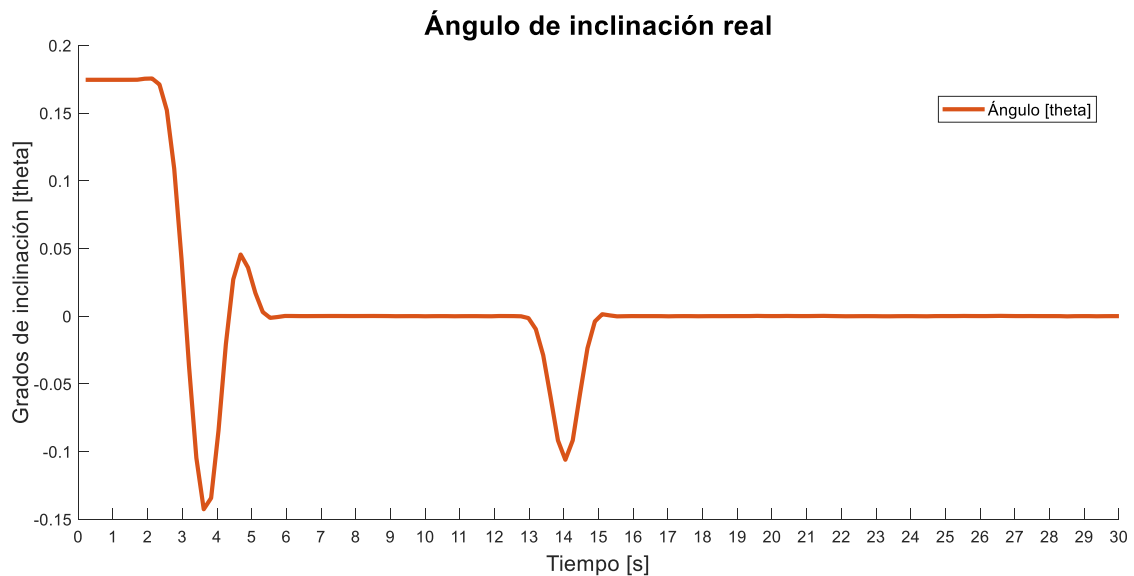


Figura 19. Resultado de la prueba física del ángulo de inclinación con entrada escalón de 4 cm.

Tabla 20. Resultados de las pruebas para una entrada escalón

Amplitud (cm)	t_{ss} (s)	M_p (%)	$u(t)$	e_{ss} (%)	θ_{max}
4	1,35	1,82	5,57	0,24	0,56

Los datos de la tabla 20 proporcionan información sobre el desempeño del control LQR en el sistema electromecánico del robot balancín en respuesta a una entrada de referencia tipo escalón, se observa que al desplazar 4 cm al robot, se logra un tiempo de estabilización es de 1,35 segundos, lo que indica una respuesta rápida del sistema para alcanzar el estado deseado. El máximo sobreimpulso del sistema alcanza

un valor de 1,82%, lo que sugiere una respuesta controlada y bien ajustada que minimiza las oscilaciones.

El error de estado estacionario es de 0,24%, demostrando la capacidad del sistema para seguir de cerca la referencia deseada, lo que es fundamental para aplicaciones que requieren una precisión extrema. Por otro lado el ángulo máximo de inclinación es de $0,56^{\circ}$, indicando que el robot experimenta una inclinación moderada en relación con la referencia, esta cantidad relativamente baja de inclinación indica que el sistema mantiene una estabilidad adecuada. La señal de control tiene un valor de 5,57 v, el cual está en un rango de funcionamiento óptimo.

4.2.6. Comparación del controlador LQR

El sistema simulado en el software de simulink del modelo del robot balancín proporciona una representación del comportamiento del sistema bajo condiciones específicas, la comparación de este controlador radica en validar la presión del modelo matemático y comprobar la efectividad y robustez del diseño del controlador LQR en un entorno físico. En las siguientes tablas se proporcionan datos de los parámetros a evaluar como el sobreimpulso, tiempo de estabilización, valores de la señal de control, ángulo de inclinación y el error de estado estacionario, los cuales permitirán determinar el comportamiento del controlador LQR sobre la planta real y sobre el modelo matemático.

La comparación del máximo sobreimpulso del sistema físico con el sistema simulado se parecía en la figura 20, tabla 21 muestra los resultados de estas pruebas.

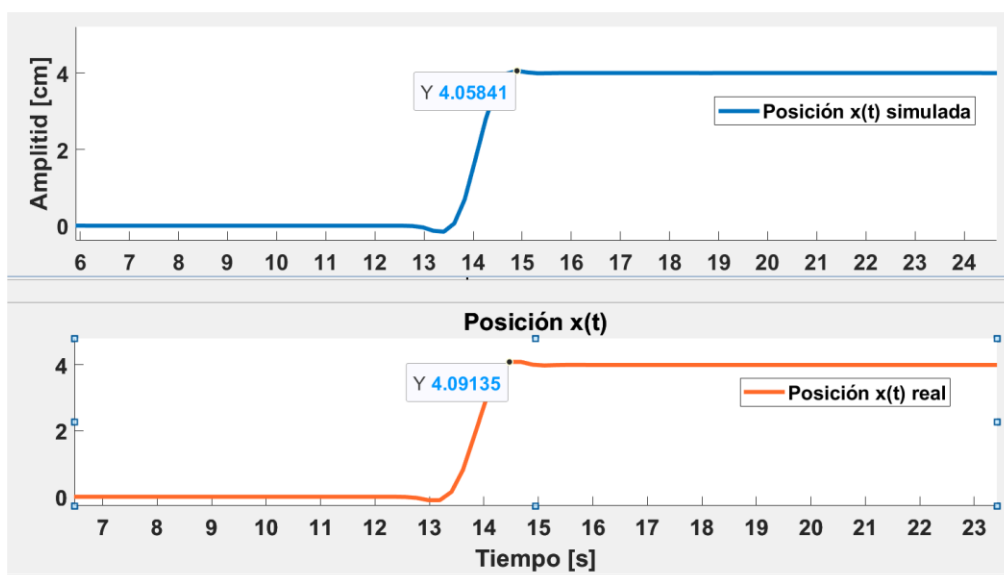


Figura 20. Comparación del sobre impulso del sistema simulado vs el real.

Tabla 21. Comparación del máximo sobreimpulso.

M_p físico	M_p simulado	Error%
1,82%	3,5%	1,68%

La comparación del tiempo de estabilización del sistema físico con el sistema simulado se aprecia en la figura 21, tabla 22 muestra los resultados de esta comparación.

Tabla 22. Comparación del tiempo de estabilización.

t_{ss} físico	t_{ss} simulado	Error%
1,35 s	0,66 s	0,69%

La comparación de la señal de control del sistema físico con el sistema simulado se aprecia en la tabla 23.

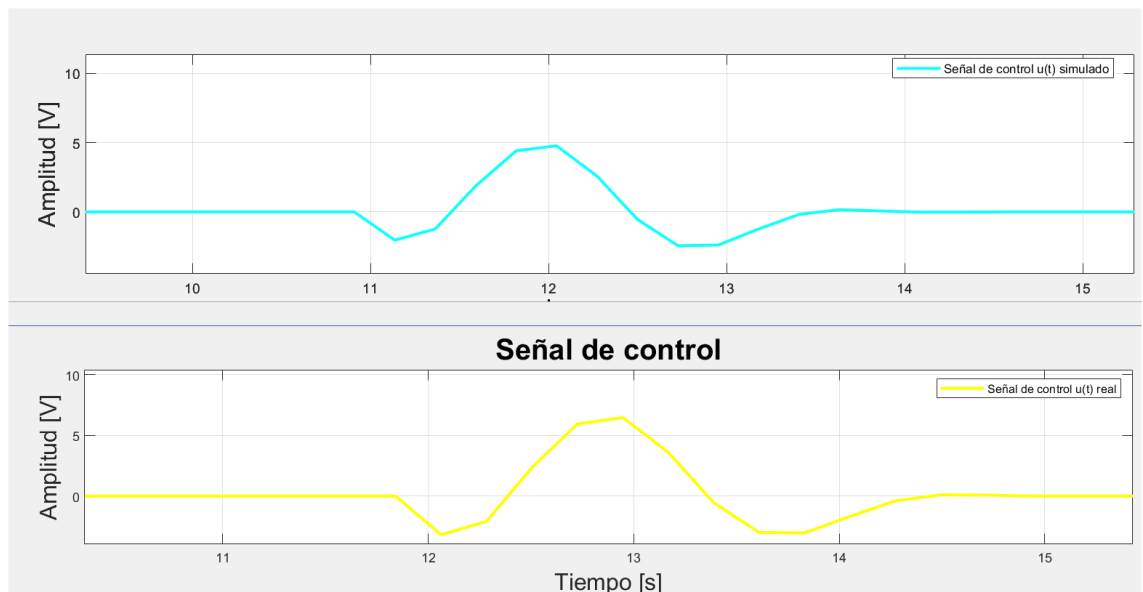


Figura 21. Comparación de la señal de control del sistema simulado vs el real.

Tabla 23. Comparación de la señal de control.

$u(t)$ físico	$u(t)$ simulado	Error%
5,57 V	3,89 V	1,68%

La comparación del ángulo de inclinación del sistema físico con el sistema simulado se aprecia en la tabla 24.

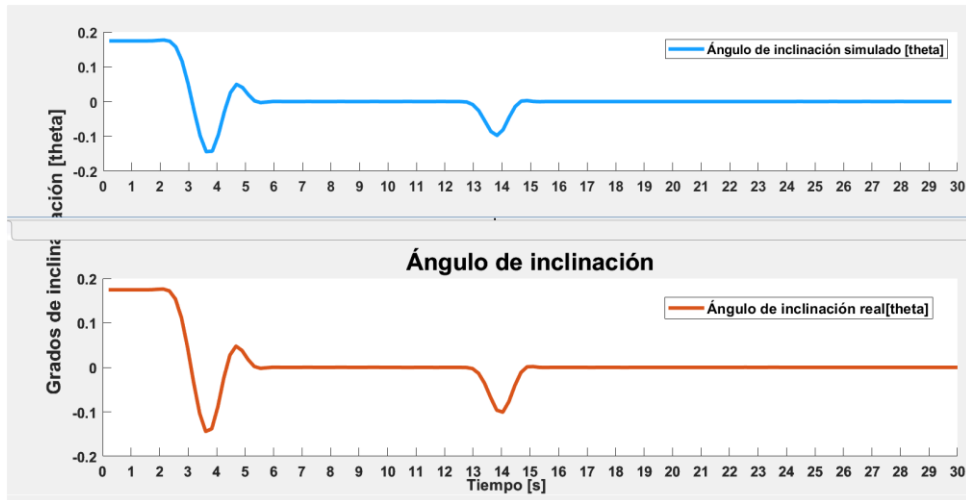


Figura 22. Comparación del ángulo de inclinación del sistema simulado vs el real.

Tabla 24. Comparación del ángulo de inclinación.

θ_{max} físico	θ_{max} simulado	Error%
0,56°	0,13°	0,43%

La comparación del error en estado estacionario del sistema físico con el sistema simulado se aprecia en la tabla 29.

Tabla 25. Comparación del máximo sobreimpulso.

e_{ss} % físico	e_{ss} % simulado	Error%
0,24	0,55	0,31%

CONCLUSIONES

El análisis de los componentes del sistema como la altura del robot, peso de cada componente electrónico y mecánico como las llantas, ecoders, motor, entre otros, la velocidad de los motores, la fuerzas que interactúan sobre la planta y otros elementos claves permitieron obtener un mejor diseño del modelo matemático del controlador LQR.

La comunicación UART permitió la obtención de 551 muestras para realizar la identificación de la planta en el software de Matlab, donde la mejor respuesta fue del modelo SS6 con un 80,8% de aproximación del modelo matemático, con el cual se pudo obtener las matrices de las ecuaciones en espacio de estado. La gran cantidad de muestras obtenidas fue crucial para asegurar la fiabilidad de los datos y permitió obtener con precisión las matrices de las ecuaciones en espacio de estado, fundamentales para el diseño y la implementación efectiva del controlador LQR.

El controlador LQR demuestra ser un controlador óptimo, tanto en el modelo matemático como en el modelo real, los valores de los parámetros evaluados indican una excelente adaptación del controlador a las condiciones del sistema, mostrando un rendimiento consistente y efectivo en ambas instancias.

La implementación exitosa del controlador LQR en el sistema electromecánico del robot balancín resalta su versatilidad y adaptabilidad para aplicaciones que requieren precisión y respuesta rápida, proporcionando una base sólida para futuras optimizaciones y mejoras en el diseño del sistema. Los resultados presentados en las tablas validan esta afirmación, mostrando un tiempo de estabilización de 1.35 segundos y un máximo sobreimpulso del 1.82% en pruebas físicas, comparables con los resultados simulados. Además, el error de estado estacionario de 0.24% y el ángulo de inclinación máximo de 0.56° demuestran la capacidad del controlador para mantener el sistema estable y preciso bajo diversas condiciones.

La capacidad del controlador LQR para mantener el ángulo de inclinación dentro de límites aceptables, incluso bajo influencias externas como la perturbación de una señal de entrada de 4 cm, subraya su eficacia en la estabilización dinámica del robot balancín en condiciones variables. Sin embargo, se observó que ciertas perturbaciones externas, como irregularidades en la superficie de desplazamiento y ruidos externos en el robot, afectaron la implementación real del controlador, lo que no se reflejó completamente en las simulaciones. Esto destaca la importancia de considerar estos

factores adicionales para optimizar aún más el rendimiento del sistema en entornos reales.

RECOMENDACIONES

Es crucial realizar un ajuste adecuado de los parámetros de codificación, específicamente las matrices Q y R del controlador LQR. Para ello, se deben identificar los valores óptimos de estas matrices mediante simulaciones y pruebas físicas detalladas. Es importante considerar las características específicas de los componentes electrónicos, mecánicos y electromecánicos del sistema, asegurando que los parámetros se ajusten en función de diferentes condiciones operativas, como variaciones en las ondas de referencia. Este proceso de ajuste continuo permitirá optimizar el desempeño del sistema y garantizar una respuesta eficiente del controlador bajo diversas condiciones.

Para validar la eficacia y consistencia del controlador LQR, es esencial realizar pruebas adicionales utilizando diferentes tipos de señales. Estas pruebas deben incluir variaciones en la amplitud y frecuencia de las señales de referencia, evaluando su impacto en el tiempo de estabilización y el error estacionario del sistema. Los resultados obtenidos deben documentarse minuciosamente y utilizarse para ajustar los parámetros del controlador según sea necesario. Este enfoque permitirá asegurar que el controlador LQR mantenga un rendimiento óptimo y una alta precisión bajo diversas condiciones de señal de entrada, fortaleciendo la robustez del sistema.

Es recomendable investigar y desarrollar estrategias de control alternativas que puedan manejar de manera más efectiva las señales sinusoidales de alta frecuencia y otras perturbaciones. Esto podría incluir la exploración de técnicas de control adaptativo que se ajusten dinámicamente a las variaciones en la señal de referencia, así como la implementación de controladores híbridos que combinen múltiples enfoques de control. La realización de simulaciones y pruebas físicas será fundamental para evaluar la eficacia de estas estrategias alternativas. Con esto, se mejorará la capacidad del sistema para seguir señales de referencia complejas sin comprometer su estabilidad o eficiencia.

Al desarrollar el modelo matemático del controlador LQR para cualquier dispositivo, es fundamental tener en cuenta factores críticos. Es necesario asegurar la precisión en la identificación del sistema y la calidad de los datos de muestra. Además, debe considerarse la dinámica completa del sistema, incluyendo posibles perturbaciones externas y variabilidad en las condiciones operativas. Validar y ajustar el modelo matemático mediante pruebas experimentales ayudará a reflejar con precisión el comportamiento real del sistema. Esto garantizará que el modelo matemático sea

robusto y representativo, proporcionando una base sólida para el diseño y la implementación efectiva del controlador LQR.

REFERENCIAS

- [1] J. Jin y W. Chung, «Obstacle Avoidance of Two-Wheel Differential Robots Considering the Uncertainty of Robot Motion on the Basis of Encoder Odometry Information», *Sensors*, vol. 19, n.º 2, Art. n.º 2, ene. 2019, doi: 10.3390/s19020289.
- [2] C. A. Bahón y C. R. Giner, *Tecnología de sistemas de control*. Univ. Politèc. de Catalunya, 2004.
- [3] C. A. Á. Ramos y F. A. S. Cárdenas, «Diseño de controladores basados en técnicas de control óptimo lqr+i y h2 para un prototipo del péndulo invertido sobre ruedas», *Rev. Politécnica*, vol. 8, n.º 15, Art. n.º 15, dic. 2012.
- [4] «MONOGRAFIA_AnáliseDesempenhoTécnicas.pdf». Accedido: 30 de enero de 2023. [En línea]. Disponible en: https://monografias.ufop.br/bitstream/35400000/2687/6/MONOGRAFIA_An%c3%a1liseDesempenhoT%c3%a9cnicas.pdf
- [5] B. C. Octavio y R. M. J. Marcelino, «Control PID de un Sistema Balancín».
- [6] A. D. Ñacata Paucar, «Implementación de un sistema WNCS mediante una plataforma basada en la nube para el control de un péndulo invertido», bachelorThesis, 2022. Accedido: 8 de febrero de 2023. [En línea]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/22274>
- [7] J.-B. He, Q.-G. Wang, y T.-H. Lee, «PI/PID controller tuning via LQR approach», *Chem. Eng. Sci.*, vol. 55, n.º 13, pp. 2429-2439, jul. 2000, doi: 10.1016/S0009-2509(99)00512-6.
- [8] C. D. G. Uribe, «COMPARACIÓN DE UN CONTROLADOR LQR VS UN CONTROLADOR PID IMPLEMENTADOS EN UN HELICÓPTERO DE DOS GRADOS DE LIBERTAD PIVOTADO.».
- [9] T. A. Nguyen, «Control an Active Suspension System by Using PID and LQR Controller», *Int. J. Mech. Prod. Eng. Res. Dev.*, vol. 10, pp. 7003-7012, ene. 2020, doi: 10.24247/ijmperdjun2020662.
- [10] J. Borggaard y L. Zietsman, «The Quadratic-Quadratic Regulator Problem: Approximating feedback controls for quadratic-in-state nonlinear systems», en *2020 American Control Conference (ACC)*, Denver, CO, USA: IEEE, jul. 2020, pp. 818-823. doi: 10.23919/ACC45564.2020.9147286.
- [11] P. A. Cardozo Cardozo y J. S. Conde Cárdenas, «Construcción de prácticas de laboratorio bajo la iniciativa CDIO a partir de la implementación de las técnicas de control PID y LQR en un minidrone parrot rolling spider para la universidad Antonio Nariño.», *InstnameUniversidad Antonio Nariño*, mar. 2021, Accedido: 8 de febrero de 2023. [En línea]. Disponible en: <http://repositorio.uan.edu.co/handle/123456789/2467>
- [12] Y. Gong *et al.*, «Feedback Control of a Cassie Bipedal Robot: Walking, Standing, and Riding a Segway», en *2019 American Control Conference (ACC)*, jul. 2019, pp. 4559-4566. doi: 10.23919/ACC.2019.8814833.
- [13] P. Chatterjee, N. Lahiri, A. Bhattacharjee, y A. Chakraborty, «Automated Hybrid Stair Climber for Physically Challenged People», en *2021 5th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*, sep. 2021, pp. 1-4. doi: 10.1109/IEMENTech53263.2021.9614713.
- [14] «ELEGOO Tumbler Self-Balancing Robot Car Kit Compatible con Arduino IDE STEM Kits - ELEGOO Oficial». Accedido: 9 de febrero de 2023. [En línea].

- Disponible en: <https://www.elegoo.com/collections/arduino-learning-sets/products/elegoo-tumbler-self-balancing-robot-car>
- [15] G. Castro y M. Ángeles, «Navegación de un vehículo equilibrista mediante seguimiento de pared o cámaras externas», 2022, Accedido: 9 de febrero de 2023. [En línea]. Disponible en: <https://repositorio.comillas.edu/xmlui/handle/11531/61927>
- [16] «Control de equilibrio de un segway a través del hardware de Texas Instruments». Accedido: 9 de febrero de 2023. [En línea]. Disponible en: <https://www.politesi.polimi.it/handle/10589/174465>
- [17] E. Ramallo García, «Control y detección de fallos en motores DC para un sistema basado en Arduino», Bachelor thesis, Universitat Politècnica de Catalunya, 2022. Accedido: 9 de febrero de 2023. [En línea]. Disponible en: <https://upcommons.upc.edu/handle/2117/370672>
- [18] V. A. Chalán Padilla, «Desarrollo de un controlador óptimo LQR utilizando herramientas IOT para un sistema de presión constante controlado remotamente», masterThesis, 2020. Accedido: 18 de enero de 2023. [En línea]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/19395>
- [19] W. H. Tambi Pupiales, «Análisis comparativo entre un control LQR y un control clásico en un convertidor Boost mediante una simulación en Simulink de Matlab», bachelorThesis, 2022. Accedido: 18 de enero de 2023. [En línea]. Disponible en: <http://repositorio.utn.edu.ec/handle/123456789/12078>
- [20] «43.1470-PEREZ-VALENCIA-JASON-EDUARDO.pdf». Accedido: 30 de enero de 2023. [En línea]. Disponible en: <http://dspace.istvidanueva.edu.ec/bitstream/123456789/116/1/43.1470-PEREZ-VALENCIA-JASON-EDUARDO.pdf>
- [21] J. E. Pérez Valencia, «Construcción de un robot balancín controlado por una aplicación móvil», feb. 2021, Accedido: 9 de febrero de 2023. [En línea]. Disponible en: <http://dspace.istvidanueva.edu.ec/xmlui/handle/123456789/116>
- [22] H. F. Quiroga Gaona, «Sistema de control automático enfocado en el proceso de enfriamiento de mosto en la cervecería artesanal», bachelorThesis, La Libertad: Universidad Estatal Península de Santa Elena, 2020, 2018. Accedido: 7 de febrero de 2023. [En línea]. Disponible en: <https://repositorio.upse.edu.ec/handle/46000/5359>
- [23] W. A. Khan, L. Wisniewski, D. Lang, y J. Jasperneite, «Analysis of the requirements for offering industrie 4.0 applications as a cloud service», en *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, Edinburgh, UK: IEEE, jun. 2017, pp. 1181-1188. doi: 10.1109/ISIE.2017.8001413.
- [24] S. Wang, J. Wan, D. Zhang, D. Li, y C. Zhang, «Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination», *Comput. Netw.*, vol. 101, pp. 158-168, jun. 2016, doi: 10.1016/j.comnet.2015.12.017.
- [25] World Bank Group, *World Development Report 2016: Digital Dividends*. Washington, DC: World Bank, 2016. doi: 10.1596/978-1-4648-0671-1.
- [26] «Segway robotic mobility platform». Accedido: 28 de abril de 2023. [En línea]. Disponible en: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/5609/0000/Segway-robotic-mobility-platform/10.1117/12.571750.short?SSO=1>

- [27] «Full article: TRIZ method for innovation applied to an hoverboard». Accedido: 28 de abril de 2023. [En línea]. Disponible en: <https://www.tandfonline.com/doi/full/10.1080/23311916.2018.1524537>
- [28] «DISEÑO INTEGRADO, MODELADO Y ANÁLISIS DE SILLA DE DOS RUEDAS PARA DISCAPACITADOS | Robótica móvil inspirada en la naturaleza». Accedido: 28 de abril de 2023. [En línea]. Disponible en: https://www.worldscientific.com/doi/abs/10.1142/9789814525534_0021
- [29] P. Hall, «Ingeniería de control moderna».
- [30] C. D. Camacho Morales, «Diseño y simulación de un controlador por modos deslizantes dinámicos para sistemas inestables en lazo abierto con retardo de tiempo.», bachelorThesis, Quito, 2021, 2021. Accedido: 5 de mayo de 2023. [En línea]. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/21956>
- [31] «Tipos de control (lazo abierto y cerrado) - 03fmeautomatismos». Accedido: 3 de mayo de 2023. [En línea]. Disponible en: <https://sites.google.com/site/03fmeautomatismos/home/tipos-control>
- [32] A. L. Anderson, «Control predictivo robusto apto para la identificación de sistemas en lazo cerrado», mar. 2019, Accedido: 5 de mayo de 2023. [En línea]. Disponible en: <https://ri.conicet.gov.ar/handle/11336/86877>
- [33] L. De La A Salinas y M. A. Velasco, «“Estimación de Estados Empleando Técnicas Basadas en la Teoría del Filtro de Kalman para la Localización de un Robot Móvil». Accedido: 3 de abril de 2024. [En línea]. Disponible en: <https://www.dspace.espol.edu.ec/retrieve/f014e351-6892-45c4-96a5-025f9f67f074/T-114119-FIEC-POSTG043.pdf>
- [34] Xavier Alvarez Brotons, «Capítulo 2. Sistemas de Control». 2004. [En línea]. Disponible en: upcommons.upc.edu/bitstream/handle/2099.1/3330/34059-5.pdf?sequence=5
- [35] Personal de Lab-Volt, *Flúidos y Sensores*, Primera edición. Quebec: Lab-Volt Ltda., 2001. [En línea]. Disponible en: biblio.url.edu.gt/Publi/Libros/2013/ManualesIng/FluidosySensores-O.pdf
- [36] Leonel Germán Corona Ramirez, Griselda Stephany Abarca Jimenez, y Jesús Mares Carreño, *Sensores y actuadores*. Grupo Editorial Patria, 2014.
- [37] Eugenio Vidósola C., «ACTUADORES». Soltex Chile S.A. [En línea]. Disponible en: academia.edu/29303028/Actuadores_1_
- [38] Mario Alberto Perez, Analía Perez Hidalgo, y Elisa Perez Berenguer, «INTRODUCCIÓN A LOS SISTEMAS DE CONTROL Y MODELO MATEMÁTICO PARA SISTEMAS LINEALES INVARIANTES EN EL TIEMPO». 2007. [En línea]. Disponible en: dea.unsj.edu.ar/control1/apuntes/unida1y2.pdf
- [39] G. A. Garrigos Anierte, J. M. Almira Picazo, y M. F. Eike-Christian Jacobus, «Teorema del muestreo, y transformadas de Fourier continua y discreta, con aplicaciones». Accedido: 3 de enero de 2024. [En línea]. Disponible en: https://webs.um.es/gustavo.garrigos/tfg/Fecht_julio2018.pdf
- [40] G. Westreicher, «Proceso», Economipedia. Accedido: 3 de enero de 2024. [En línea]. Disponible en: <https://economipedia.com/definiciones/proceso.html>
- [41] H. J. Rosas Ramos, «Diseño y análisis comparativo de un sistema de péndulo invertido aplicando tecnicas de control lineal cuadratico Gaussiano», *Univ. Católica St. María*, sep. 2022, Accedido: 3 de enero de 2024. [En línea]. Disponible en: <https://repositorio.ucsm.edu.pe/handle/20.500.12920/12050>

- [42] «Optimización Multiobjetivo en Modelado y Control de un Péndulo Invertido Rotatorio». Accedido: 3 de enero de 2024. [En línea]. Disponible en: <https://riunet.upv.es/handle/10251/142820>
- [43] J. C. Geromel y A. G. B. Palhares, *Análise linear de sistemas dinâmicos*. Editora Blucher, 2019.
- [44] J. P. L. Moreno, «Construcción, instrumentación y control de un robot balancín.», ago. 2023, Accedido: 3 de enero de 2024. [En línea]. Disponible en: <https://ring.uaq.mx/handle/123456789/8970>
- [45] A. Florián, «Comparación de desempeño de observadores de estado en sistemas lineales con aplicación a un motor de corriente continua», *Rev. CINTEX*, vol. 23, n.º 1, Art. n.º 1, oct. 2018, doi: 10.33131/24222208.310.
- [46] «Vista de METODOLOGÍAS DE ENSEÑANZA ACTIVA EN INGENIERÍA. CASO PRÁCTICO DE PROYECTO INTEGRADOR PARA LAS ASIGNATURAS DE ROBÓTICA INDUSTRIAL Y CONTROL DIGITAL». Accedido: 5 de mayo de 2023. [En línea]. Disponible en: <https://acofipapers.org/index.php/eiei/article/view/53/48>
- [47] «Dynamic balancing methods for aerial manipulators». Accedido: 3 de abril de 2024. [En línea]. Disponible en: <https://thesis.unipd.it/handle/20.500.12608/45784>
- [48] I. E. Erazo-Velasco, B. F. Chere-Quiñónez, y J. C. Anchundia-Morales, «Diseño de un sistema de control realimentado tipo servo para un modelo electromecánico», *Dominio Las Cienc.*, vol. 8, n.º 1, Art. n.º 1, ene. 2022, doi: 10.23857/dc.v8i1.2509.
- [49] J. P. Cardona, J. J. Leal, J. E. Ustariz, J. P. Cardona, J. J. Leal, y J. E. Ustariz, «Modelado matemático de caja blanca y negra en educación en ingeniería», *Form. Univ.*, vol. 13, n.º 6, pp. 105-118, dic. 2020, doi: 10.4067/S0718-50062020000600105.
- [50] P. N. Roldán, «Modelo matemático - Definición, qué es y concepto», Economipedia. Accedido: 24 de noviembre de 2022. [En línea]. Disponible en: <https://economipedia.com/definiciones/modelo-matematico.html>
- [51] J. P. Cardona, J. J. Leal, J. E. Ustariz, J. P. Cardona, J. J. Leal, y J. E. Ustariz, «Modelado matemático de caja blanca y negra en educación en ingeniería», *Form. Univ.*, vol. 13, n.º 6, pp. 105-118, dic. 2020, doi: 10.4067/S0718-50062020000600105.
- [52] A. Díez González, «Diseño e implementación en Aerostack de un servosistema para el control en velocidad del dron BeBop2». Accedido: 30 de enero de 2024. [En línea]. Disponible en: <https://oa.upm.es/56871/>
- [53] D. Rotondo, H. S. Sánchez, F. Nejari, y V. Puig, «Análisis y Diseño de Sistemas Lineales con Parámetros Variantes Utilizando LMIs», *Rev. Iberoam. Automática E Informática Ind.*, vol. 16, n.º 1, Art. n.º 1, ene. 2019, doi: 10.4995/riai.2018.10436.
- [54] J. L. A. Valverde, M. A. B. Pérez, C. R. C. Zavaleta, S. B. R. Rodríguez, J. H. A. Alcántara, y E. A. M. Ramos, «CONTROL ÓPTIMO LQR PARA ORIENTAR LA PLATAFORMA MÓVIL DE UN ROBOT PARALELO DE 2GDL EXPERIMENTAL», *Rev. Investig.*, vol. 10, n.º 4, Art. n.º 4, dic. 2021, doi: 10.26788/riepg.v10i4.3500.
- [55] H. González-Acevedo y O. G. Villamizar-Galvis, «Implementación de un sistema de control para regular la velocidad y posición de motores industriales utilizando el protocolo de comunicación OPC», *Rev. UIS Ing.*, vol. 18, n.º 2, Art. n.º 2, feb. 2019, doi: 10.18273/revuin.v18n2-2019014.

- [56] «Optimal tracking of the water level for a coupled tank system using Linear Quadratic Regulator | Tekhné». Accedido: 31 de enero de 2024. [En línea]. Disponible en: <https://revistasenlinea.saber.ucab.edu.ve/index.php/tekhne/article/view/5849>
- [57] R. Zapana Yerba, «Aproximación de las integrales definidas simples y compuestas mediante procesos estocásticos numéricos».
- [58] J. L. Romero Béjar, «Probabilidad - Variable aleatoria (Aspectos formales)», 2023, Accedido: 31 de enero de 2024. [En línea]. Disponible en: <https://digibug.ugr.es/handle/10481/84863>
- [59] «Algoritmo para analizar decisiones con objetivos múltiples bajo incertidumbre». Accedido: 31 de enero de 2024. [En línea]. Disponible en: https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-77432019000100010
- [60] I. Miller y J. E. Freund, *Probabilidad y estadística para ingenieros*. Reverte, 2021.
- [61] C. Mares Angulo, «Desarrollo del control de equilibrio y cinemática de un robot autobalanceado», dic. 2020, doi: 10/48831.
- [62] J. E. Pérez Valencia, «Construcción de un robot balancín controlado por una aplicación móvil», feb. 2021, Accedido: 29 de agosto de 2023. [En línea]. Disponible en: <http://dspace.istvidanueva.edu.ec/handle/123456789/116>
- [63] A. Sebastián Narváez, «Estimación de la velocidad y aceleración de un servomotor a partir de las medidas de un encoder incremental», Proyecto/Trabajo fin de carrera/grado, Universitat Politècnica de València, 2019. Accedido: 5 de mayo de 2023. [En línea]. Disponible en: <https://riunet.upv.es/handle/10251/130526>
- [64] R. A. T. Figueroa y W. Valderrama, «Control de velocidad y sentido de giro para un motor DC.», *Infometric - Ser. Ing. Básicas Agríc.*, vol. 3, n.º 1, Art. n.º 1, ago. 2020.
- [65] T. Guillermo, J. C. Cobos, y M. A. Chiluisa, *Microcontroladores*. PRODUCCIÓN DOCENTES UCE, 2022. Accedido: 5 de marzo de 2023. [En línea]. Disponible en: <https://librosdigitales.uce.edu.ec/index.php/PDU/catalog/book/8>
- [66] J. Albiter Jaimes, R. V. Mendoza Mendez, y E. J. Dorantes Coronado, «El pensamiento computacional en la electrónica: la importancia del software de simulación en la comprensión del principio de funcionamiento de los componentes electrónicos», *3C TIC Cuad. Desarro. Apl. Las TIC*, vol. 8, n.º 4, pp. 85-113, dic. 2019, doi: 10.17993/3ctic.2019.84.85-113.
- [67] O. G. Woge, C. O. G. Morán, y A. L. Chau, «Introducción al método del elemento finito: Solidworks y Matlab», *Ideas En Cienc. Ing.*, vol. 1, n.º 1, Art. n.º 1, jun. 2020.
- [68] S. Arévalo Fernández y A. Herreros López, «Aplicaciones de Matlab y Simulink para estaciones robóticas», presentado en XLII Jornadas de Automática, Universidade da Coruña, Servizo de Publicacións, 2021, pp. 625-631. Accedido: 19 de septiembre de 2023. [En línea]. Disponible en: <https://ruc.udc.es/dspace/handle/2183/28347>
- [69] J. Jiménez Alonso, «Esquema de comunicaciones avanzado para convertidores modulares», 2022, Accedido: 5 de mayo de 2023. [En línea]. Disponible en: <https://idus.us.es/handle/11441/139820>
- [70] A. S. Ismailov y Z. B. Jo‘rayev, «Study of arduino microcontroller board», *Sci. Educ.*, vol. 3, n.º 3, Art. n.º 3, mar. 2022.

- [71] E. Okyere, A. Bousbaine, G. T. Poyi, A. K. Joseph, y J. M. Andrade, «LQR controller design for quad-rotor helicopters», *J. Eng.*, vol. 2019, n.º 17, pp. 4003-4007, jun. 2019, doi: 10.1049/joe.2018.8126.
- [72] Y. Xin, H. Chai, Y. Li, X. Rong, B. Li, y Y. Li, «Speed and Acceleration Control for a Two Wheel-Leg Robot Based on Distributed Dynamic Model and Whole-Body Control», *IEEE Access*, vol. 7, pp. 180630-180639, 2019, doi: 10.1109/ACCESS.2019.2959333.
- [73] C.-F. Hsu, B.-R. Chen, y Z.-L. Lin, «Implementation and Control of a Wheeled Bipedal Robot Using a Fuzzy Logic Approach», *Actuators*, vol. 11, n.º 12, Art. n.º 12, dic. 2022, doi: 10.3390/act11120357.
- [74] E. Ackerman, «A Robot for the Worst Job in the Warehouse: Boston Dynamics' Stretch can move 800 heavy boxes per hour», *IEEE Spectr.*, vol. 59, n.º 1, pp. 50-51, ene. 2022, doi: 10.1109/MSPEC.2022.9676361.
- [75] I. Rey-Álvaro, «SIMULACIÓN Y CONTROL CINEMÁTICO DE ROBOTS MEDIANTE SOFTWARE SIMULINK.», mar. 2022, Accedido: 19 de septiembre de 2023. [En línea]. Disponible en: <http://crea.ujaen.es/jspui/handle/10953.1/16442>
- [76] «CON-1_C.pdf». Accedido: 5 de mayo de 2023. [En línea]. Disponible en: http://electro.itchiuhua.edu.mx/revista/2019/CON-1_C.pdf
- [77] A. A. C. Santiago, «DISEÑO Y CONSTRUCCIÓN DE UN ROBOT SEGUIDOR DE LÍNEA EVASOR DE OBSTÁCULOS EMPLEANDO ARDUINO NANO», 2020.
- [78] V. Luna Zempoalteca, «Un enfoque lagrangiano para el control basado en aprendizaje de robots manipuladores», ago. 2022, Accedido: 28 de julio de 2023. [En línea]. Disponible en: <https://hdl.handle.net/20.500.12371/17108>

ANEXOS

Anexo 1. Reporte de plagio



CERTIFICADO DE ANÁLISIS
magister

Tesis_Pillasagua Lídice

2%
Textos sospechosos

2% Similitudes
< 1% similitudes entre comillas
< 1% entre las fuentes mencionadas

< 1% Idiomas no reconocidos

Nombre del documento: Tesis_Pillasagua Lídice.pdf
ID del documento: b5c7bd1d2df4f967ed3cf048cb47e89221b35bf8
Tamaño del documento original: 2,22 MB

Depositante: CARLOS ALBERTO SALDAÑA ENDERICA
Fecha de depósito: 22/5/2024
Tipo de carga: interface
fecha de fin de análisis: 22/5/2024

Número de palabras: 30.752
Número de caracteres: 208.618

Ubicación de las similitudes en el documento:



Fuentes principales detectadas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	 repositorio.ucsm.edu.pe https://repositorio.ucsm.edu.pe/bitstream/20.500.12920/12050/5/74.0218.IE.pdf	< 1%		Palabras idénticas: < 1% (189 palabras)
2	 repositorio.uan.edu.co Construcción de prácticas de laboratorio bajo la iniciativa... http://repositorio.uan.edu.co/handle/123456789/2467	< 1%		Palabras idénticas: < 1% (27 palabras)
3	 detectando.com > Cómo funciona un péndulo y cuáles son sus principios básicos https://detectando.com/v-cmo-funciona-un-pendulo-y-cuales-son-sus-principios-basicos/	< 1%		Palabras idénticas: < 1% (21 palabras)

Fuentes con similitudes fortuitas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	 investigacionistct.ec https://investigacionistct.ec/ojs/index.php/investigacion_tecnologica/articulo/download/15/2/53	< 1%		Palabras idénticas: < 1% (30 palabras)
2	 dialnet.unirioja.es https://dialnet.unirioja.es/descarga/articulo/8383493.pdf	< 1%		Palabras idénticas: < 1% (25 palabras)
3	 dspace.ups.edu.ec http://dspace.ups.edu.ec/bitstream/123456789/22274/1/AUPS - TTS719.pdf	< 1%		Palabras idénticas: < 1% (36 palabras)
4	 repositorio.utn.edu.ec https://repositorio.utn.edu.ec/bitstream/123456789/12078/2/04_MEL_145_TRABAJO_GRADO.pdf	< 1%		Palabras idénticas: < 1% (19 palabras)
5	 bibdigital.epn.edu.ec http://bibdigital.epn.edu.ec/bitstream/15000/21956/1/CD_11447.pdf	< 1%		Palabras idénticas: < 1% (18 palabras)

Fuentes mencionadas (sin similitudes detectadas) Estas fuentes han sido citadas en el documento sin encontrar similitudes.

-  <https://www.elegoo.com/collections/arduino-learning>
-  <https://repositorio.comillas.edu/xmlui/handle/11531/61927>
-  <https://www.politesi.polimi.it/handle/10589/174465>
-  <http://dspace.ups.edu.ec/handle/123456789/19395>
-  <http://dspace.istvidanueva.edu.ec/bitstream/123456789/116/1/43.1470-PEREZ>

Anexo 2. Proceso de linealización de las ecuaciones

En este apartado se mostrará el proceso de linealización de las ecuaciones en espacio de estado. Se tomará como referencia las siguientes ecuaciones para el análisis correspondiente

$$\dot{x}_1 = \dot{x} = x_2 \quad 1$$

$$\dot{x}_2 = \dot{x} = -\frac{m_p l}{I} \ddot{\theta} \cos \theta + \frac{k_s k_t}{I r R} u - \left(\frac{k_e k_t}{r^2 R} - \frac{(\mu_s r^2 \mu_e)}{r^2 I} \right) \dot{x} + \left(\frac{k_e k_t}{I r R} + \frac{\mu_e}{I r} \right) \dot{\theta} \quad 2$$

$$\dot{x}_3 = \dot{\theta} = x_4 \quad 3$$

$$\dot{x}_4 = \ddot{\theta} = -\frac{m_p l}{J} \dot{x} \cos \theta + \frac{m_p g l}{J} \text{sen}(\theta) - \frac{k_s k_t}{J R} u + \left(\frac{k_e k_t}{J R r} + \frac{\mu_e}{J r} \right) \dot{x} - \left(\frac{k_e k_t}{J R r} + \frac{\mu_e}{J} \right) \dot{\theta} \quad 4$$

Para conocer cuáles son las condiciones en los cuales el sistema permanece en un estado constante sin cambiar con el tiempo, los puntos de equilibrio son los siguientes

$$x_1 = 0; x_2 = 0; x_3 = 0; x_4 = 0 \quad 5$$

Para el sistema de este trabajo las componentes en x son x_1, x_2, x_3 y x_4 , las funciones que se utilizaran para el análisis son las ecuaciones 1, 2, 3, 4, de esta manera se obtiene la matriz jacobiana A. Al aplicar el jacobiano a la primera ecuación se obtiene:

$$\frac{\partial f_1}{\partial x_1} = \frac{\partial x_2}{\partial x_1} = 0 \quad 6$$

$$\frac{\partial f_1}{\partial x_2} = \frac{\partial x_2}{\partial x_2} = 1 \quad 7$$

$$\frac{\partial f_1}{\partial x_3} = \frac{\partial x_2}{\partial x_3} = 0 \quad 8$$

$$\frac{\partial f_1}{\partial x_4} = \frac{\partial x_2}{\partial x_4} = 0 \quad 9$$

Jacobiano aplicado a la segunda ecuación:

$$\frac{\partial f_2}{\partial x_1} = \frac{\partial x_2}{\partial x_1} = 0 \quad 10$$

$$\frac{\partial f_2}{\partial x_2} = \frac{\partial x_2}{\partial x_2} = \frac{C_3 + C_1 * C_8}{C_1 * C_5 - 1} \quad 11$$

$$\frac{\partial f_2}{\partial x_3} = \frac{\partial x_2}{\partial x_3} = \frac{C_1 * C_6}{C_1 * C_5 - 1} \quad 12$$

$$\frac{\partial f_2}{\partial x_4} = \frac{\partial x_2}{\partial x_4} = -\frac{C_4 + C_1 * C_9}{C_1 * C_5 - 1} \quad 13$$

Jacobiano aplicado a la tercera ecuación:

$$\frac{\partial f_3}{\partial x_1} = \frac{\partial x_4}{\partial x_1} = 0 \quad 14$$

$$\frac{\partial f_3}{\partial x_2} = \frac{\partial x_4}{\partial x_2} = 0 \quad 15$$

$$\frac{\partial f_3}{\partial x_3} = \frac{\partial x_4}{\partial x_3} = 0 \quad 16$$

$$\frac{\partial f_3}{\partial x_4} = \frac{\partial x_4}{\partial x_1} = 0 \quad 17$$

Jacobiano aplicado a la cuarta ecuación:

$$\frac{\partial f_4}{\partial x_1} = \frac{\partial x_4}{\partial x_1} = 0 \quad 18$$

$$\frac{\partial f_4}{\partial x_2} = \frac{\partial x_4}{\partial x_2} = -\frac{C_2 + C_1 * C_7}{C_1 * C_5 - 1} \quad 19$$

$$\frac{\partial f_4}{\partial x_3} = \frac{\partial x_4}{\partial x_3} = 0 \quad 20$$

$$\frac{\partial f_4}{\partial x_4} = \frac{\partial x_4}{\partial x_1} = \frac{0}{C_1 * C_5 - 1} \quad 21$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{C_3 + C_1 * C_8}{C_1 * C_5 - 1} & \frac{C_1 * C_6}{C_1 * C_5 - 1} & -\frac{C_4 + C_1 * C_9}{C_1 * C_5 - 1} \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{C_8 + C_3 * C_5}{C_1 * C_5 - 1} & -\frac{C_6}{C_1 * C_5 - 1} & \frac{C_9 + C_4 * C_5}{C_1 * C_5 - 1} \end{bmatrix}$$

La matriz jacobiana B tendrá el mismo proceso considerando las variables adecuadas para su desarrollo

$$\frac{\partial f_1}{\partial x} = \frac{\partial x_1}{\partial x} = 0 \quad 22$$

$$\frac{\partial f_2}{\partial x} = \frac{\partial x_2}{\partial x} = -\frac{C_2 + C_1 * C_7}{C_1 * C_5 - 1} \quad 23$$

$$\frac{\partial f_3}{\partial x} = \frac{\partial x_3}{\partial x} = 0 \quad 24$$

$$\frac{\partial f_4}{\partial x} = \frac{\partial x_4}{\partial x} = \frac{C_7 + C_2 * C_5}{C_1 * C_5 - 1} \quad 25$$

$$B = \begin{bmatrix} 0 \\ -\frac{C_2 + C_1 * C_7}{C_1 * C_5 - 1} \\ 0 \\ \frac{C_7 + C_2 * C_5}{C_1 * C_5 - 1} \end{bmatrix}$$

Para la matriz jacobiana C se tomará en cuenta las variables necesarias, quedando de la siguiente manera para la primera ecuación:

$$\frac{\partial f_1}{\partial x} = \frac{\partial x_1}{\partial x} = 1 \quad 26$$

$$\frac{\partial f_2}{\partial x} = \frac{\partial x_2}{\partial x} = 0 \quad 27$$

$$\frac{\partial f_3}{\partial x} = \frac{\partial x_3}{\partial x} = 0 \quad 28$$

$$\frac{\partial f_4}{\partial x} = \frac{\partial x_4}{\partial x} = 0 \quad 29$$

Jacobiano aplicado a la segunda ecuación:

$$\frac{\partial f_1}{\partial \theta} = \frac{\partial x_1}{\partial x} = 0 \quad 30$$

$$\frac{\partial f_2}{\partial \theta} = \frac{\partial x_2}{\partial x} = 1 \quad 31$$

$$\frac{\partial f_3}{\partial \theta} = \frac{\partial x_3}{\partial x} = 0 \quad 32$$

$$\frac{\partial f_4}{\partial \theta} = \frac{\partial x_4}{\partial x} = 0 \quad 33$$

Jacobiano aplicado a la tercera ecuación:

$$\frac{\partial f_1}{\partial \theta} = \frac{\partial x_1}{\partial x} = 0 \quad 34$$

$$\frac{\partial f_2}{\partial \theta} = \frac{\partial x_2}{\partial x} = 0 \quad 35$$

$$\frac{\partial f_3}{\partial \theta} = \frac{\partial x_3}{\partial x} = 1 \quad 36$$

$$\frac{\partial f_4}{\partial \theta} = \frac{\partial x_4}{\partial x} = 0 \quad 37$$

Jacobiano aplicado a la cuarta ecuación:

$$\frac{\partial f_1}{\partial \theta} = \frac{\partial x_1}{\partial x} = 0 \quad 38$$

$$\frac{\partial f_2}{\partial \theta} = \frac{\partial x_2}{\partial x} = 0 \quad 39$$

$$\frac{\partial f_3}{\partial \theta} = \frac{\partial x_3}{\partial x} = 0 \quad 40$$

$$\frac{\partial f_4}{\partial \theta} = \frac{\partial x_4}{\partial x} = 1 \quad 41$$

Para la matriz D queda de la siguiente manera:

$$\frac{\partial f_1}{\partial \theta} = \frac{\partial x_1}{\partial x} = 0 \quad 42$$

$$\frac{\partial f_2}{\partial \theta} = \frac{\partial x_2}{\partial x} = 0 \quad 43$$

$$\frac{\partial f_3}{\partial \theta} = \frac{\partial x_3}{\partial x} = 0 \quad 44$$

$$\frac{\partial f_4}{\partial \theta} = \frac{\partial x_4}{\partial x} = 0 \quad 45$$

Anexo 3. Código en matlab para la obtención de matrices de estado mediante jacobiano.

```
% Parámetros y constantes
syms C1 C2 C3 C4 C5 C6 C7 C8 C9 x1 x2 x3 x4 u

% Definir las ecuaciones del sistema
ecu1 = x2
ecu2 = -(C1 * C6 * sin(x3) * cos(x3)) / (1 - C5 * C1 * (cos(x3))^2) + ((C1 * C7 * cos(x3) + C2) / (1 - C5 * C1 * (cos(x3))^2)) * u - ((C1 * C8 * cos(x3) + C3) / (1 - C5 * C1 * (cos(x3))^2)) * x2 + ((C1 * C9 * cos(x3) + C4 - C1 * C5 * sin(x3) * cos(x3) * x2) / (1 - C5 * C1 * (cos(x3))^2)) * x4;

ecu3 = x4
ecu4 = -(C2 * C5 * cos(x3) + C7) / (1 - C1 * C5 * (cos(x3))^2) * u + ((C3 * C5 * cos(x3) + C8 + C5 * sin(x3) * x4) / (1 - C1 * C5 * (cos(x3))^2)) * x2 - ((C4 * C5 * cos(x3) + C9) / (1 - C1 * C5 * (cos(x3))^2)) * x4 + (C6 * sin(x3)) / (1 - C1 * C5 * (cos(x3))^2);

% Derivar parcialmente las ecuaciones para obtener matrices A y B
A = jacobian([ecu1; ecu2; ecu3; ecu4], [x1, x2, x3, x4])
B = jacobian([ecu1; ecu2; ecu3; ecu4], u)

% Establecer cero en x1 x2 x3 x4 en las matrices A y B
A = subs(A, [x1,x2,x3,x4],[0,0,0,0]);
B = subs(B, [x1,x2,x3,x4] ,[0,0,0,0]);

disp('Matriz A de estados (con puntos de equilibrio):')
disp(A);

disp('Matriz B de estados (sin punto de equilibrio):')
disp(B);
```

Anexo 4. Código en Matlab para la inestabilidad.

```
%Autor:Lidice Pillasagua
%Parámetros del sistema
clc,close all,clear all
lp=0.65; %Longitud del péndulo
m_r=0.0908; %Masa del carrito
m_p=0.752; %masa del péndulo
ue=0.25; %coeficiente de fricción del carro
us=0.018; %coeficiente de fricción del péndulo
r = 0.045;
g=9.81; %aceleración de la gravedad
l = 0.18;
n = 1/30; % Relación de trnsmisión
k_t = 0.017; %0.008; % [Nm/A] Constante de torque de un
motor dc
k_E = 0.0342857; %0.27; % [Vsec/rad] Constante de fuerza
contraelectromotriz
V=12;
ue = 0.08;
k_s = (V/ue);
J_p = 0.049
R = 0.0024; %2.17; % [ohm] Resistencia del motor
us = 0.01
J_r = 0.000172125
I=(m_r+(J_r/r^2) +m_p)
J=(m_p*l^2)+J_p
%Análisis Inestabilidad
% Calcula los valores propios (polos) de la matriz A
polos = eig(A);
% Verifica la estabilidad del sistema
estable = all(real(polos) < 0); % Verifica si todas las partes reales
son negativas
% Muestra los polos
disp('Polos del sistema:')
disp(polos)
% Muestra el mensaje de estabilidad
if estable
    disp('Estable. ');
else
    disp('Inestable. ');
end
end
```


Anexo 5. Código en Matlab para la controlabilidad.

```
%Autor:Lidice Pillasagua
%Parámetros del sistema
clc,close all,clear all
lp=0.65; %Longitud del péndulo
m_r=0.0908; %Masa del carrito
m_p=0.752; %masa del péndulo
ue=0.25; %coeficiente de fricción del carro
us=0.018; %coeficiente de fricción del péndulo
r = 0.045;
g=9.81; %aceleración de la gravedad
l = 0.18;
n = 1/30; % Relación de trnsmisión
k_t = 0.017; %0.008; % [Nm/A] Constante de torque de un
motor dc
k_E = 0.0342857; %0.27; % [Vsec/rad] Constante de fuerza
contraelectromotriz
V=12;
ue = 0.08;
k_s = (V/ue);
J_p = 0.049
R = 0.0024; %2.17; % [ohm] Resistencia del motor
us = 0.01
J_r = 0.000172125
I=(m_r+(J_r/r^2) +m_p)
J=(m_p*l^2)+J_p
% SERVO-CONTROLADOR LQR CON SEGUIMIENTO A LA REFERENCIA
%Matrices ampliadas, se agrega el integrador
A_1=[A zeros(4,1);-C2 0]
B_1=[B;0]
C_1=[C2 0]
D_1=[0]
sys_a=ss(A_1,B_1,C_1,D_1)
ctrb_a=[B_1 A_1*B_1 (A_1^2)*B_1 (A_1^3)*B_1]
rangctrb=rank(ctrb_a)
%Matriz de Controlabilidad del sistema aumentado (P)
Pp=[A_1 B_1;-C_1 0]
rangP=rank(Pp)
%Rango de P n=5, el sistema aumentado es controlable
rangA=rank(A_1); %Rango de la matriz A aumentada n=3
rangA_p1=rank(A); %Rango de la matriz A original n=3
```

Anexo 6. Código Matlab para la Observabilidad.

```
%Autor:Lidice Pillasagua
%Parámetros del sistema
clc,close all,clear all
lp=0.65; %Longitud del péndulo
m_r=0.0908; %Masa del carrito
m_p=0.752; %masa del péndulo
ue=0.25; %coeficiente de fricción del carro
us=0.018; %coeficiente de fricción del péndulo
r = 0.045;
g=9.81; %aceleración de la gravedad
l = 0.18;
n = 1/30; % Relación de trnsmisión
k_t = 0.017; %0.008; % [Nm/A] Constante de torque de un
motor dc
k_E = 0.0342857; %0.27; % [Vsec/rad] Constante de fuerza
contraelectromotriz
V=12;
ue = 0.08;
k_s = (V/ue);
J_p = 0.049
R = 0.0024; %2.17; % [ohm] Resistencia del motor
us = 0.01
J_r = 0.000172125
I=(m_r+(J_r/r^2) +m_p)
J=(m_p*l^2)+J_p
%% Comprobación de Observabilidad
obs=[C;C*A;C*(A^2);C*(A^3)];
if rank(obs)==4
    disp('El Sistema es Observable.')
else
    disp('El Sistema no es Observable.')
end
```

Anexo 7. Código en Matlab del controlador LQR

```

%Autor:Lidice Pillasagua
%Parámetros del sistema
clc,close all,clear all
lp=0.65; %Longitud del péndulo
m_r=0.0908; %Masa del carrito
m_p=0.752; %masa del péndulo
ue=0.25; %coeficiente de fricción del carro
us=0.018; %coeficiente de fricción del péndulo
r = 0.045;
g=9.81; %aceleración de la gravedad
l = 0.18;
n = 1/30; % Relación de trnsmisión
k_t = 0.017; %0.008; % [Nm/A] Constante de torque de un motor dc
k_E = 0.0342857; %0.27; % [Vsec/rad] Constante de fuerza contraelectromotriz
V=12;
ue = 0.08;
k_s = (V/ue);
J_p = 0.049;
R = 0.0024; %2.17; % [ohm] Resistencia del motor
us = 0.01;
J_r = 0.000172125;
l=(m_r+(J_r/r^2) +m_p);
J=(m_p*l^2)+J_p;
C1=(m_p*l)/l
C2=(k_s*k_t)/l*r*R
C3=((k_E*k_t)/(r^2*R))-(((us*r^2*ue ))/(r^2*l))
C4=((k_E*k_t)/l*r*R)+(ue/l*r)
C5=(m_p*l)/J
C6=(m_p*g*l)/J
C7= (k_s*k_t)/J*R
C8=((k_E*k_t)/J*R*r)+(ue/J_r)
C9=((k_E*k_t)/J*R*r)+(ue/J)
%%
%Matrices linealizadas
A=[0 1 0 0;
    0 (C3+(C1*C8)/((C1*C5)-1)) ((C1*C6)/((C1*C5)-1)) -((C4+(C1*C9))/(C1*C5-1));
    0 0 0 1;
    0 -(C8+(C3*C5)/((C1*C5)-1)) -(C6/((C1*C5)-1)) (C9+(C4*C5)/((C1*C5)-1))]
B=[0;
    -C2+(C1*C7) / ((C1*C5)-1);
    0;
    (C7+(C1*C5) )/ ((C1*C5)-1)]
C=[1 0 0 0;0 0 1 0]
C2=[1 0 0 0]
%%
%Para el control LQR con seguimiento a la referencia , se considera
% la posición del carro x
D=[0;0];
%Modelo en el espacio de estados
disp('Modelo del sistema en espacio de estados ')
model_2=ss(A,B,C,D)
%Modelo discretizado
disp('Modelo discreto en espacio de estados ');
model_2d=c2d(model_2,0.1,'T');

```

```

%%
%Análisis Inestabilidad
% Calcula los valores propios (polos) de la matriz A
polos = eig(A);
% Verifica la estabilidad del sistema
estable = all(real(polos) < 0); % Verifica si todas las partes reales son negativas
% Muestra los polos
disp('Polos del sistema:');
disp(polos);
% Muestra el mensaje de estabilidad
if estable
    disp('Estable.')
else
    disp('Inestable.')
end
%Anansisi Bibo
%Tiempo continuo
[n1,d1]=ss2tf(A,B,C,D)
n1a=n1(1,:);
n1b=n1(2,:);
gx=tf(n1a,d1) %Funcion de transferencia X(s)/F(s)
zpk(gx)
gt=tf(n1b,d1) %Funcion de transfrencia Theta(s)/F(s)
zpk(gt)
impulse(model_2) %Respuesta al impulso del sistema inestable
step(model_2)
grid on
%Tiempo discreto
[n1,d1]=ss2tf(model_2d.A,model_2d.B,model_2d.C,model_2d.D)
n1ad=n1(1,:);
n1bd=n1(2,:);
gxd=tf(n1ad,d1,0.1) %Función de transferencia X(z)/F(z)
zpk(gxd)
gtd=tf(n1bd,d1,0.1) %Función de transferencia Theta(z)/F(z)
zpk(gtd)
impulse(model_2d) %Respuesta al impulso del sistema discreto
step(model_2d)
grid on
%Analisis de Inestabilidad Interna
%Tiempo continuo
x0=[0.25;0;pi/18;0];
initial(model_2,x0)
grid on
eig(A)
%Tiempo discreto
x0=[0.25;0;pi/18;0];
initial(model_2d,x0)
grid on
eig(model_2d.A)
%%
% SERVO-CONTROLADOR LQR CON SEGUIMIENTO A LA REFERENCIA
%Matrices ampliadas, se agrega el integrador
A_1=[A zeros(4,1);-C2 0]
B_1=[B;0]
C_1=[C2 0]
D_1=[0]
sys_a=ss(A_1,B_1,C_1,D_1)

```

```

ctrb_a=[B_1 A_1*B_1 (A_1^2)*B_1 (A_1^3)*B_1]
rangoctrb=rank(ctrb_a)
%Matriz de Controlabilidad del sistema aumentado (P)
Pp=[A_1 B_1;-C_1 0]
rangP=rank(Pp)
%Rango de P n=5, el sistema aumentado es controlable
rangA=rank(A_1); %Rango de la matriz A aumentada n=3
rangA_p1=rank(A); %Rango de la matriz A original n=3
%Diseño LQR para sistema aumentado para 1 salida x, posición del carro
Q_lqr=[20 0 0 0 0;0 6 0 0 0;0 0 50 0 0;0 0 0 6 0;0 0 0 0 15]
R_lqr=0.5;
[k_lqr,S,e]=lqr(A_1,B_1,Q_lqr,R_lqr)
disp('***Vector de Control Optimo*** ')
k_lqr=k_lqr(1:end-1) %Vector óptimo de regulación de estados
k_i=k_lqr(end) %Vector de regulación del error(Ki) - SEGUIMIENTO

disp('Autovalores del sistema realimentado ')
eig(A_1 - B_1*k_lqr)
%Sistema de control LQR con seguimiento
disp('Sistema de Control LQR Realimentado')
sys_r=ss(A_1 - B_1*k_lqr,B_1,C_1,D_1)

```

Anexo 8. Código en Arduino del control LQR

```
/*Control LQR para robot con autoequilibrio.*/

#include "Motor.h"
#include "LQR.h"

Temporizador Temporizador;
extern Mpu6050 Mpu6050;
extern Motor Motor;
extern LQR LQR;
extern TestAngulos TestAngulos;
/*Declaración de las constantes de realimentación*/
LQR::LQR(){

/*  theta , theta_dot , x_dot , x , omega ,
omega_dot*/
  K[0] = 150, K[1] = 0.05, K[2] = 15, K[3] = 0.24, K[4] = 2.5, K[5] =
0.5;
}

void setup(){

  Motor.IniciarPines();
  Motor.IniciarEncoder();
  Temporizador.iniciar(TIMER);
  Mpu6050.iniciar();
  Serial.begin(9600);
  delay(100);
}

void loop(){

  int bufer_direccion[] = { ADELANTE,
                           REVERSA,
                           IZQUIERDA,
                           DERECHA,
                           DETENER};

  static unsigned long imprimir_tiempo;

  if(millis() - imprimir_tiempo > 500){
    imprimir_tiempo = millis();
    LQR.Control_de_Movimiento(bufer_direccion[4]);
  }
  Serial.print(LQR.pwm_izquierda);
  Serial.print(',');
  Serial.println(TestAngulos.Angulo_RW);
}
```

Anexo 9. Extensión del Código en Arduino para la obtención del ángulo.

```
#include "Angulo.h"

//////////////////////////////////// Prueba de
ángulo////////////////////////////////////
void TestAngulos::Angletest(int16_t ax,int16_t ay,int16_t az,int16_t
gx,int16_t gy,int16_t gz,float dt,float Q_angulo,float Q_giro,
float R_angulo,float C_0,float K1)
{
  // int flag;
  //parámetro de saldo
  //180/π = 57. 3;
  float Angulo = atan2(ay , az) * 57.3; //Fórmula de cálculo del
ángulo, Angle: el ángulo de inclinación final del
  Angulo_RW = Angulo;
  //automóvil calculado por el filtro complementario de primer orden

  Giro_x = (gx - 128.1) / 131; //conversión de ángulo

  //parámetro del eje Z del ángulo de rotación
  if (gz > 32768) gz -= 65536; //forzar conversión 2g 1g
  Giro_z = -gz / 131; //Conversión de parámetros
del eje Z
  acelz = az / 16.4;

  float angleAx = atan2(ax, az) * 180 / PI; //calcular el ángulo con el
eje x
  Giro_y = -gy / 131.00; //calcula la velocidad angular
}
```

Anexo 10. Extensión del Código en Arduino para la obtención del ángulo.

```
#ifndef Angulo_h
#define Angulo_h

#if defined(ARDUINO) && (ARDUINO >= 100)
#include <Arduino.h>
#else
#include <WProgram.h>
#endif

class TestAngulos{
public:

    void Angletest(int16_t ax,int16_t ay,int16_t az,int16_t gx,int16_t
gy,int16_t gz,float dt,float Q_angulo,float Q_giro,
                float R_angulo,float C_0,float K1); /*float C_1*/
    float Giro_x,Giro_y,Giro_z;
    float acelz = 0;

    float Angulo_RW;
};
#endif
```


Anexo 11. Código en Arduino del LQR para estabilización.

```
#include "LQR.h"
#include "Wire.h"
#include "Motor.h"
#include "MPU6050.h"
#include "Angulo.h"

MPU6050 MPU6050;
Mpu6050 Mpu6050;
LQR LQR;
TestAngulos TestAngulos;
Motor Motor;

void Temporizador::iniciar(int tiempo)
{
  MsTimer2::set(tiempo, interrupcion);
  MsTimer2::start();
}

static void Temporizador::interrupcion(){

  sei();//Habilitar las interrupciones Globales
  LQR.Obtener_VelocidadEncoder();
  Mpu6050.ProcesarDatos();
  LQR.Estado_Theta_dot();

  LQR.contador_interrupciones++;
  if(LQR.contador_interrupciones > 8){
    LQR.contador_interrupciones = 0;
    LQR.Estado_x_dot();
    LQR.Estado_omega_dot();
  }

  LQR.Control_Total();
}

void LQR::Control_Total()
{
  pwm_izquierda = balance_control_output - speed_control_output -
  rotation_control_output;//Superposition of Vertical Velocity Steering
  Ring
  pwm_derecha = balance_control_output - speed_control_output +
  rotation_control_output;//Superposition of Vertical Velocity Steering
  Ring

  pwm_izquierda = constrain(pwm_izquierda, -255, 255);
  pwm_derecha = constrain(pwm_derecha, -255, 255);

  /*while(EXCESSIVE_ANGLE_TILT || PICKED_UP)
  {
    Mpu6050.ProcesarDatos();
    Motor.Detener();
  }
  */
}
```

```

}*/

(pwm_izquierda < 0) ? (Motor.Control(AIN1,1,PWMA_IZQUIERDA,-
pwm_izquierda)):
    (Motor.Control(AIN1,0,PWMA_IZQUIERDA,pwm_izquierda));

(pwm_derecha < 0) ? (Motor.Control(BIN1,1,PWMB_DERECHA,-pwm_derecha)):
    (Motor.Control(BIN1,0,PWMB_DERECHA,pwm_derecha));
}

void LQR::Obtener_VelocidadEncoder()
{
    encoder_izquierda_pulso_num_velocidad += pwm_izquierda < 0 ? (-
Motor::contador_encoder_izquierda_a) :

Motor::contador_encoder_izquierda_a;

    encoder_derecha_pulso_num_velocidad += pwm_derecha < 0 ? (-
Motor::contador_encoder_derecha_a) :

Motor::contador_encoder_derecha_a;
    Motor::contador_encoder_izquierda_a=0;
    Motor::contador_encoder_derecha_a=0;
}

void LQR::Control_de_Movimiento(Direccion direccion)
{
    switch(direccion)
    {
        case DETENER:
            Detener();break;
        case ADELANTE:
            Adelante(40);break;
        case REVERSA:
            Reversa(40);break;
        case IZQUIERDA:
            Izquierda(50);break;
        case DERECHA:
            Derecha(50);break;
        default:
            Detener();break;
    }
}

void LQR::Detener()
{
    ajuste_velocidad_robot = 0;
    ajuste_velocidad_de_giro = 0;
}

void LQR::Adelante(int velocidad)
{
    ajuste_velocidad_robot = velocidad;
    ajuste_velocidad_de_giro = 0;
}

```

```

void LQR::Reversa(int velocidad)
{
    ajuste_velocidad_robot = -velocidad;
    ajuste_velocidad_de_giro = 0;
}

void LQR::Izquierda(int velocidad)
{
    ajuste_velocidad_robot = 0;
    ajuste_velocidad_de_giro = velocidad;
}

void LQR::Derecha(int velocidad)
{
    ajuste_velocidad_robot = 0;
    ajuste_velocidad_de_giro = -velocidad;
}

void LQR::Estado_x_dot()
{
    double velocidad_robot=(encoder_izquierda_pulso_num_velocidad +
encoder_derecha_pulso_num_velocidad) * 0.5;
    encoder_izquierda_pulso_num_velocidad = 0;
    encoder_derecha_pulso_num_velocidad = 0;

    filtro_velocidad = filtro_velocidad_antiguo * 0.7 + velocidad_robot *
0.3; /*Obtención de velocidad óptima con ponderaciones*/

    filtro_velocidad_antiguo = filtro_velocidad;
    integral_velocidad_robot += filtro_velocidad;
    integral_velocidad_robot = integral_velocidad_robot -
ajuste_velocidad_robot;
    integral_velocidad_robot = constrain(integral_velocidad_robot, -3000,
3000);

    speed_control_output = - K[2] * filtro_velocidad - K[3] *
integral_velocidad_robot;
}

void LQR::Estado_Theta_dot()
{
    balance_control_output = K[0] * (TestAngulos.Angulo_RW - 0) + K[1] *
(TestAngulos.Giro_x - 0);
}

void LQR::Estado_omega_dot()
{
    rotation_control_output = ajuste_velocidad_de_giro + K[5] *
TestAngulos.Giro_z;////control with Z-axis gyroscope
}

void Mpu6050::iniciar()
{
    Wire.begin();
    MPU6050.initialize();
}

```

```
}  
  
Mpu6050::Mpu6050()  
{  
    dt = 0.005, Q_angulo = 0.001, Q_giro = 0.005, R_angulo = 0.5, C_0 =  
    1, K1 = 0.05;  
}  
  
void Mpu6050::ProcesarDatos()  
{  
    MPU6050.getMotion6(&ax, &ay, &az, &gx, &gy, &gz); // Adquisición de  
    datos del giroscopio y acelerómetro  
    TestAngulos.Angletest(ax, ay, az, gx, gy, gz, dt, Q_angulo, Q_giro,  
    R_angulo, C_0, K1); // Obtaining Angle by Kalman Filter  
}
```

Anexo 12. Clases y funciones necesarias para implementar el control LQR.

```
#ifndef _LQR_h
#define _LQR_h

#include "MsTimer2.h"
#include "Angulo.h"

enum Direccion
{
    ADELANTE,
    REVERSA,
    IZQUIERDA,
    DERECHA,
    DETENER,
};

class LQR
{
public:
    LQR();
    void Obtener_VelocidadEncoder();

    /*Estados del Sistema*/
    /*[theta; theta_dot ; x ; x_dot ; omega ; omega_dot */
    void Estado_Theta_dot(); /*Estado de Inclinación & velocidad
de Inclinación*/
    void Estado_x_dot(); /*Estado de posición en x del robot &
velocidad*/
    void Estado_omega_dot(); /*Estado de guiñada & velocidad de
guiñada */

    /*Control LQR total*/
    void Control_Total();

    /*Movimiento del Robot*/
    void Control_de_Movimiento(Direccion direccion);
    void Detener();
    void Adelante(int velocidad);
    void Reversa(int velocidad);
    void Izquierda(int velocidad);
    void Derecha(int velocidad);

    double pwm_izquierda;
    double pwm_derecha;
    int encoder_izquierda_pulso_num_velocidad;
    int encoder_derecha_pulso_num_velocidad;

    /*Contador de interrupciones*/
    int contador_interrupciones;

    /*Constantes de LQR*/
};
```

```

/*PD_VerticalRing*/

/*[   theta; --> K[0]
      theta_dot; --> K[1]
          x; --> K[2]
          x_dot; --> K[3]
      ]*/
double K[6] = {0, 0, 0, 0, 0, 0}; //Matriz que contiene las
constantes LQR

double filtro_velocidad;
double filtro_velocidad_antiguo;
double integral_velocidad_robot;
double speed_control_output;

double balance_control_output;
double rotation_control_output;

/*Referencias del sistema*/
int ajuste_velocidad_de_giro;
int ajuste_velocidad_robot;

private:
#define ANGLE_MIN -27
#define ANGLE_MAX 27
#define EXCESSIVE_ANGLE_TILT (FiltroKalman.angulo < ANGLE_MIN ||
ANGLE_MAX < FiltroKalman.angulo)
#define PICKED_UP (FiltroKalman.angle6 < -10 || 22 <
FiltroKalman.angle6)
};

class Mpu6050
{
public:
    void iniciar();
    void ProcesarDatos();
    Mpu6050();

public:
    int ax, ay, az, gx, gy, gz;
    float dt, Q_angulo, Q_giro, R_angulo, C_0, K1;
};

class Temporizador
{
public:
    void iniciar(int tiempo);
    static void interrupcion();
private:
    #define TIMER 5
};

#endif

```

Anexo 13. Funcionalidad de control de motores del robot balancín.

```
#include <Arduino.h>
#include "Motor.h"

void Motor::IniciarPines()
{
  pinMode(AIN1, OUTPUT);
  pinMode(BIN1, OUTPUT);
  pinMode(PWMA_IZQUIERDA, OUTPUT);
  pinMode(PWMB_DERECHA, OUTPUT);
  pinMode(STBY_PIN, OUTPUT);
  digitalWrite(STBY_PIN, HIGH);
}

Motor::Motor()
{
  // MOVE[0] = &Motor::Forward;
  // MOVE[1] = &Motor::Back;
  // MOVE[2] = &Motor::Left;
  // MOVE[3] = &Motor::Right;
}

void Motor::Detener()
{
  analogWrite(PWMA_IZQUIERDA, 0);
  analogWrite(PWMB_DERECHA, 0);
}

void Motor::Adelante(int velocidad)
{
  digitalWrite(AIN1, 0);
  digitalWrite(BIN1, 0);
  analogWrite(PWMA_IZQUIERDA, velocidad);
  analogWrite(PWMB_DERECHA, velocidad);
}

void Motor::Reversa(int velocidad)
{
  digitalWrite(AIN1, 1);
  digitalWrite(BIN1, 1);
  analogWrite(PWMA_IZQUIERDA, velocidad);
  analogWrite(PWMB_DERECHA, velocidad);
}

void Motor::Izquierda(int velocidad)
{
  digitalWrite(AIN1, 1);
  digitalWrite(BIN1, 1);
  analogWrite(PWMA_IZQUIERDA, velocidad);
  analogWrite(PWMB_DERECHA, 0);
}

void Motor::Derecha(int velocidad)
{

```

```
digitalWrite(AIN1, 1);
digitalWrite(BIN1, 1);
analogWrite(PWMA_IZQUIERDA,0);
analogWrite(PWMB_DERECHA,velocidad);
}

void Motor::Control(int PIN,int PIN_value,int PWM_pin,int velocidad)
{
digitalWrite(PIN, PIN_value);
analogWrite(PWM_pin,velocidad);
}
```


Anexo 14. Funciones para controlar motores y medir la velocidad de los encoders.

```
#ifndef _MOTOR_H
#define _MOTOR_H

class Motor
{
public:
    Motor();

    void IniciarPines();
    /*Medición de Velocidad*/
    void IniciarEncoder();

    // void (Motor::*MOVE[4])(int velocidad);
    void Control(int valor_AIN1,int valor_BIN1,int pin_PWM,int
velocidad);

    void Detener();
    void Adelante(int velocidad);
    void Reversa(int velocidad);
    void Izquierda(int velocidad);
    void Derecha(int velocidad);

public:
    static unsigned long contador_encoder_derecha_a;
    static unsigned long contador_encoder_izquierda_a;

private:
    /*Pines de los motores*/
    #define AIN1 7
    #define PWMA_IZQUIERDA 5
    #define BIN1 12
    #define PWMB_DERECHA 6
    #define STBY_PIN 8

    /*Medición de los encoders*/
    #define PIN_ENCODER_IZQUIERDA_A 2
    #define PIN_ENCODER_DERECHA_A 3

};

#endif
```

Anexo 15. Configuración de interrupciones para contar los pulsos de los encoders.

```
#include "Motor.h"
#include "PinChangeInt.h"

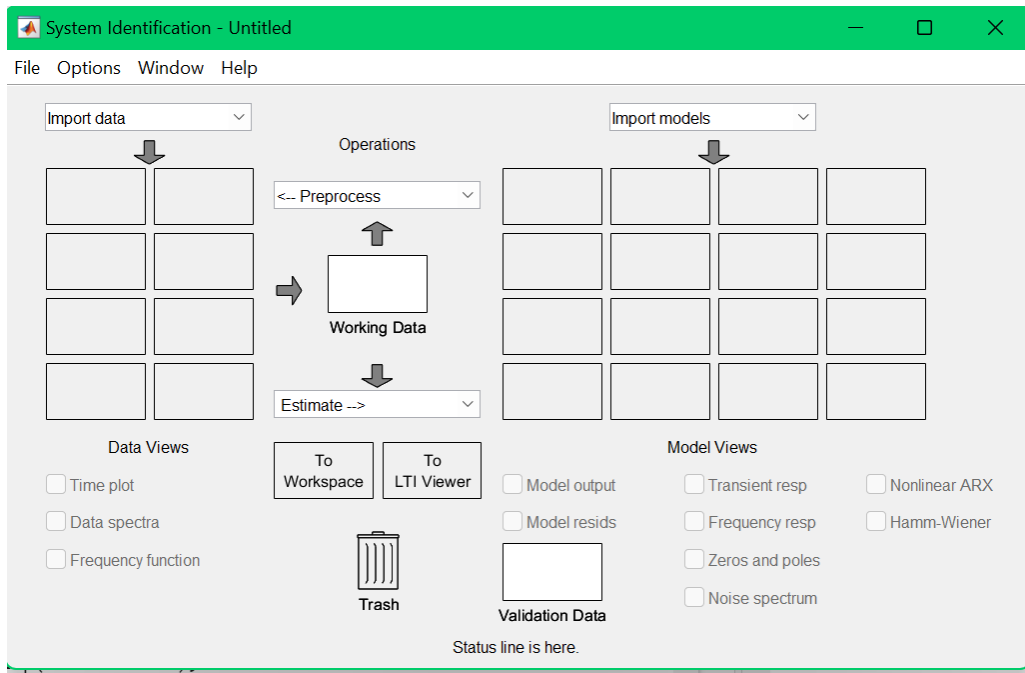
static void ContadorEncoderDerechaA();
static void ContadorEncoderIzquierdaA();

void Motor::IniciarEncoder()
{
    attachInterrupt(digitalPinToInterrupt(PIN_ENCODER_IZQUIERDA_A),
ContadorEncoderIzquierdaA, CHANGE);
    attachInterrupt(digitalPinToInterrupt(PIN_ENCODER_DERECHA_A),
ContadorEncoderDerechaA, CHANGE);
}

unsigned long Motor::contador_encoder_derecha_a;
//Conseguir velocidad en la rueda derecha.
static void ContadorEncoderDerechaA()
{
    Motor::contador_encoder_derecha_a++;
}

unsigned long Motor::contador_encoder_izquierda_a;
//Conseguir velocidad en la rueda izquierda.
static void ContadorEncoderIzquierdaA()
{
    Motor::contador_encoder_izquierda_a++;
}
```

Anexo 16. Herramienta System Identification.



Anexo 17. Control del robot balancín.

