



UNIVERSIDAD ESTATAL PENINSULA 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:

INGENIERA EN ELECTRÓNICA Y TELECOMUNICACIONES

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL LINEAL
CUADRÁTICO GAUSSIANO EN UN SISTEMA NO LINEAL
ELECTROMECAÁNICO DENOMINADO ROBOT BALANCÍN”

AUTOR

YURI CLARIBEL TORRES JOSÉ

TUTOR

ING. CARLOS ALBERTO SALDAÑA ENDERICA

LA LIBERTAD – ECUADOR

2024

AGRADECIMIENTO

Primero resalto el nombre de Dios por ser el héroe de esta historia, su amor infinito ha convertido en realidad los sueños que una vez parecían imposible, sinceros agradecimientos por todo lo que ha hecho en mi vida.

A mi familia gracias por entender y apoyar mis sueños. Este logro no solo es mío, sino nuestro. ¡Los amo esposo e hijos!

“Gratitud al maestro que alumbra nuestra vida”, dice una frase de aquel himno al maestro; hoy tomo aquella frase para agradecer a mi estimado tutor Ingeniero Carlos Saldaña, su orientación experta fue una luz en mis conocimientos referentes a la teoría de control. Su compromiso y dedicación en mi crecimiento académico han sido invaluable, permitiéndome superar obstáculos y alcanzar un nuevo nivel de comprensión en controles automáticos.

Para finalizar, agradezco a los docentes de la Facultad de Sistemas y Telecomunicaciones, cuyo impacto en mi formación académica fue la luz de mi bello sueño al despertar. Hoy como ingeniera de la Universidad Estatal Península de Santa Elena expreso mi más sincero aprecio a estos ejemplares, cuya influencia positiva ha trascendido los confines de las aulas y han marcado mi crecimiento intelectual y personal resaltando la enseñanza y la calidad del saber.

Yuri Claribel Torres José

DEDICATORIA

Quiero dedicar este trabajo a mis padres. Ellos sembraron en mí el deseo de superación. Hoy soy lo que ellos anhelaron y me siento feliz de valorar su esfuerzo. Cada logro, cada obstáculo que supere, recordaré sus influencias y llevaré conmigo las lecciones que me han impartido, espero que mi camino refleje la dignidad y el éxito que han personificado en mí. A mis hermanos, por todo el apoyo y amor brindado. A ti Mario Torres, aunque ya no estas en este mundo, sé que este momento sería de mucha felicidad para ti como lo es para mí. A mi compañero de vida, que demostró ser la persona ideal en mi aventura académica y por supuesto a mis hijos; ellos fueron el motivo de haber llegado al final de esta carrera. A mis compañeros del aula y docentes, por vuestro apoyo que siempre será recordado con cariño y su amistad que guardaré por siempre en mi corazón.

Yuri Claribel Torres José

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del trabajo de titulación denominado: “Diseño e implementación de un sistema de control lineal cuadrático gaussiano en un sistema no lineal electromecánico denominado robot balancin”, presentado por la estudiante Yuri Claribel Torres José, de la carrera de Electrónica y Telecomunicaciones, me permito declarar que, tras haber proporcionado orientación, análisis y revisiones, se aprueba en todas sus partes.

Esta aprobación se informa para los fines pertinentes.

Ing. Carlos Alberto Saldaña Enderica, MSc.
Docente tutor

TRIBUNAL DE GRADO

Ing, Washington Torres Guin, Mgt.
DECANO DE FACULTAD

Ing. José Aquino Sánchez, Mgt.
DIRECTOR DE CARRERA

Ing. Carlos Saldaña Enderica, Mgt.
DOCENTE TUTOR

Ing. Luis Chuquimarca Jiménez, Mgt.
DOCENTE ESPECIALISTA

Ab. María Rivera González, Mgt.
SECRETARIA GENERAL

DECLARACIÓN DEL AUTOR

Yuri Claribel Torres José, egresada de la Universidad Estatal Península de Santa Elena, Facultad Sistemas y Telecomunicaciones, de la Carrea Electrónica y Telecomunicaciones, con libertad y en plena voluntad declaro que la responsabilidad del contenido del presente Trabajo de Titulación me corresponde exclusivamente y la propiedad intelectual de la misma le pertenece a la Universidad Estatal Península de Santa Elena.



Yuri Claribel Torres José

RESUMEN

El enfoque de la investigación es: diseño e implementación de un sistema de control Lineal Cuadrático Gaussiano (LQG) en un sistema no lineal electromecánico del "Robot balancín". El objetivo principal es lograr la estabilización del robot, mediante el uso de un control por retroalimentación de estados. Para ello, se emplea un estimador lineal cuadrático (LQE) junto a un regulador lineal cuadrático (LQR), ambos complementarios para el control LQG. Esto implica un estudio detallado de la teoría de control clásica y moderna, así como la adquisición de conocimientos previos de electrónica y las telecomunicaciones. Los pasos principales son: aplicar conocimientos de electrónica de programación de microcontroladores, establecer la comunicación entre el dispositivo y la computadora, desarrollar el modelo matemático en lazo abierto no lineal, linealizar alrededor del puntos de equilibrio por medio de variables de estados, aplicar el control al sistema linealizado y evaluar el modelo en Simulink e implementar el controlador en Arduino. El propósito final es lograr la estabilización del sistema mediante el controlador LQG, implementado en la tarjeta Arduino Uno.

PALABRAS CLAVES: Control, LQG, LQR, LQE, Filtro de Kalman, Estados, Linealización

ABSTRACT

The research focus is on the design and implementation of a Linear Quadratic Gaussian (LQG) control system for a nonlinear electromechanical system known as the "Balancing Robot." The primary objective is to achieve the stabilization of the robot through the use of state feedback control. To accomplish this, a Linear Quadratic Estimator (LQE) is employed in conjunction with a Linear Quadratic Regulator (LQR), both of which complement each other in the LQG control scheme. This involves a detailed study of both classical and modern control theory, as well as the acquisition of prerequisite knowledge in electronics and telecommunications. The key steps include applying knowledge of microcontroller programming and electronics, establishing communication between the device and the computer, developing the mathematical model for the nonlinear open-loop system, linearizing around equilibrium points using state variables, applying control to the linearized system, and evaluating the model in Simulink. Additionally, the control algorithm is implemented on an Arduino platform. The ultimate goal is to achieve system stabilization through the LQG controller, implemented on the Arduino Uno board.

KEYWORDS: Control, LQG, LQR, LQE, Kalman Filter, States, Linearization

INDICE GENERAL

AGRADECIMIENTO.....	i
DEDICATORIA	ii
APROBACIÓN DEL TUTOR.....	iii
TRIBUNAL DE GRADO	iv
DECLARACIÓN DEL AUTOR.....	v
RESUMEN.....	vi
ABSTRACT	vii
INDICE GENERAL.....	viii
INDICE DE FIGURAS	xiii
INDICE DE TABLAS	xvi
INDICE DE ANEXOS.....	1
INTRODUCCIÓN	2
CAPÍTULO I.....	3
1. FUNDAMENTACIÓN.....	3
1.1 Antecedentes.....	3
1.2 Descripción del Proyecto	5
1.3 Objetivos del Proyecto.....	6
1.3.1 Objetivo General:	6
1.3.2 Objetivos Específicos:.....	6
1.4 Justificación	6
1.5 Alcance del Proyecto	8
1.6 Metodología de la investigación	8
Investigación Descriptiva	9
Investigación Documental	9
Investigación Aplicada	9
1.7 Resultados esperados	9
CAPÍTULO II	11
2 MARCO REFERENCIAL.....	11
2.1 Marco Contextual.....	11
2.2 Marco Conceptual	13
2.2.1 Sistemas lineales invariantes en el tiempo (SLIT).....	13

2.2.2	Sistemas no lineales	13
2.2.3	Sistemas continuos o discretos	13
2.2.4	Perturbación.....	14
2.2.5	Robot Balancín	14
2.2.5.1	Péndulo Invertido	15
2.2.6	Dinámica del Sistema	17
2.2.6.1	Modelo matemático del sistema dinámico del péndulo invertido	17
2.2.6.2	Ecuaciones de Movimiento	17
2.2.6.3	Fuerzas Generalizadas	18
2.2.6.4	Euler Lagrange	19
2.2.6.5	Esquema del modelo matemático	19
2.2.7	Sistemas de control lazo abierto	20
2.2.8	Sistema de control lazo cerrado	21
2.2.8.1	Variables de estado.....	21
2.2.8.2	Ecuaciones de estados	22
2.2.9	Linealización de sistemas no lineales	23
2.2.9.1	Linealización extendida o por espacios de Estados.....	23
2.2.9.2	Puntos de equilibrio del sistema.....	25
2.2.10	Control automático	25
2.2.11	Teoría de control.....	25
2.2.11.1	Elementos de un sistema de control	26
□	Entrada.....	27
□	Salida	27
2.2.12	Componentes electrónicos y de comunicación.....	27
□	Giroscopio	27
□	Acelerómetro	27
□	Encoder.....	27
□	Actuadores.....	27
2.2.13	Comunicación Bluetooth.	28
2.2.13.1	Ventajas y desventajas de Bluetooth	28
2.2.14	Inestabilidad de un sistema.....	28
2.2.15	Controlabilidad y Observabilidad.....	29
2.2.15.1	Controlabilidad.....	29

2.2.15.2 Observabilidad.....	29
2.2.16 Observador de estados	30
2.2.17 Ganancia de control (K).....	30
2.2.18 Control de referencia	31
2.2.19 Control óptimo (LQR)	31
Regulador Óptimo Cuadrático lineal.....	31
2.2.20 Ruido blanco y ruido gaussiano.....	35
2.2.20.1 Densidad espectral de potencia	35
2.2.20.2 Ruido blanco.....	35
2.2.20.3 Ruido blanco gaussiano.....	36
2.2.21 Estimador Lineal Cuadrático (LQE) o Filtro de Kalman	36
2.2.21.1 Filtro de Kalman.....	37
2.2.22 Control Lineal Cuadrático Gaussiano (LQG).....	39
2.2.22.1 Ventajas y desventajas del control LQG	40
2.3 Marco teórico	41
CAPÍTULO III.....	43
3. DESARROLLO	43
3.1 Componentes de la propuesta.....	43
3.1.1 Proceso de la planta robótica	43
3.1.2 Componentes físicos (Hardware).....	43
3.1.2.1 Giroscopio- Acelerómetro MPU6050	44
3.1.2.2 Encoder double hall switch pulse output.....	44
3.1.2.3 Driver IC TBB6612FNG para motor DC	44
3.1.2.4 Motor DC GA37-520	45
3.1.2.5 Arduino Uno.....	45
3.1.3 Componentes lógicos (Software).....	46
3.1.3.1 Simulación Numérica	46
3.1.3.2 Arduino IDE	46
3.1.3.3 Soporte Arduino-Simulink	46
3.2 Diseño de la propuesta	46
3.2.1 Diseño Electrónico del Robot Balancín.....	47
3.2.2 Simulador Numérico.....	48
3.2.3 Aplicación del modelo matemático	51

3.2.2.1 Ecuaciones de movimiento.....	51
3.2.2.2 Lagrangiano.....	52
3.2.2.3 Mecánica de Lagrange.....	52
3.2.2.4 Fuerzas Generalizadas (F).....	54
3.2.2.5 Euler Lagrange.....	56
3.3 Linealización por espacios de estados.....	58
3.4 Entornos para el diseño del control.....	59
3.4.1 Evaluación de la estabilidad del sistema.....	62
3.4.2 Respuesta ante el impulso del sistema.....	62
3.4.3 Respuesta al escalón unitario.....	63
3.4.4 Análisis de controlabilidad.....	64
3.4.5 Análisis de observabilidad.....	64
3.5 Diseño del servo-controlador (LQG).....	65
3.5.1 Control LQR sin ruido ni perturbación.....	67
3.5.1.1 Respuesta del tiempo de estabilización (t_{ss}).....	72
3.5.1.2 Respuesta del sobre impulso máximo (M_p).....	73
3.5.1.3 Respuesta del ángulo de inclinación (θ).....	73
3.5.1.4 Respuesta de la señal de control $u(t)$	74
3.5.2 Control LQR con ruido y perturbación.....	75
3.5.2.1 Respuesta del tiempo de estabilización (t_{ss}).....	79
3.5.2.2 Respuesta del sobre impulso máximo (M_p).....	79
3.5.2.3 Respuesta del ángulo de inclinación (θ).....	80
3.5.2.4 Respuesta de la señal de control (u).....	80
3.5.3 Diseño del Filtro de Kalman.....	81
3.5.3.1 Filtro de Kalman Bucy.....	82
3.5.3.2 Filtro de Kalman.....	85
3.5.4 Servo-controlador LQG.....	89
3.5.5 Control LQG diseño en solo bloque.....	91
3.5.6 Diseño del servo controlador LQG por bloques separados.....	95
3.5.6.1 Respuesta del tiempo de estabilización (t_{ss}).....	98
3.5.6.2 Respuesta del sobre impulso máximo (M_p).....	99
3.5.6.3 Respuesta del ángulo de inclinación (θ).....	99
3.5.6.4 Respuesta de la señal de control.....	100

3.7 Diseño del controlador LQG en Arduino IDE	101
3.9 Comunicación Arduino IDE - Robot balancín.....	102
3.8 Comunicación del módulo Bluetooth - Robot balancín.....	103
CAPÍTULO IV.....	107
4. EVALUACIONES Y RESULTADOS	107
4.1. Evaluaciones	107
4.1.1 Comparativa del control LQR vs LQG.....	107
4.1.1.1 Desempeño del sistema de control LQG	107
4.1.1.2 Desempeño del sistema de control LQR	108
4.1.2 Evaluaciones del servo-control LQG.....	109
4.1.2.1 Análisis de Evaluación 1: $x = 0$ y $\theta = 10^\circ$	109
4.1.2.2 Análisis de Evaluación 2: $x = 3$ y $\theta = 30^\circ$	111
4.1.2.3 Análisis de Evaluación 3: $x = 0$ y $\theta = 0^\circ$	114
4.1.2.4 Análisis de Evaluación 4: $x = 10$ y $\theta = 15^\circ$	117
Conclusiones	120
Recomendaciones.....	122
Bibliografías.....	123
Anexos.....	130

INDICE DE FIGURAS

Figura 1 Descripción del proyecto.....	5
Figura 2 Sistema Continuo [45].....	14
Figura 3 Sistema Discreto [46].	14
Figura 4 Robot Balancín.	15
Figura 5 Modelos de Péndulo Invertido [43].....	15
Figura 6 Esquema del Modelo Matemático [49].	20
Figura 7 Diagrama Lazo Abierto [43].	21
Figura 8 Diagrama Lazo Cerrado [51].....	21
Figura 9 Representación de una variable en Espacio de Estados [45].....	22
Figura 10 Diagrama de bloques en Espacios de Estados [53].	24
Figura 11 Observador de Estados [58].....	30
Figura 12 Sistema Regulador Óptimo [58].	32
Figura 13 (Pds) Ruido Térmico y Ruido Blanco [61].....	36
Figura 14 Filtro de Kalman [33].	38
Figura 15 Diagrama del Control Lqg [40].	40
Figura 16 Proceso del trabajo de la planta [9].	43
Figura 17 Conexión de Pines [66].	44
Figura 18 Encoder Jga25-370 [66].	44
Figura 19 Driver Ic Tbb6612fng [66].	45
Figura 20 Motor Dc Ga37-520 [66].....	45
Figura 21 Diseño Electrónico del Robot Balancín.	47
Figura 22 Bloque Matlab© de System Identification [67].	48
Figura 23 A) Señal de Salida B) Señal de Entrada.	48
Figura 24 Mejores Estimaciones del Sistema.	49
Figura 25 Esquema del Péndulo para el modelo dinámico [45].....	51
Figura 26 Diagrama del Motor Dc.....	53
Figura 27 Respuesta al Impulso en Tiempo Continuo.....	62
Figura 28 Respuesta al Impulso en Tiempo Discreto.	63
Figura 29 Respuesta al Escalón Unitario.	63
Figura 30 Control Lqr sin Perturbación ni Ruido.	70
Figura 31 Señal de Referencia.	71
Figura 32 Estados del Sistema Regulado con Lqr	71
Figura 33 Salidas Lqr.....	72
Figura 34 Tiempo de Estabilización Lqr.	72
Figura 35 Máximo Impulso Lqr.....	73
Figura 36 Ángulo de Inclinación en el Control Lqr.....	73
Figura 37 Regulación del Ángulo en Lqr	74
Figura 38 Valor Inicial en Señal de Control en Lqr.....	74
Figura 39 Señal de Control en Lqr.....	75
Figura 40 Control Lqr Con Ruidos y Perturbaciones.	75
Figura 41 Ruido de Media del Sistema Controlador Lqr.....	76
Figura 42 Ruido de Proceso del Sistema del Controlador Lqr	77

Figura 43 Perturbación Externa en el Control Lqr.....	77
Figura 44 Salidas Lqr con Perturbación y Ruido.....	78
Figura 45 Contaminación del Ángulo del Sistema en el Control Lqr.....	78
Figura 46 Señal de Control Contaminada en Lqr.	78
Figura 47 Tiempo de Estabilización Lqr Perturbado.....	79
Figura 48 Máximo Impulso Lqr Perturbado.	80
Figura 49 Ángulo de Inclinación en el Control Lqr Perturbado.	80
Figura 50 Valor Inicial en Señal de Control en Lqr Perturbado.....	81
Figura 51 Señal de Control en Lqr Perturbado.	81
Figura 52 Bloque del Filtro de Kalman.	85
Figura 53 Kalman Estimador [40].	85
Figura 54 Configuración de Bloque Kalman Filter [67].....	86
Figura 55 Configuración de Opciones Kalman Filter [67].	87
Figura 56 Sistema de Control y Filtro De Kalman.	87
Figura 57 Salidas Estimadas Óptimas.....	88
Figura 58 Salidas Óptimas Estimadas.....	88
Figura 59 Salidas Reales y Salidas Filtradas.	89
Figura 60 Servo Controlador Lqg [68].	90
Figura 61 Controlador Lqg en un solo bloque.	92
Figura 62 Ruido de Media en el Sistema Lqg.....	92
Figura 63 Ruido de Proceso en el Sistema Lqg.	93
Figura 64 Referencia- Salidas Lqg.	93
Figura 65 Ángulo del Servosistema Lqg.	94
Figura 66 Señal de Control del Servo Lqg.....	94
Figura 67 Diseño del Servosistema Lqg por separado.....	95
Figura 68 Ruido de Media en el Servosistema Lqg, por bloques separados.	95
Figura 69 Ruido de Proceso en el Servosistema Lqg, por bloques separados.....	96
Figura 70 Perturbación al Sistema por bloques separados.....	96
Figura 71 Estados Óptimos Estimados del Servo Sistema Lqg diseñado por separado...	97
Figura 72 Salidas Reales y Salidas Estimadas Óptimas.	97
Figura 73 Salidas Lqg en bloques separados.	98
Figura 74 Tiempo de Estabilización (t_{ss}) del Control Lqg.....	98
Figura 75 Máximo Sobre Impulso (M_p) en Lqg.	99
Figura 76 Ángulo de Inclinación en el Control Lqg.....	99
Figura 77 Regulación del Ángulo en Lqg.....	100
Figura 78 Señal de Control (u) en el Servo Controlador Lqg Diseño de bloques separados.	100
Figura 79 Señal de Control con sus valores máximo y mínimo para el Control Lqg.....	101
Figura 80 Intervalos de la Señal de Control para el Control Lqg.	101
Figura 81 Carga del Código al microcontrolador.	103
Figura 82 Conexión Inalámbrica de la planta.	104
Figura 83 Ángulo del Robot en Tiempo Real.....	104
Figura 84 Aceleración Angular del Robot en Tiempo Real.	105
Figura 85 Comunicación Bluetooth.....	106

Figura 86 Tiempo de Estabilización Lqg para Evaluación 1.....	109
Figura 87 Sobre Impulso Máximo (Mp) en controlador Lqg para Evaluación 1.....	110
Figura 88 Ángulo de Inclinación en el controlador Lqg en la Evaluación 1.....	110
Figura 89 Error de Estado Estacionario (ess) en controlador Lqg para Evaluación 1...	111
Figura 90 Señal de Control en Controlador Lqg ara Evaluación 1.....	111
Figura 91 Tiempo de Estabilización (tss) en Controlador Lqg para Evaluación 2.	112
Figura 92 Sobre Impulso Máximo (Mp) en Controlador Lqg para Evaluación 2.....	112
Figura 93 Ángulo de Inclinación en el controlador Lqg en la Evaluación 2.	113
Figura 94 Error de Estado Estacionario (ess) en ontrolador Lqg para Evaluación 2....	113
Figura 95 Señal de Control en Controlador Lqg para Evaluación 2.....	114
Figura 96 Tiempo de Estabilización (tss) en controlador Lqg para Evaluación 3.....	114
Figura 97 Sobre Impulso Máximo (Mp) en controlador Lqg para Evaluación 3.....	115
Figura 98 Ángulo de Inclinación en Controlador Lqg para Evaluación 3.....	115
Figura 99 Error de Estado Estacionario (ess) en Controlador Lqg para Evaluación 3. .	116
Figura 100 Señal de Control Para El Controlador Lqg para Evaluación 3.....	116
Figura 101 Tiempo de Estabilización (tss) en controlador Lqg para Evaluación 4.....	117
Figura 102 Sobre Impulso Máximo (Mp) en controlador Lqg para Evaluación 4.....	117
Figura 103 Error de Estado Estacionario (ess) en ontrolador Lqg para Evaluación 4...	118
Figura 104 Ángulo de Inclinación en el controlador Lqg para Evaluación 4.....	118
Figura 105 Señal de Control en Controlador Lqg para Evaluación 4.....	119
Figura 106 Diagrama del Puente H [66].....	165
Figura 107 System Identification [67].	165
Figura 108 Identificación de la planta.	166
Figura 109 Robot Balancín en su acción equilibrista.	166

INDICE DE TABLAS

Tabla 1 Variables del sistema dinámico [44].....	16
Tabla 2 Identificación de la planta.....	49
Tabla 3 Simbologías de variables de estado	58
Tabla 4 Especificaciones del control lqg.	108
Tabla 5 Especificaciones del control lqr.	108
Tabla 6 Especificaciones del ruido gaussiano blanco y perturbaciones.	108
Tabla 7 Evaluaciones y comparaciones del control lqg.....	109
Tabla 8 Datos técnicos del motor dc ga37-520 [66].	160
Tabla 9 Datos técnicos del arduino uno [9]	160
Tabla 10 Datos técnicos del sensor mpu-6050 [66].	161
Tabla 11 Datos técnicos de modulo bluetooth hc-05 [58].	162
Tabla 12 Puente h del robot balancín [66].	163

INDICE DE ANEXOS

Anexo 1. Desarrollo de las ecuaciones de movimiento	130
Anexo 2. Desarrollo de Lagrangiano	131
Anexo 3. Desarrollo de Euler Lagrange	132
Anexo 4. Cambio de Ecuaciones dinámicas Eecuaciones en variables de estados y puntos de equilibrio.....	134
Anexo 5. Desarrollo de Jacobianos.....	134
Anexo 6. Código Matlab© para la comunicación Bluetooth.....	137
Anexo 7. Análisis para la Estabilidad del sistema	142
Anexo 8. Análisis para la Controlabilidad.....	143
Anexo 9. Análisis para la Observabilidad.....	144
Anexo 10. Diseño del código para el servo-controlador Lqg en Matlab©	144
Anexo 11. Diseño del código para el controlador Lqg en Arduino	147
Anexo 12. Datos técnicos del Motor DC Ga37-520	160
Anexo 13. Datos técnicos del Arduino Uno	160
Anexo 14. Datos técnicos del Sensor Mpu-6050.....	161
Anexo 15. Datos técnicos de modulo Bluetooth Hc-05.....	162
Anexo 16. Datos técnicos del puente H Tb6612fng	162
Anexo 17. Diagrama del circuito itegrado puente H	165
Anexo 18. Herramienta System Identification.....	165
Anexo 19. Acción controlada del robot balancín.....	166

INTRODUCCIÓN

La mayoría de los sistemas físicos presentan complejidad respecto a su no linealidad, también sufren perturbaciones externas, punto muy esencial para estudiar la dinámica de los sistemas lineales y no lineales mediante modelos matemáticos en tiempo real.

Los comportamientos de los sistemas dependen mucho de sus condiciones iniciales, pero si se aplica un control es más eficiente y necesario ante perturbaciones externas. Este controlador óptimo lineal cuadrático gaussiano (LQG) es aplicado en un robot Balancín ELEGOO TUMBLER, que como objetivo principal se equilibra y rechaza ruidos blancos aditivos, procesos aleatorios, alteraciones externas, entre otros.

El sistema no lineal electromecánico se linealiza en varios puntos de equilibrio mediante la linealización extendida o variables de estados que se enfoca en el Lagrangiano del sistema, disminuyendo la complejidad del análisis de la planta.

Los controles modernos son algoritmos que difieren de los controles clásico lineal, debido a su precisión y menor tiempo de estabilización, muy eficaz para desarrollos tecnológicos industriales, representando un desafío a la naturaleza no lineal y la complejidad de control y estabilización.

El presente trabajo consta de IV capítulos: capítulo I presenta antecedentes, descripción del proyecto, objetivos, justificación, alcance y metodologías, en el capítulo II comprende: marco conceptual, contextual y marco teórico necesario para el tema investigativo, capítulo III: contiene el desarrollo del modelo matemático, linealización por variables de estados y diseño del control LQG y en el capítulo IV consta de gráficas del comportamiento lineal y no lineal, controlado mediante el software MATLAB©. También se encuentra la conclusión, recomendaciones fuentes bibliográficas y anexos.

CAPÍTULO I

1. FUNDAMENTACIÓN

1.1 Antecedentes

A lo largo del tiempo la teoría del control ha desarrollado grandes propuestas; como las estrategias deterministas y heurísticas; un claro ejemplo es el control PID y el control por retroalimentación de estado, y las heurísticas las redes neuronales y lógica difusa. La eficiencia del control lineal parte de la base elemental de una teoría de control robusto y el exitoso progreso en la industria; sin embargo, la Facultad de Ingenierías, Universidad de San Buenaventura, sede Medellín (USBMed) ha desarrollado comparaciones que demuestran que el control no lineal es la linealización de entrada-salida, siendo base de varias propuestas en control y perturbaciones [1] [2].

El trabajo de investigación realizado en la Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica menciona que dentro de las empresas industriales tecnológicas el diseño de un sistema de control óptimo es considerablemente importante. Por lo general, se piensa que el tipo de automatización que se emplea no es muy eficiente debido a la tendencia de aplicar sistemas de control clásicos, por lo cual, surge la necesidad del uso de técnicas modernas en sistemas de control [3].

En una publicación de la revista Visión Electrónica, Universidad Francisco José de Celdas de Bogotá señala que la tecnología ha tenido un gran impacto en la sociedad, transformando rápidamente la forma en que se realizan diversas tareas. Por ejemplo, el robot balancín, basado en el principio del péndulo invertido, puede desplazarse sobre dos ruedas y realizar las mismas funciones que un robot humanoide [4].

Algunas aplicaciones son: el control de estabilidad de grúas y vehículos eléctricos. Por ejemplo, la compañía Segway indica que ha diseñado vehículos donde el conductor introduce como referencia la velocidad y dirección, mientras el vehículo debe mantener el equilibrio [4].

Por otro lado, Handle es un robot de alta movilidad diseñado por Boston Dynamics para manipulación y transporte de objetos. Es capaz de moverse de manera ágil y rápida por superficies irregulares y terrenos difíciles, gracias a sus ruedas que pueden girar 360 grados y sus piernas con amortiguación que le permiten saltar obstáculos. También cuenta con brazos robóticos que pueden manipular objetos con gran precisión y velocidad.

Handle mejora la eficiencia y la productividad en una amplia gama de industrias, desde la logística y el transporte hasta la manufactura y la construcción [5].

Varios estudios sobre el péndulo invertido dan a conocer que, el problema de control lineal cuadrático gaussiano LQG es una de las complicaciones de control óptimo más esenciales [6]. Aquí se define que son sistemas lineales inducidos por ruido gaussiano blanco aditivo y el problema es definir la ley de retroalimentación de salida óptima en el sentido de que minimice el valor esperado del criterio de costo cuadrático [7]. Además, indican que las mediciones de salida están perturbadas por el ruido gaussiano y, de manera análoga, se cree que el estado inicial es un vector gaussiano aleatorio. Bajo estos supuestos, el esquema de control óptimo se puede inferir mediante el argumento de finalización al cuadrado. Esta ley de control es única y que combina un filtro de Kalman (estimador de estado cuadrático lineal (LQE)) con un regulador cuadrático lineal (LQR) [5] [7] [8].

Según Dermendjian [10], indica que en los últimos años la flexibilidad proporcionada por criterio cuadrático de LQG es notable y amplia donde es posible elegir parámetros libres de no solo minimizar errores por mínimos cuadrados, sino también modelar bucles, compensando la retroalimentación, lograr la sensibilidad y funciones deseables de transferencia en los puntos críticos [11].

En la Universidad de Harvard, estudios revelan que en el 2019 ya se aborda una clase de problemas Gaussianos Cuadráticos Lineales, parcialmente observados con dinámica desconocida, estableciendo una complejidad de muestra de extremo a extremo vinculada al aprendizaje de un controlador LQG robusto para plantas estables de bucle abierto [12]. Por tanto, el caso de los péndulos invertidos se considera uno de los sistemas no lineales más utilizados para la experimentación de diferentes estrategias de control [13].

Cabe resaltar que el mecanismo ha sido ampliamente estudiado por la industria aeroespacial, dado que representa uno de los procedimientos clásicos más conocidos para estabilizar sistemas [13]

Por lo antes descrito, y siendo este un tema de investigación es necesario modelar algoritmos de retroalimentación de estados, que promuevan técnicas convencionales en el área de control automático.

1.2 Descripción del Proyecto

El presente trabajo de investigación tiene como propósito el diseño y la implementación de un sistema para la estabilización de un robot balancín, mediante el uso de un control por retroalimentación de estados, donde se procede a modelar un estimador lineal cuadrático LQE con un regulador lineal cuadrático LQR, ambos perteneciente a un sistema de control Lineal Cuadrático Gaussiano LQG, con el propósito de equilibrar y rechazar las perturbaciones en el sistema de péndulo invertido, revisando ampliamente la teoría de control clásica y moderna. Para la ejecución de este proyecto proceden los siguientes pasos como indica la Figura (1).

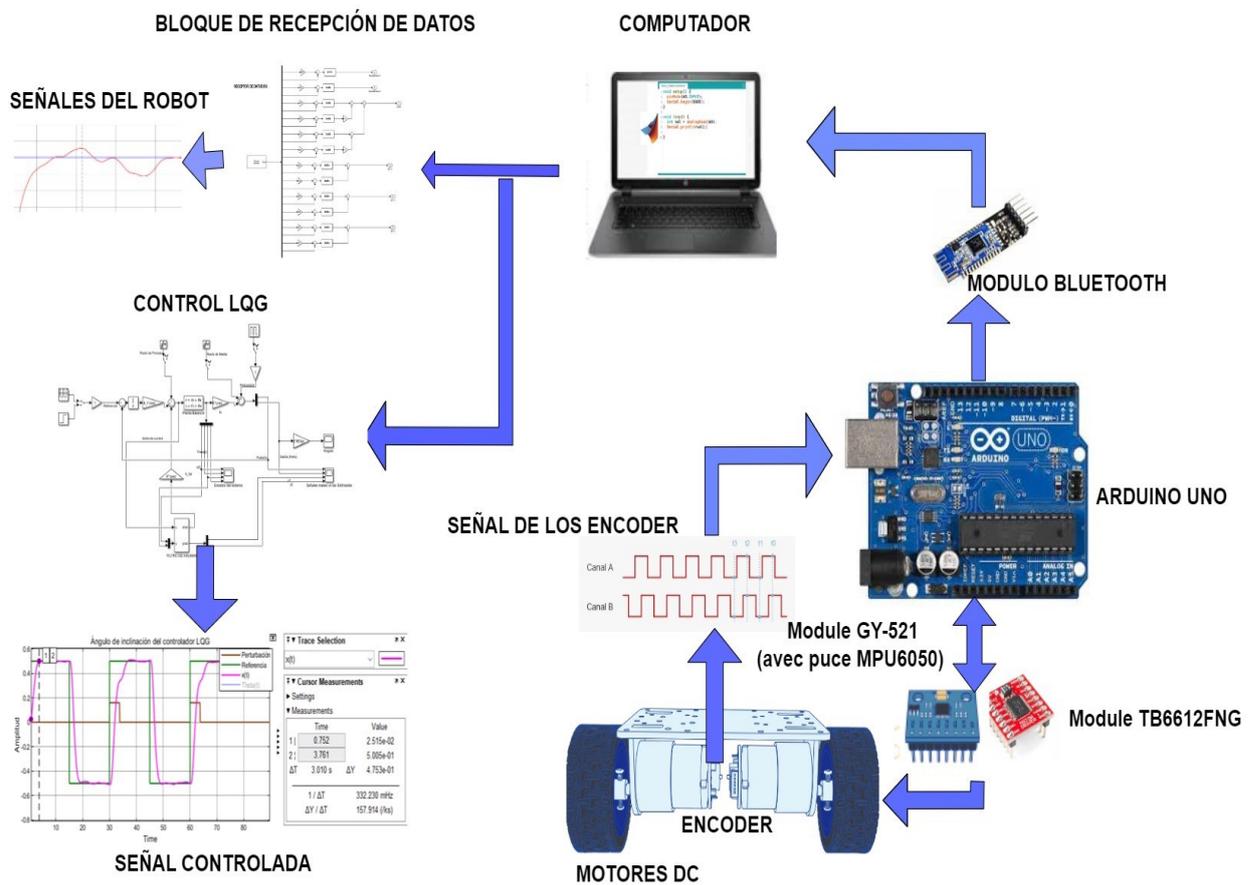


Figura 1 Descripción del proyecto.
Realizado por: Yuri Torres, 2024.

1.3 Objetivos del Proyecto

1.3.1 Objetivo General:

Modelar e implementar un control retroalimentado mediante un sistema de control Lineal Cuadrático Gaussiano LQG, aplicado a la estabilización de un sistema electrónico robótico balanceado.

1.3.2 Objetivos Específicos:

- Investigar e identificar los componentes del esquema electrónico que conforman el robot auto equilibrista para el funcionamiento correcto del hardware y software del sistema.
- Analizar el diagrama de conexiones de los elementos electrónicos en un software de simulación para tener una mejor comprensión de sus principios y funcionamiento de los componentes de manera efectiva y precisa.
- Modelar matemáticamente la planta para realizar el debido control de las variables del sistema, aplicando las teorías de modelado correspondiente en el sistema electromecánico.
- Diseñar el control lineal cuadrático gaussiano retroalimentado usando un simulador numérico para estimar y regular perturbaciones en el sistema no lineal dentro de una simulación tomando en consideración los parámetros que se obtuvieron en el modelado.
- Embeber el algoritmo de control en la placa del microcontrolador para evidenciar el balanceo que realiza el control lineal cuadrático gaussiano.
- Establecer una comunicación mediante el bluetooth del simulador numérico y la planta para recopilar datos generados por los sensores que obtienen información de los actuadores, permitiendo una inspección visual de resultados con respecto a la planta electrónica auto equilibrista.

1.4 Justificación

Los procesos industriales y el contar con personal capacitado en el área de control, se considera una necesidad para la optimización de los sistemas de producción. Las industrias deben buscar la innovación a través de la implementación de nuevos sistemas de control para mejorar la calidad del producto final. Gran parte de los países desarrollados, han

invertido en la robótica industrial como una alianza cooperativa entre el ser humano y la máquina, para asistir y mejorar los procesos de fabricación y atención técnica [14].

Además, como estudiante de la carrera Electrónica y Telecomunicaciones estudio los componentes electrónicos que también se utilizan para procesar, analizar la información de retroalimentación y para generar las señales de control que se envían al motor del robot para ajustar su posición y velocidad. Por ejemplo, los microcontroladores y los procesadores digitales de señales se utilizan para realizar los cálculos necesarios para implementar la retroalimentación LQG y generar las señales de control adecuadas.

La rama de la electrónica es fundamental para la implementación de la retroalimentación LQG en un robot balancín, ya que los componentes electrónicos se utilizan para medir la posición y la velocidad del robot, procesar la información de retroalimentación y generar las señales de control son necesarias para mantener el equilibrio del robot.

De esta manera el tema de tesis busca generar un camino para que futuros estudiantes tengan la oportunidad de mejorar sus capacidades de exploración en el área, alineando este proyecto con los objetivos de la norma ecuatoriana: “ El propósito del sistema nacional de ciencia, tecnología, innovación y saberes ancestrales, en concordancia con el respeto hacia el medio ambiente, la naturaleza, la vida, las culturas y la soberanía, consiste en: Producir, ajustar y divulgar conocimientos científico tecnológicos.”, mencionado en los Artículos 350, 355 y 386 de la Constitución de la República Ecuatoriana [15], [16].

Por lo antes descrito, existe el compromiso con el país, bajo los aportes de estudios relacionados a la tecnología donde fluyen innovaciones a nivel nacional e internacional donde ya es familiar el concepto del robot en sistemas industriales. Basado en estos precedentes se motiva el desarrollo de la investigación del caso, además el control y modelado de sistemas no lineales es un tema de alto interés científico en la actualidad, y una necesidad de continuar indagando e investigando, y luego desarrollar temas de investigación locales, nacionales o internacionales [17].

Hay diversos campos de investigación enfocados en distintos tipos de control, sin embargo, son pocos los que han sido implementados en un sistema real. Por lo tanto, se propone utilizar un control LQG óptimo que considere las perturbaciones externas, con el objetivo de mejorar la estabilidad a través de la aplicación de dicho control.

En base a estos indicadores queda demostrada la importancia del enfoque investigativo que conlleva un gran aporte al desarrollo tecnológico y en un futuro la innovación de nuevas herramientas. A nivel provincial se conoce que en el Repositorio de la Universidad Península de Santa Elena se encuentran estudios orientados a la aplicación de equipos industriales en complemento de controladores clásicos como PID, sin embargo, el análisis e implementación de estos sistemas puede considerarse superficial, por esta razón se realiza y se justifica la presente propuesta [18]. A pesar de la robustez del controlador, no funciona de manera eficiente en caso de un control óptimo; según el estudio de Kouakou. Por tanto, es un justificativo para este trabajo de investigación [17], [19], [20].

1.5 Alcance del Proyecto

Se realiza el estudio de la aplicación de un control LQG sobre un robot balancín, mediante un modelado que explica el comportamiento dinámico del sistema con la finalidad de mantener el equilibrio. En base a teorías del control moderno analizamos las perturbaciones, desafíos y dificultades que afectan la estabilidad del sistema cuando se implementa el controlador en un entorno real.

Este proyecto contempla los siguientes puntos:

- Desarrollo del modelo matemático por medio de técnicas convencionales en el área de control automático.
- Simular el sistema con la finalidad de realizar evaluaciones y comparaciones del control sobre el modelo matemático mediante un software numérico.
- Con la simulación precisa se procede a cargar el controlador a la planta propuesta, para las respectivas evaluaciones de experimentación real. Con esta herramienta se ajusta las señales del sistema mediante evaluaciones y resultados.

En general; la retroalimentación LQG, es una técnica de control muy efectiva para sistemas dinámicos complejos, como los robots balancín; por lo tanto, se puede proporcionar una respuesta rápida y precisa a las perturbaciones del sistema. Sin embargo, la eficacia de la retroalimentación LQG dependerá de la precisión y la fiabilidad del modelo del sistema, la calidad y la precisión de los sensores y la habilidad del controlador para ajustar los parámetros de control en tiempo real.

1.6 Metodología de la investigación

Para el desarrollar la propuesta se aplican las siguientes estrategias de investigación:

Investigación Descriptiva

Este proyecto se enfoca en describir los sistemas de control mediante retroalimentación de estados, abordando variables de estados, controlabilidad, estabilidad y el controlador lineal cuadrático gaussiano LQG. Se aplica este enfoque al modelamiento de un algoritmo de control para un robot balancín, explorando cómo estos elementos contribuyen a lograr un control preciso y estable. El énfasis recae en entender cómo la retroalimentación de estados y el LQG optimizan el rendimiento del sistema y la estimación de estados para proporcionar una base sólida para un control efectivo del robot balancín y explorar aplicaciones en otros contextos similares [21].

Investigación Documental

Se realiza un análisis exhaustivo de fuentes bibliográficas relevantes para el control aplicado al péndulo invertido, donde se exploran conceptos teóricos esenciales, como el modelo matemático del sistema y los fundamentos del control LQG; destacando el estado actual de la investigación, comparando enfoques de control alternativos y resaltando casos de estudio aplicados al péndulo invertido. Además, se examinan desafíos y limitaciones, proporcionando una visión integral de las implicaciones prácticas y teóricas. Esta revisión bibliográfica fundamenta la comprensión y desarrollo del tema [21].

Investigación Aplicada

Este proyecto busca analizar el comportamiento y aplicar el conocimiento del sistema de control en un campo científico o industrial; en este caso, agregándolo a la planta representada en la variable estado [21].

Resultados.

1.7 Resultados esperados

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

- Extracción de la información respectiva del robot ELEGOO Tumbller, incluyendo datos técnicos, características y detalles sobre los componentes electrónicos para entender capacidades y limitaciones del robot.
- Resolución del diagrama de conexiones del sistema electrónico del "robot balancín".
- Creación del modelo matemático del sistema dinámico no lineal.

- Identificación de la planta robótica y selección de la mejor estimación.
- Linealización del modelo matemático a través de variables de estados y técnicas jacobianas alrededor del punto de equilibrio.
- Desarrollo del controlador lineal cuadrático gaussiano LQG, y adaptación al sistema linealizado.
- Pruebas y evaluaciones que identifiquen el funcionamiento del sistema equilibrado, mediante software de alto nivel.
- Análisis del mejor rendimiento del control LQG ante el control LQR en base a las respectivas simulaciones.
- Se espera que el algoritmo de control LQG en Arduino IDE, sea embebido en el microcontrolador para evidenciar el eficiente equilibrio del robot.
- Comunicación mediante el módulo Bluetooth y el robot balancín para visualizar la aceleración angular en tiempo real.

CAPÍTULO II

2 MARCO REFERENCIAL

En este apartado se describen los avances tecnológicos referentes a la teoría de control, aplicados en diferentes regiones del mundo, lo que permite ver el campo de aplicación para este trabajo.

2.1 Marco Contextual

En la actualidad el desarrollo tecnológico avanza en varios campos de conocimientos en la electrónica y las telecomunicaciones, por ejemplo, evoBOT un robot auto equilibrado y autónomo muestra la importancia de cómo aplicar el estudio de la propuesta, fue desarrollado por el Instituto Fraunhofer de Flujo de Materiales y Logística IML en Dortmund, Alemania. Este instituto es uno de los principales centros de investigación aplicada en Europa y desarrolla soluciones innovadoras para la logística y la gestión de la cadena de suministro. Cabe indicar que EvoBOT es un robot móvil balanceado que se utiliza para la automatización de procesos logísticos, como la entrega de piezas en líneas de producción o la gestión de inventarios en almacenes [22]. Con este paso, muy importante en un futuro, se espera que los robots se utilicen cada vez más dentro de las universidades como laboratorios de Electrónica, Automatización, en ambientes hospitalario, oficinas y lugares donde la variedad de sistemas utilizados es amplia y la interacción con las personas es muy esencial como lo hace este robot.

En el área industrial tenemos una optimización de la cadena de producción, donde las aplicaciones de control son una necesidad en el mundo y las aplicaciones automatizadas representan un gran apoyo en el área por ejemplo, el control del equilibrio de Robot Camarero que ayuda en la transportación de objetos llevando una bandeja sujeta a la muñeca con diseños basados en el Robot Humanoide Teo [23], esta importante aplicación tiene relevancia directa con el control LQG aplicado en el péndulo invertido cumpliendo la capacidad de moverse de manera autónoma y equilibrada en superficies irregulares donde se requiere una gran movilidad y estabilidad, como la inspección de instalaciones, el mapeo de terrenos o la entrega de paquetes. Ideales para trabajar en espacios reducidos o para maniobrar en entornos con obstáculos [24].

En América Latina, alrededor del 60% de las actividades en la industria son susceptibles de ser automatizadas. Asimismo, en Ecuador, se estima que aproximadamente

el 49% de los procesos de manufactura pueden ser reemplazados por tecnologías avanzadas [25] [26]. Si embargo, este tipo de tecnología es uno de los más interesantes dentro del mundo actual de la electrónica y programación automática, uno de los hechos más relevantes en la compañía Segway tenemos el S-Pod, que fue presentado por primera vez en el CES 2020 [27]. Este vehículo autónomo auto balanceado fue diseñado para ser utilizado en entornos cerrados como aeropuertos, parques temáticos y campus universitarios, y cuenta con un sistema de control de gestos que permite a los usuarios conducir el vehículo sin necesidad de un volante o joystick [28] [29] [30] con base de estos ejemplos el control LQG puede ser aplicado en ambiente similares o en áreas de investigación como los laboratorios de la Universidad Estatal Península de Santa Elena.

En varias universidades e Institutos antes mencionados, se estudia las perturbaciones que existen en un sistema, donde se ha comprobado que el controlador LQG, permite mejorar el rendimiento y la estabilidad de los sistemas dinámicos mediante la aplicación del control por ejemplo la implementación activa de este control en los laboratorios universitarios de electrónica o automatización, no solo contribuirá al avance académico sino que también potenciará la eficiencia y productividad en diversos sectores, mejorando así la calidad de vida de la comunidad estudiantil y de la población en general [31] [32] [33].

Algunas empresa ecuatorianas como Simax han implementado robots colaborativos que trabajan con los operadores humanos y se encargan de tareas como el ensamblaje y la inspección de los productos [34], Mabe ha implementado robots de soldadura y ensamblaje en su planta de producción de Quito, estos son de la marca Yaskawa y se utilizan en el proceso de fabricación de electrodomésticos [35]. Otra implementación se encuentra en la empresa de Cervecería Nacional del Guayas, donde se ha empleado robots marca Krones, estos se encargan del proceso de envasado, etiquetado y actualmente lavado de botellas [36] [37], dando como resultado, la automatización de diversas empresas mejorando sus modelos de negocio y obteniendo beneficios significativos en términos de eficiencia y precisión [38] [39]. A medida que la tecnología avanza, es posible que empresas ecuatorianas, consideren la implementación del robot de péndulo invertido y puedan existir mejoras en sus procesos de trabajo.

En la Universidad Estatal Península de Santa Elena se empieza a contribuir con la investigación en teoría de control y modernizar conocimientos en un área competitiva,

donde esto será un peldaño para los futuros investigadores. De tal manera el tema de los controladores es una necesidad significativa en la eficiencia y rendimiento de los sistemas de control donde permitan una mayor precisión en la medición y control de las variables de los procesos, lo que a su vez puede redundar en un aumento de la calidad del producto, una reducción de costos y una mejora en la seguridad de las operaciones. Además, la capacidad de controlar sistemas complejos en tiempo real hace que los controladores LQG sean ideales para aplicaciones de automatización y control de procesos críticos, lo que es una ventaja competitiva para las empresas en el entorno industrial actual [10].

2.2 Marco Conceptual

En este marco de conceptos se detallan los principios básicos y fundamentales de la teoría de control.

2.2.1 Sistemas lineales invariantes en el tiempo (SLIT)

Sistema se refiere a componentes que trabajan con un solo propósito, pueden ser físicos o abstractos que mantienen objetivos determinados. Entonces la palabra sistema comprende un amplio medio de sistemas físicos, biológicos, económicos entre otros [40].

Los sistemas lineales son procedimientos que relacionan las entradas y salidas siguiendo el principio de superposición y homogeneidad. Es decir, la respuesta se calcula de manera secuencial, abordando cada entrada y sumando los resultado [40]. Además, se considera la solución a la complejidad de los sistemas no lineales.

SLIT, son los que hacen que su comportamiento y propiedades no cambien en el tiempo. Es decir, si aplicas una entrada, se obtiene una salida y luego aplicas la misma entrada y tendré la misma salida. por ejemplo, sistemas de control en estado estacionario, circuitos electrónicos, entre otros [40].

2.2.2 Sistemas no lineales

Los sistemas no lineales no cumplen el principio de superposición ni homogeneidad, es decir no se puede describir una linealidad entre las entradas y salidas porque pequeños cambios en las entradas no producen cambios proporcionados en la salida provocando complicidad en el modelo matemático [40].

2.2.3 Sistemas continuos o discretos

Sistema continuo es aquel cuando las entradas y salidas de un sistema varían continuamente en el tiempo. Son fundamentales en el análisis de sistemas mecánicos en

movimiento, péndulo oscilatorio, entre otros procesos matemáticos [2], [41] . Además, las señales continuas de entradas son transformadas en señales continuas de salidas como indica la Figura (2) [40].



Figura 2 Sistema continuo [45].

Un sistema discreto hace referencia a variaciones en las señales de entrada y salida en un intervalo de tiempo Figura (3). Por ejemplo, el control de un inventario, sistemas de telecomunicaciones, etc. Es importante señalar que la señal de entrada en discreto transforma la señal de salida en tiempo discreto [2] [41].

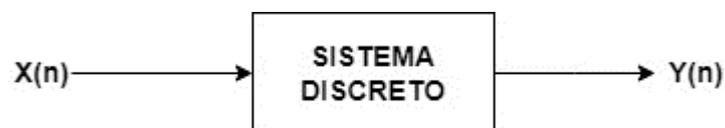


Figura 3 Sistema discreto [46].

Se dispone diversas técnicas en el proceso de un modelo matemático y se consideran técnicas externas, donde interviene la observación y análisis de las entradas y salidas del sistema y técnicas de representación interna que se enfocan en el estudio de los procesos y variables internas del sistema. Ambos enfoques son importantes para construir un modelo matemático [40] [41].

2.2.4 Perturbación

Es una señal que afecta, cambia o interfiere en la salida de un sistema. Se presentan de forma interna o externa lo que significa que están dentro o fuera del sistema, suelen ser ruidos o cualquier factor que desvíe el funcionamiento del sistema [41].

2.2.5 Robot Balancín

Un robot de balanceo es un dispositivo diseñado para mantenerse equilibrado mientras realiza tareas. Estos robots generalmente incluyen una combinación de sensores, controladores, motores y componentes mecánicos para ayudar a mantener el equilibrio su equilibrio. Pueden tener una variedad de aplicaciones, desde ayudar a los humanos a realizar tareas que requieran balancearse como llevar cargas pesadas, hasta aplicaciones de entretenimiento como los robots de patinaje. Donde el péndulo invertido es una técnica de

estabilidad que utiliza un sistema de control para mantener el equilibrio. Los modelos matemáticos de estos sistemas se caracterizan por ser de tipo no lineal. En este caso usamos el tipo de Robot Auto equilibrante Elegoo Tumbler como se muestra en la Figura (3) [42].

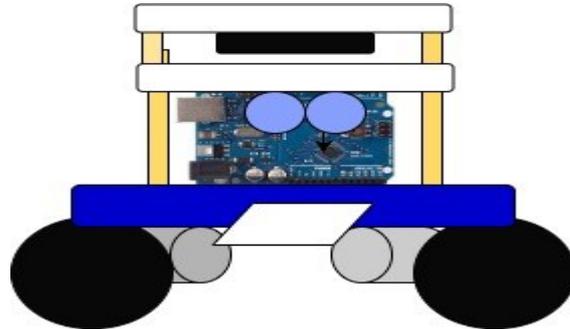


Figura 4 Robot Balancín.

Realizado por: Yuri Torres, 2024

2.2.5.1 Péndulo Invertido

Un péndulo invertido es un dispositivo mecánico que consiste en una masa suspendida de un punto fijo. Esta masa está conectada a un resorte flexible que permite que la masa se mueva sin contacto con el suelo. Esta masa se balancea de un lado a otro, igual que un péndulo tradicional, pero al revés, con la masa más baja que el punto de suspensión. En la Figura (4), en la parte derecha se muestra el péndulo sobre una base móvil de dos ruedas [13], y en el lado izquierdo se consideran las fuerzas de energía cinética y potencial del mismo péndulo.

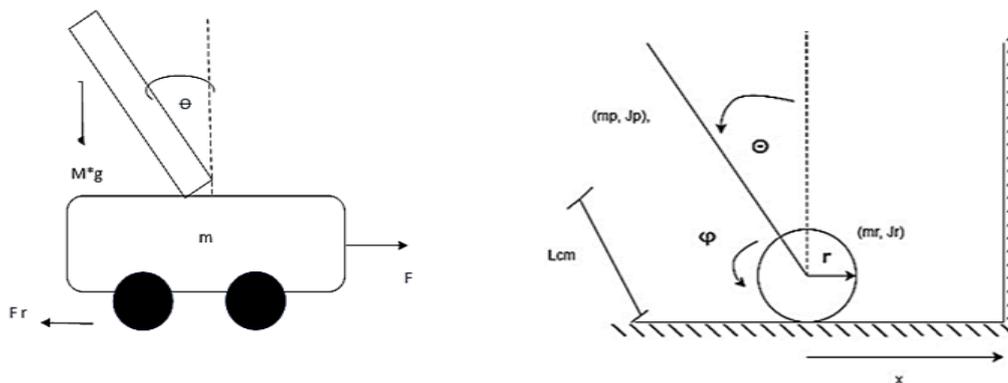


Figura 5 Modelos de péndulo invertido [43].

En la Tabla 1, se muestran las variables que permiten cuantificar y analizar las características dinámicas y cinemáticas del péndulo invertido; son fundamentales en la formulación de las ecuaciones de movimiento que describen el comportamiento del sistema.

Tabla 1 Variables del sistema dinámico [44].

Variables	Descripción	Unidades
J_p	Momento de inercia del péndulo respecto al centro de masa	$[Kg \cdot m^2]$
J_r	Momento de inercia de la rueda respecto al centro de masa	$[Kg \cdot m^2]$
m_p	Masa del péndulo	$[Kg]$
m_r	Masa de la rueda	$[Kg]$
M	Masa del carro	$[Kg]$
$\dot{\phi}$	Velocidad angular de la rueda	$[rad/s]$
$\dot{\theta}$	Velocidad angular del péndulo respecto a la vertical	$[rad/s]$
l	Distancia desde la base hasta el centro de la masa	$[cm]$
θ	Ángulo del péndulo en relación con la vertical	$^\circ$ grados
v	Velocidad a lo largo de la dirección tangente a la trayectoria circular.	$[km/h]$
r	Radio de la trayectoria circular	$[m]$
s	Longitud de arco	$[\theta \cdot r]$
ϕ	Desplazamiento angular	$[rad]$
g	Aceleración debida a la gravedad	$[m/s^2]$
F	Fuerzas aplicadas	$[Kg \cdot m]/s^2$
Fr	Fuerza de fricción	$[u \cdot N]$
μ_s	Coefficiente de fricción estática	$[N \cdot s/m]$
μ_e	Coefficiente de fricción dinámica	$[N \cdot s/m]$
K_s	Constante de escala.	$[V/Ue]$
K_t	Constante de torques.	$[Nm] / [A]$
K_e	Constante electromotriz.	V / rad
R	Resistencia del sistema.	$[\Omega]$
u	Valor de setpoint (Escala).	$[bits]$
T	Torque aplicado en la dirección de Theta	$[N \cdot m]$
$x(t)$	vector de estado en orden (n x 1).	$rad, \frac{rad}{s}, m, m/s$
$y(t)$	vector de salida en orden (p x 1).	$[rad]$

$\mathbf{u}_{(t)}$	Vector de entrada de orden (m x 1).	%
x	Posición.	[m]
\dot{x}	Velocidad	[m/s]
\ddot{x}	Aceleración	[m/s ²]
θ	Posición Angular	[s/r]
$\dot{\theta}$	Velocidad angular	[rad/s]
$\ddot{\theta}$	Aceleración angular.	[rad/s ²]
$\mathbf{A}_{(t)}$	El número de variables de estado, representado por n.	$n \cdot n$
$\mathbf{B}_{(t)}$	Siendo que el número de entradas es representado por m, es la matriz de entrada.	$n \cdot m$
$\mathbf{C}_{(t)}$	Siendo que el número de salidas es representado por p, es la matriz de transmisión directa.	$p \cdot n$
$\mathbf{D}_{(t)}$	Es la matriz de transmisión directa.	$p \cdot m$

2.2.6 Dinámica del Sistema

En la dinámica del sistema se analiza el cuerpo del sistema en movimiento por tanto se desarrollan los siguientes ítems:

2.2.6.1 Modelo matemático del sistema dinámico del péndulo invertido

Se desarrolla un modelo matemático que describe el comportamiento dinámico del sistema. A través de las ecuaciones de energía mecánica y energía potencial, refiere el comportamiento dinámico del sistema del péndulo invertido como indica en la Figura (5). Basado en principios físicos y matemáticos que representan las interacciones entre el péndulo y las fuerzas que actúan sobre él. Proporciona una representación cuantitativa de variables como posición, velocidad y aceleración, permitiendo un análisis y predicción de la dinámica del sistema [44], [45].

2.2.6.2 Ecuaciones de Movimiento

Son ecuaciones matemáticas que describen los cambios que sufre la posición, velocidad y aceleración en el tiempo, fundamentales en la física, mecánica entre otras ramas de ingeniería que ayudan a predecir y modelar sistemas de diferentes ámbitos [46].

➤ Energía Cinética

Forma de energía asociada al movimiento, depende tanto de la masa y la velocidad del objeto; a mayor masa y velocidad; mayor energía cinética. Esto indica que un pequeño aumento en la velocidad resulta ser muy significativo [46] como establece la ecuación (1).

$$E_C = \frac{1}{2} m \cdot v^2 \quad (1)$$

➤ Energía Potencial

Es la energía almacenada que puede convertirse en cinética o energía de movimiento cuando el cuerpo se mueve o cambia de posición y varía según la ubicación del objeto en el campo de fuerza [44].

$$E_p = m \cdot g \cdot h \quad (2)$$

➤ Lagrangiano

Es una herramienta utilizada en la mecánica clásica para describir y analizar sistemas físicos; se define como la diferencia entre la energía cinética (E_C) y la energía potencial (E_P) [44].

$$L = E_C - E_P \quad (3)$$

2.2.6.3 Fuerzas Generalizadas

Algunas fuerzas que actúan sobre los cuerpos para prevenir su caída son: Torque, fricción y el amortiguamiento viscoso. Estas fuerzas se pueden expresar en términos de velocidad lineal y se llevan a términos de desplazamiento x , como se indica en la expresión (6) [44].

$$v = r \omega \quad (4)$$

$$v = r \cdot \dot{\varphi} \quad (5)$$

$$\frac{v}{r} = \dot{\varphi} \quad \varphi = \frac{x}{r} \quad (6)$$

➤ Trabajo del torque del motor

Cantidad de energías que se transfieren en un sistema mecánico debido al giro del motor, provocando el trabajo mientras se realiza un torque o el sistema entra en función. Se calcula mediante la expresión (7) proporcional al torque del motor y radianes del ángulo [44].

$$w_T = T \cdot \theta \quad (7)$$

➤ Trabajo de la fuerza de fricción con el suelo

Fuerza que se opone al movimiento de un cuerpo, se considera por tener contacto con el suelo existen dos fuerzas de fricción: estática (F_e) y dinámica (F_d); la primera actúa al instante del movimiento mientras la segunda actúa cuando el cuerpo ya está en movimiento [44].

$$F_d = u_d \cdot N \quad (8)$$

$$F_e = u_e \cdot N \quad (9)$$

$$w_{FS} = F_d \cdot d \quad (10)$$

➤ Trabajo de la fuerza viscosa

Es una componente de la fuerza de fricción, actúa en el momento que el objeto está en movimiento oponiéndose proporcionalmente a la velocidad relativa entre el objeto y el fluido. Por tanto, la ecuación está conformada por: F_v y d que representa la dirección o movimiento [44].

$$w_{Fv} = F_v \cdot d \quad (11)$$

2.2.6.4 Euler Lagrange

La ecuación de Euler-Lagrange, mediante coordenadas generalizadas (qi) y velocidades generalizadas (\dot{qi}), nos permite obtener las ecuaciones de movimiento de un sistema al derivar parcialmente la variación de una función fundamental conocida como el Lagrangiano [44], tal como se describe en la ecuación (3). Por tanto, son necesarias las fuerzas externas que actúan sobre el robot tal como indica la expresión (12).

$$\frac{d}{dt} \left(\frac{dL}{d\dot{qi}} \right) - \frac{dL}{dqi} = \sum Qi \quad (12)$$

2.2.6.5 Esquema del modelo matemático

En la actualidad, se dispone del modelado matemático como herramienta para analizar y predecir el comportamiento de sistemas físicos a través de términos matemáticos, ecuaciones diferenciales, fórmulas o gráficos. Un modelo es una representación de un fenómeno físico en el que se conocen sus procesos y las relaciones entre ellos, lo que permite conocer las ecuaciones que lo describen [47] [48].

La naturaleza de un fenómeno físico permite la construcción de múltiples modelos matemáticos distintos entre sí, cuya eficacia dependerá del conocimiento de los procesos

que se buscan representar [49]. Por lo tanto, es fundamental conocer las diferentes etapas de construcción del modelado matemático:

A continuación, las etapas se presentan las etapas del proceso del modelo matemático a utilizar como indica Figura (6).

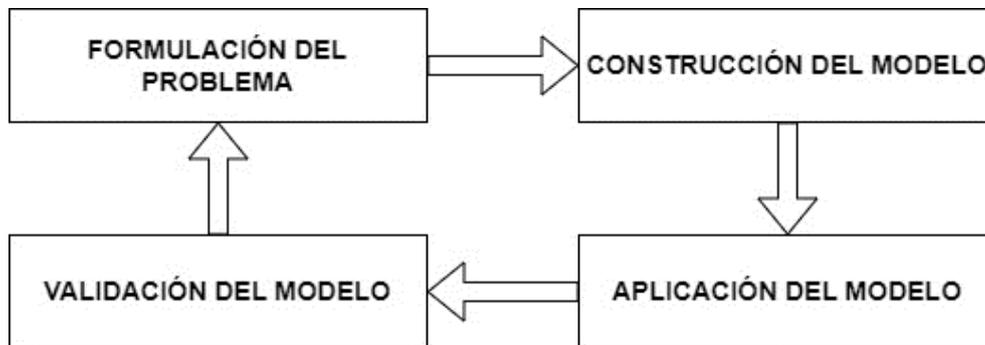


Figura 6 Esquema del Modelo Matemático [49].

- **Formulación del problema:** Se establece claramente el problema que se quiere resolver y se definen los objetivos del modelo.
- **Construcción del modelo:** Se elige el enfoque matemático y se desarrolla el modelo con las ecuaciones que representan el problema.
- **Aplicación del modelo:** Se utiliza el modelo para realizar predicciones o para tomar decisiones, lo que puede ayudar a resolver el problema original o aportar nuevas soluciones.
- **Validación del modelo:** Se verifica si el modelo es válido mediante la comparación de los resultados obtenidos con los datos reales y se ajusta el modelo si es necesario. Se utiliza el modelo para realizar predicciones o para tomar decisiones, lo que puede ayudar a resolver el problema original o aportar nuevas soluciones [47] [49].

2.2.7 Sistemas de control lazo abierto

Un sistema de lazo abierto como se observa en la Figura (7), muestra el control de un proceso sin recibir ninguna retroalimentación. Esto significa que el sistema no se ajusta automáticamente en respuesta a los cambios en el proceso, sino que depende de una entrada externa para ajustar el proceso. Esto es diferente de un sistema de control de lazo cerrado, que recibe retroalimentación del proceso para ajustar automáticamente el proceso [50].

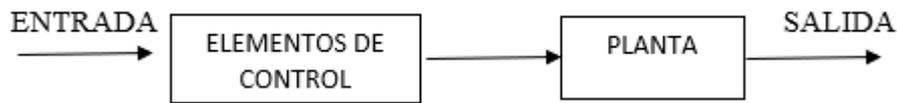


Figura 7 Diagrama Lazo Abierto [43].

2.2.8 Sistema de control lazo cerrado

Un sistema de control de lazo cerrado es un sistema dinámico en el cual la salida del sistema, es decir la variable controlada es comparada con la referencia deseada y la diferencia se utiliza para ajustar el sistema y mantener la salida en valores deseados. Ver Figura (8). En otras palabras es una forma eficaz de controlar y mantener un proceso a la condición deseada [50].

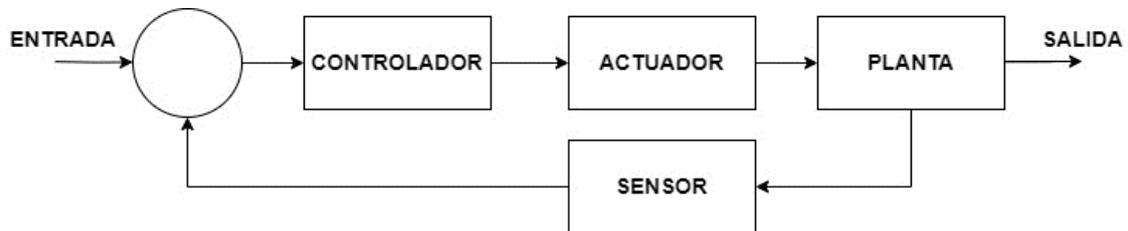


Figura 8 Diagrama Lazo Cerrado [51].

2.2.8.1 Variables de estado

En la teoría de control los estados, representan una variable que hace referencia a la condición actual de un sistema en un determinado tiempo [21]. Con el fin de comprender los elementos que conforman el modelo en espacio de estados. Cabe indicar que existe un conjunto de variables que describen completamente el estado actual del sistema en un momento determinado $x(t)$ [50]. Tal como indica la ecuación (13).

$$x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix} \quad (13)$$

También se conoce que un espacio de n dimensiones, cada dimensión corresponde a una de las variables de estado. Por tanto, cada punto en este espacio representa un vector de estado diferente y se encuentra en una posición específica en el espacio, en función de los valores tomados por sus variables de estado [52]. En la Figura (9) se muestran las representaciones de variables de estados junto a sus dimensiones.

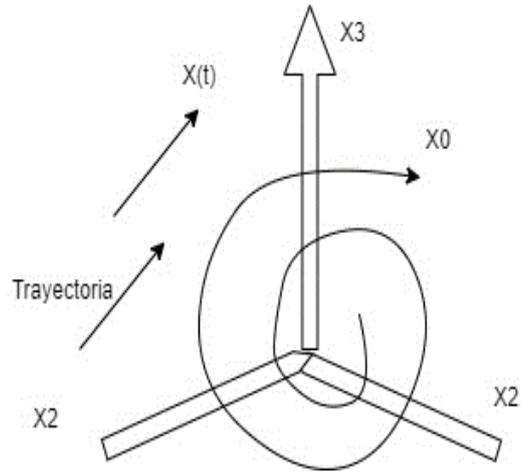


Figura 9 Representación de una variable en espacio de estados [45].

2.2.8.2 Ecuaciones de estados

La matriz de ecuaciones que describe tanto el comportamiento dinámico como las salidas del sistema es un conjunto de ecuaciones que involucran tanto el vector de estado como el vector de entrada como variables. Estas ecuaciones proporcionan una descripción completa del comportamiento del sistema [52].

Para entender el comportamiento matemático de la (Figura 8), se detallan las siguientes ecuaciones con n integradores y m entradas:

$$\dot{x}_1(t) = f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_m; t) \quad (14)$$

$$\dot{x}_n(t) = f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_m; t) \quad (15)$$

Por tanto, la salida es:

$$y_1(t) = g_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_m; t) \quad (16)$$

$$y_p(t) = g_p(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_m; t) \quad (17)$$

A continuación, se expresa las mismas ecuaciones de forma matricial:

$$\begin{bmatrix} \dot{x}_1(t) \\ \vdots \\ \dot{x}_n(t) \end{bmatrix} = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_m; t) \\ \vdots \\ f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_m; t) \end{bmatrix} \quad (18)$$

$$\begin{bmatrix} y_1(t) \\ \vdots \\ y_p(t) \end{bmatrix} = \begin{bmatrix} g_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_m; t) \\ \vdots \\ g_p(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_m; t) \end{bmatrix} \quad (19)$$

O a su vez:

$$\dot{x} = f(x, u, t) \quad (20)$$

$$y = g(x, u, t) \quad (21)$$

2.2.9 Linealización de sistemas no lineales

En la ingeniería de control, es común que los sistemas operen en torno a un punto de equilibrio con señales pequeñas, lo que permite aproximar el sistema no lineal a un sistema lineal dentro de un rango de operación deseado. Existen varias formas de linealizar los sistemas no lineales: linealización en serie de Taylor, linealización por control de retroalimentación de estados, linealización por transformada Laplace, entre otras [53]. También se conoce que el modelo lineal e invariante en el tiempo, permite analizar y diseñar controladores con una respuesta óptima en el rango de operación [52].

Es importante destacar que la aproximación lineal simplifica tanto el análisis como el diseño de sistemas, gracias a su naturaleza fácil de comprensión.

Para un modelo matemático lineal de un sistema no lineal, las variables del sistema experimentan pequeñas desviaciones con respecto a los estados de operación.

2.2.9.1 Linealización extendida o por espacios de Estados

Técnica utilizada en el campo del control para aproximar un sistema no lineal mediante un modelo lineal en torno a un punto de operación o un conjunto de condiciones de equilibrio. Identificar las ecuaciones (22) y (23) representaciones generales para el proceso de variables de estados.

x : Representa una variable de estado del sistema, que podría ser la posición.

u : Representa las variables de control, que se pueden manipular para influir en el comportamiento del sistema. A, B, C, D son las matrices del sistema [47] [52].

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

$$y = C x + D u \quad (23)$$

En el caso linealizado de que todos los elementos de la matriz son constantes, dichas ecuaciones se reducen a:

$$\dot{x}(t) = A x(t) + B u(t) \quad (24)$$

$$y(t) = C x(t) + D u(t) \quad (25)$$

En la Tabla (1) están los detalles de cada matriz con sus respectivos indicadores que conforman los elementos del siguiente diagrama: Ver Figura (10) del modelo lineal e invariante en el tiempo considerando los elementos constantes de las ecuaciones (24) y (25).

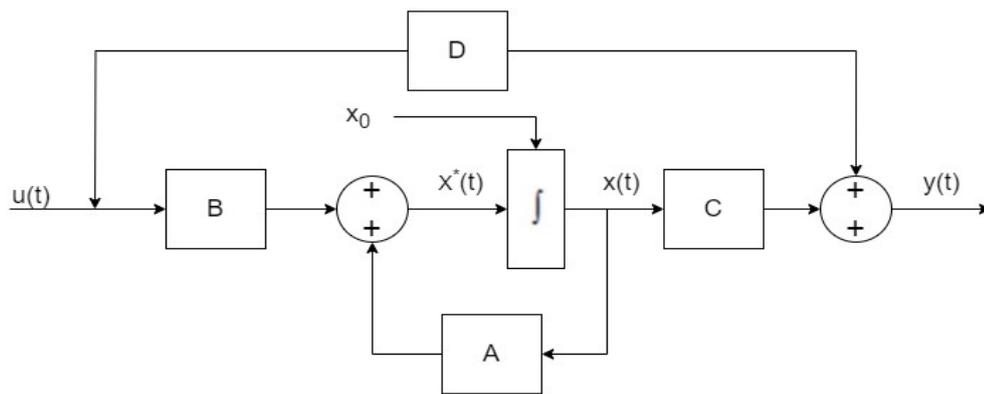


Figura 10 Diagrama de bloques en espacios de estados [53].

Para realizar el proceso linealizado se necesita los siguiente:

➤ Jacobianos

Para encontrar los jacobianos se deriva parcialmente cada expresión respecto a cada variable del sistema y formar las cuatro matrices como indican las siguientes ecuaciones:

$$A = \left. \frac{\partial f(x, u)}{\partial x} \right|_{\substack{x=x_e \\ u=u_e}} \quad (26)$$

$$B = \left. \frac{\partial f(x, u)}{\partial u} \right|_{\substack{x=x_e \\ u=u_e}} \quad (27)$$

$$C = \left. \frac{\partial g(x, u)}{\partial x} \right|_{\substack{x=x_e \\ u=u_e}} \quad (28)$$

$$D = \left. \frac{\partial g(x, u)}{\partial u} \right|_{\substack{x=x_e \\ u=u_e}} \quad (29)$$

2.2.9.2 Puntos de equilibrio del sistema

Los puntos de equilibrio se utilizan como referencia para aproximar el comportamiento del sistema. Para hacerlo, se calculan las derivadas parciales del sistema con respecto a las variables en los puntos equilibrados con aproximación del ángulo a 0° . Estas derivadas se utilizan para construir la matriz jacobiana, que proporciona información sobre la estabilidad y el comportamiento de la planta [53].

➤ Matrices linealizadas

Las matrices que identifican la planta del modelo linealizado en lazo abierto son:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \\ \frac{\partial f_3}{\partial u} \\ \frac{\partial f_4}{\partial u} \end{bmatrix} u \quad (30)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} u \quad (31)$$

2.2.10 Control automático

Control automático se define como la adquisición y preservación de un valor esperado de una variable en ejecución. Utiliza sensores para recopilar información del sistema, luego procesa la información para determinar cómo ajustar los parámetros para mantenerlos dentro de los límites. Se lleva a cabo mediante la aplicación de dispositivos de control. Estos dispositivos reciben una señal que les indica cómo ajustar el sistema para mantener los parámetros deseados [21].

2.2.11 Teoría de control

La teoría de control es un campo interdisciplinario que utiliza herramientas matemáticas e ingenieriles para analizar y diseñar sistemas dinámicos. Influye en el comportamiento de los sistemas, optimizando variables para lograr resultados deseados. Abarca una variedad de aplicaciones y se centra en cómo las señales de entrada afectan las variables de estado y salida. Proporciona métodos para crear controladores que aseguran estabilidad, precisión y respuestas eficientes en sistemas diversos [50] [54].

2.2.11.1 Elementos de un sistema de control

Un sistema de control está definido por varios componentes que actúan conjuntamente para otorgar una respuesta en relación con el objetivo a alcanzar. Todo sistema se caracteriza por estar formado de los siguientes elementos [47].

➤ Sensores

Sensor es un dispositivo que tiene la capacidad de contar, posicionar, medir o clasificar. En otras palabras, es un dispositivo que detecta cambios en una determinada variable y los convierte en información que se puede interpretar como un valor de medida [50].

➤ Actuadores

Son dispositivos que transforman la energía en movimiento o fuerza para controlar o modificar una variable en un sistema. La fuerza que proporciona un actuador tiene tres fuentes, que los clasifican en actuadores neumáticos, eléctricos o hidráulicos [21].

➤ Unidad de control

El controlador es el elemento responsable de comparar el valor de referencia (valor deseado) aplicado al sistema y ajustar las variables a controlar en un sistema dado. Es el encargado de tomar decisiones y emitir señales de control para mantener el sistema en un estado deseado [50].

➤ Planta

Se considera una planta a cualquier equipo o conjunto de elementos de un sistema físico, donde su objetivo es realizar una operación deseada, algunos ejemplos de sistemas físicos son: hornos de calentamiento, vehículos espaciales o reactores químicos [55].

En el presente proyecto la planta donde se aplica el control Lineal Cuadrático Gaussiano corresponde a un robot de equilibrio denominado Balancín, el cual deberá equilibrarse tratando de ignorar perturbaciones de ruido con distribución gaussiana.

➤ Control realimentado

Representa el ajuste o corrección al sistema en un tiempo real permitiendo un estado deseado al compensar las perturbaciones [45].

➤ Variable de control

Cantidad ajustable dentro del proceso de corrección del control deseado [50].

➤ **Entrada**

Señal proporcionada al sistema que influye en su comportamiento.

➤ **Salida**

Respuesta del sistema que permite comparar la entrada, perturbaciones y correcciones si influye un control de realimentación.

2.2.12 Componentes electrónicos y de comunicación

En este apartado se detallan de forma general los componentes de electrónica y telecomunicación utilizados en la propuesta:

➤ **Giroscopio**

Es un dispositivo mecánico que mide la orientación o la velocidad angulares de un objeto. Funciona mediante el principio de conservación del momento angular, donde un rotor gira a alta velocidad en una dirección determinada, y cuando el objeto que lo contiene gira, el eje del rotor se mantiene fijo en su orientación original en el espacio, lo que permite medir la velocidad angular y la orientación del objeto. Es más preciso que el acelerómetro, pero acumula error con el tiempo [56] [57].

➤ **Acelerómetro**

Es un tipo de sensor que se utiliza para medir la aceleración lineal de un objeto. Es menos preciso que el giroscopio en movimiento, su señal es ruidosa pero no acumula más error en el tiempo como el giroscopio [57].

➤ **Encoder**

Es un dispositivo utilizado en sistemas de control de movimiento para medir la posición, velocidad y dirección de un objeto. Puede ser rotatorio o lineal y funciona mediante la detección de pulsos generados por un disco codificado o una cinta magnética, que es leída por un sensor óptico o magnético, es responsable de contar o reproducir a partir del movimiento rotacional de su eje, convirtiendo los movimientos rotacionales o desplazamientos lineales en impulsos eléctricos de onda cuadrada o senoidal, generando una cantidad de pulsos por rotación precisa a lo largo de su rotación [57].

➤ **Actuadores.**

Los actuadores son los encargados de realizar las operaciones asignadas por el microcontrolador una vez analizado los parámetros necesarios [57].

➤ **Motor**

Es aquella herramienta que convierte la energía eléctrica o mecánica a energía cinética y genera movimientos. Este se acciona mediante interacción de campos magnéticos y corrientes eléctricas que impulsan el movimiento de la máquina [43].

➤ **Microcontrolador**

El microcontrolador se encarga de leer y ejecutar los programas introducidos en él. De esta forma puede controlar todo el proceso que se está dando en el sistema, convirtiéndose así en el cerebro del robot.

2.2.13 Comunicación Bluetooth.

La comunicación bluetooth empezó en 1994 en base a la necesidad de comunicaciones inalámbricas. Esta tecnología facilita la comunicación incluso en situaciones donde hay obstáculos y distancias significantes, tratándose de hasta 100 metros. Un ejemplo es la transmisión de archivos musicales desde un teléfono móvil a un estéreo para automóvil [58].

2.2.13.1 Ventajas y desventajas de Bluetooth

El propósito principal fue aportar con avances tecnológicos en la humanidad, permitiendo conexiones inalámbrica entre dispositivos y computadoras. Aunque tenía ventajas notables, como la transferencia de archivos sin cables, su limitación crítica era la distancia de conexión limitada y la susceptibilidad a obstáculos, lo que eventualmente condujo a mejoras tecnológicas y su reemplazo [58].

2.2.14 Inestabilidad de un sistema

Rudolf E. Kalman, un influyente matemático y científico de la computación del siglo XX, dejó una marca indeleble en la teoría de control y sistemas dinámicos. Sus contribuciones, como el renombrado filtro de Kalman, transformaron nuestra comprensión y manejo de sistemas afectados por ruido y perturbaciones. Kalman introdujo conceptos esenciales como la expresión (32) donde "A" es la matriz de coeficientes de un sistema y "s" es una variable compleja que representa el dominio de la frecuencia. Esta expresión está intrínsecamente vinculada con los valores propios de la matriz y desempeña un papel fundamental al analizar y diseñar sistemas de control estables y eficientes. Además, permite identificar o verificar la estabilidad de un sistema al observar la ubicación de los valores propios en el plano complejo. Gracias a los aportes de Kalman y sus contemporáneos, se ha allanado el camino para la ingeniería moderna de control, abriendo puertas a la automatización, la robótica y la exploración espacial, entre otros campos [59].

$$\det(sI - A) \quad (32)$$

$$\det(sI - A) = s^4 + s \cdot s^3 + b \cdot s^2 + c \cdot s + d \quad (33)$$

Por tanto, ' a, b, c, d ', son coeficientes calculados a partir de los elementos de la matriz "A".

2.2.15 Controlabilidad y Observabilidad

En el siglo XX, la controlabilidad y la observabilidad surgieron en la teoría de sistemas dinámicos y el desarrollo de métodos de control. Ingenieros, matemáticos y científicos de diversas disciplinas colaboraron en su formulación, a medida que la automatización y el control avanzaban. Richard Bilman, Rudolf Kalman y Lev Pontryagin contribuyeron al desarrollo temprano del tema. La evolución de la teoría de control, junto con avances en matemáticas y tecnología, profundizó la comprensión y aplicación de estos conceptos en áreas de ingeniería y ciencia de control [52].

2.2.15.1 Controlabilidad

El criterio de Controlabilidad de Kalman se basa en la matriz controlable, se forma de las matrices $B, AB, A^2B, \dots, A^{n-1}B$, donde n es la dimensión del vector de estado x . Si el rango de esta matriz es igual a la dimensión del espacio de estados del sistema, entonces se puede afirmar que es completamente controlable [52]. También es importante considerar que todas las variables de estados deben ser controlables en alguna señal de control sin restricciones cumpliendo la siguiente condición:

$$\xi = [B, AB, A^2B, \dots, A^{n-1}B] = n \quad (34)$$

2.2.15.2 Observabilidad

La observabilidad se refiere a la capacidad de deducir el estado actual de un sistema dinámico únicamente a partir de las mediciones de sus salidas. Un sistema es observable si su estado puede ser determinado de manera única y precisa a lo largo del tiempo. Es esencial para diseñar sistemas de control que puedan estimar el estado interno del sistema, incluso si no todas las variables de estado son directamente medibles. Se forma de las matrices $C, CA, CA^2, \dots, CA^{n-1}$, donde n es la dimensión del vector de estado x estrategias de observación eficientes para cumpliendo de la siguiente condición [52]:

$$\sigma = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} = n \quad (35)$$

2.2.16 Observador de estados

Un observador de estados en el contexto de sistemas dinámicos es un algoritmo utilizado para estimar o rehacer los estados no medidos de un sistema dependiendo de las señales de entrada y salida. Utiliza ecuaciones matemáticas que se encargan del monitoreo del modelo matemático [40]. De la ecuación (22) de estado inicial lineal se define el observador y la ecuación de estado lineal de n dimensiones con sus entradas $u(t)$ e $y(t)$ como indican las ecuaciones (36). Además, se evidencia el término $L[y(t) - \hat{y}(t)]$ que representa la matriz de ganancia L , con un término de error $y(t) - \hat{y}(t)$ considerado a impulsar la estimación de estado $\hat{x}(t)$ al estado real $x(t)$ [40]. Ver Figura (11).

$$\dot{x}(t) = A\hat{x}(t) + Bu(t) + L[y(t) - \hat{y}(t)] \quad (36)$$

$$y(t) = C\hat{x}(t) \quad (37)$$

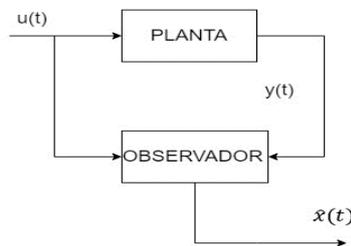


Figura 11 Observador de estados [58].

2.2.17 Ganancia de control (K)

Mediante la matriz de transformación, $T = M V$ se calcula la ganancia de realimentación que ajusta los valores realimentados del sistema, es fundamental para la estabilidad porque depende de los valores altos para una divergencia u oscilaciones no deseadas y bajos provoca tardíos o anula su respuesta [45].

Ya comprobado las condiciones de controlabilidad y estabilidad, se analiza los polinomios de la matriz A de la ecuación (33) y (34) para determinar la transformación siguiente:

$$W = \begin{bmatrix} a_{n-1} & a_{n-2} & \dots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \dots & 1 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ a_1 & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix} \quad (38)$$

En la ecuación (38), se describen los coeficientes de la expresión polinómica característica que conforman la matriz de ganancia de realimentación de estado (39) [40].

$$K = [a_n - a_n \quad a_{n-1} - a_{n-1} \quad \dots \quad a_2 - a_2 \quad a_1 - a_1] T^{-1} \quad (39)$$

2.2.18 Control de referencia

Es un procedimiento diseñado para que una variable se ajuste y siga de manera necesaria una señal de referencia. Es decir, permita ver la diferencia entre la entre la señal real y la señal actual. De aquí depende el contexto de la señal de control y su aplicación al seguimiento más aproximado a la referencia, reduciendo el índice de error [40] [55] . Su expresión matemática se detalla a continuación:

$$\dot{\xi} = r(t) - y(t) \quad (40)$$

$$u(t) = -K x(t) + k_1 \xi(t) \quad (41)$$

En la expresión (40) describe la entrada referida $r(t)$ que debe seguir $y(t)$, (ξ) es la integral de error de seguimiento que corrige desviaciones acumuladas de la referencia deseada [40]. Entonces la ley de control está dada por:

$$u(t) = -[K - k_1] \begin{bmatrix} x(t) \\ \xi(t) \end{bmatrix} \quad (42)$$

La retroalimentación de estados implica el vector de estado elevado (n+1) en lazo abierto $x(t)$ con la variable de estado del integrador $\xi(t)$ [40]. Con el proceso antes descrito se establece la ecuación de lazo cerrado (n+1) según sus dimensiones.

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\xi}(t) \end{bmatrix} = \begin{bmatrix} A - BK & B k_1 \\ C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \xi(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(t) \quad (43)$$

$$y(t) = [C \quad 0] \begin{bmatrix} x(t) \\ \xi(t) \end{bmatrix} \quad (44)$$

2.2.19 Control óptimo (LQR)

Control óptimo es importante en la teoría de sistemas dinámicos, por ser el encargado de la estrategia de control que minimiza o maximiza una función de costo, esencial para temas de ingeniería de control con diferentes índices y desempeños [2] [40].

Regulador Óptimo Cuadrático lineal

Es aquel método que realiza un procedimiento sistemático calculando la matriz de ganancia de control de realimentación de estados. Por ejemplo, para resolver problemas

cuadráticos se considera que el regulador óptimo esté dado por las ecuaciones de estados siguientes:

$$\dot{x} = A x + B u \quad (45)$$

Se determina la matriz \mathbf{K} del vector de control óptimo.

$$u(t) = -Kx(t) \quad (46)$$

Con la finalidad de minimizar la función de costo:

$$J = \int_0^{\infty} (x * Qx + u * Ru) dt \quad (47)$$

Donde \mathbf{Q} representa la matriz hermítica definida positiva o semidefinida positiva o a su vez una matriz simétrica real y \mathbf{R} es la matriz simétrica definida positiva o simétrica real que define la ecuación de costo de energía de la señales de control \mathbf{J} .

Cabe indicar que las matrices \mathbf{Q} y \mathbf{R} son las que determinan la importancia relativa del error y del coste de energía. Además, la ley de control lineal obtenida mediante la ecuación (46) es la ley de control óptimo en los casos de vectores de control $\mathbf{u}(t)$ no restringidos. Entonces, si se determinan los elementos desconocidos de la matriz \mathbf{K} para que minimice la función de costo, la expresión (46) es óptima para cualquier estado inicial $x(0)$. Ver la Figura (12).

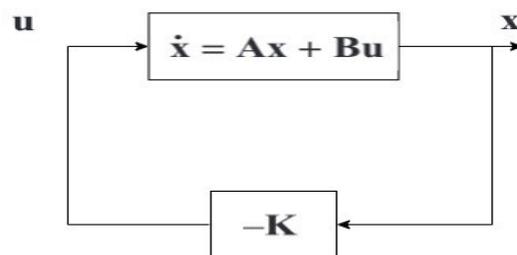


Figura 12 Sistema Regulador Óptimo [58].

A continuación, se resuelve el problema de optimización sustituyendo la ecuación (46) en la (45), lo que resulta la expresión siguiente:

$$\dot{x} = A x - BKx = (A - BK)x \quad (48)$$

Posteriormente, se considera estable la matriz $\mathbf{A} - \mathbf{BK}$, o que sus valores propios poseen parte reales negativas. Al sustituir la expresión (46) en la (47) resulta la expresión (49):

$$J = \int_0^{\infty} (x^* Q x + x^* K^* R K x) dt$$

$$\int_0^{\infty} x^* (Q + K^* R K) x dt \quad (49)$$

Al establecer la igualdad:

$$x^* (Q + K^* R K) x = -\frac{d}{dt} (x^* P x) \quad (50)$$

Donde \mathbf{P} representa la matriz hermitica definida positiva o simétrica real. Consecuente a la siguiente expresión:

$$x^* (Q + K^* R K) x = -\dot{x}^* P x - x^* P \dot{x} = -x^* [(A - BK)^* P + P(A - BK)] x \quad (51)$$

Al contrastar ambos lados de la expresión anterior y considerando que debe ser válida para cualquier valor de x , se requiere que:

$$(A - BK)^* P + P(A - BK) = -(Q + K^* R K) \quad (52)$$

Es posible demostrar que si $(A - BK)$ es una matriz estable, existe una matriz P definida positiva que cumple con la ecuación (52).

Por lo tanto, el proceso implica la determinación de los elementos de \mathbf{P} a partir de la ecuación (52) y se comprueba su definición positiva. Es importante indicar que más de una matriz \mathbf{P} puede cumplir esta condición. De tal manera que al resolver la ecuación y encontrar una matriz \mathbf{P} definida positiva, se garantiza la estabilidad sistema. Si otras matrices satisfacen la ecuación y no son positivas, deben ser descartadas.

Se procede a calcular el índice de comportamiento J :

$$J = \int_0^{\infty} (x^* Q + K^* R K) x dt = -x^* P x \Big|_0^{\infty} = -x^*(\infty) P x(\infty) + x^*(0) P x(0) \quad (53)$$

Considerando que todos los valores propios de $(A - BK)$ tienen parte real negativa, esto implica que $x(t)$ tiende a cero a medida que t tiende al infinito. Por tanto, se obtiene:

$$J = x^*(0) P x(0) \quad (54)$$

Lo que define que el índice de comportamiento J se obtiene mediante condiciones iniciales $x(0)$ y P . Para seguir con la solución del control óptimo cuadrático, se debe

considerar que R es una matriz hermítica definida positiva o simétrica real, como indica la expresión:

$$R = T^*T \quad (55)$$

Donde T representa una matriz no singular. Y posteriormente representa la ecuación (52) en la expresión (56):

$$(A^* - K^*B^*)P + P(A - BK) + Q + K^*T^*TK = 0 \quad (56)$$

O también:

$$A^*P + PA + [TK - (T^*)^{-1}B^*P]^*[TK - (T^*)^{-1}B^*P] - PBR^{-1}B^*P + Q = 0 \quad (57)$$

Para minimizar J respecto a K se necesita minimizar:

$$x^*[TK - (T^*)^{-1}B^*P]^*[TK - (T^*)^{-1}B^*P]x \quad (58)$$

Como indica la última expresión no negativa, el mínimo ocurre cuando es cero o en otros casos:

$$TK = (T^*)^{-1}B^*P \quad (59)$$

Por consiguiente, la ecuación (60) permite obtener la matriz óptima K .

$$K = T^{-1}(T^*)^{-1}B^*P = R^{-1}B^*P \quad (60)$$

Es así como la ley de control óptimo para problemas de control óptimo cuadrático como la matriz K de la expresión (60), es lineal cuando el índice de comportamiento esta dado por la ecuación (47) y se puede expresar de la siguiente manera:

$$u(t) = -Kx(t) = -R^{-1}B^*Px(t) \quad (61)$$

Por último, para la ecuación matricial reducida de Riccati, la matriz P en la expresión (60) debe satisfacer la expresión (52) como se indica a continuación:

$$A^*P + PA - PBR^{-1}B^*P + Q = 0 \quad (62)$$

Cabe indicar que si la matriz $A - BK$, es estable la metodología anterior dará resultados correctos. También se destaca que, si el índice de rendimiento se formula en relación con el vector de salida en lugar del vector d estado, es decir:

$$J = \int_0^{\infty} (y^*Qy + u^*Ru)dt \quad (63)$$

En estos casos el índice se debe modificar en la ecuación de salida.

$$y = Cx \quad (64)$$

A

$$J = \int_0^{\infty} (x^*C^*QCx + u^*Ru)dt \quad (65)$$

Y replicando los pasos anteriores se obtiene la matriz óptima K .

2.2.20 Ruido blanco y ruido gaussiano

En este ítems se menciona de forma general los ruidos que perturban el sistema de control LQG.

2.2.20.1 Densidad espectral de potencia

La densidad espectral de potencia (PSD), considera como se distribuye la energía del ruido en función de la frecuencia, un aspecto muy importante al momento de perturbar el sistema. Se considera proceso aleatorio $x(t)$ por $S_x(f)$ para la transformada de Fourier de $R_x(\tau)$ función de autocorrelación como indica en el apartado siguiente:

$$S_x(f) = F\{R_x(\tau)\} = \int_{-\infty}^{\infty} R_x(\tau) e^{-2j\pi f\tau} df \quad (66)$$

La ecuación (56) refiere al proceso matemático de la transformada de Fourier de la función de autocorrelación para obtener (PSD) en el dominio de la frecuencia (df) respetando las condiciones para $S_x(f) \geq 0$ y $S_x(-f) = S_x(f)$ en toda frecuencia [40].

2.2.20.2 Ruido blanco

Es una señal aleatoria que tiene una distribución de probabilidad similar e invariable en un determinado instante de tiempo. Su expresión matemática $X(t)$ es denominada ruido

blanco si la densidad espectral de potencia $S_x(f)$ es constante $\frac{N_0}{2}$ para todas las frecuencias [60].

$$S_x(f) = \frac{N_0}{2} \tag{67}$$

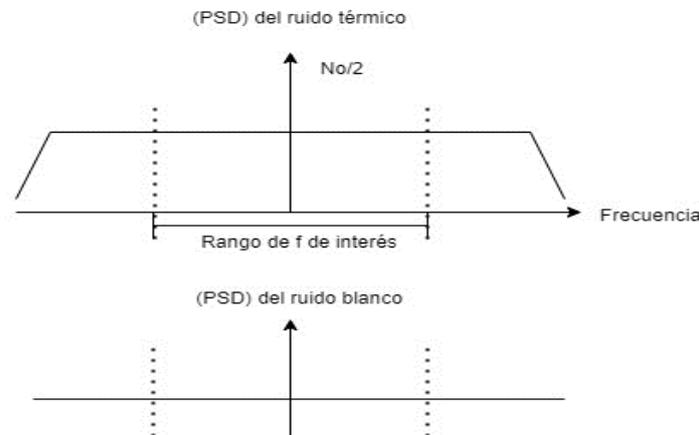


Figura 13 (PDS) Ruido térmico y Ruido blanco [61].

2.2.20.3 Ruido blanco gaussiano

Tipo específico de ruido blanco con amplitudes de distribución uniforme, es completamente aleatorio y sus valores no influyen en el tiempo. Su expresión matemática depende de las distribuciones de probabilidad [61]. Ver Figura (13).

2.2.21 Estimador Lineal Cuadrático (LQE) o Filtro de Kalman

Y LQE es el acrónimo de "Linear LQ Estimation" (estimación lineal LQ). La estimación lineal LQ es un método utilizado en ingeniería y matemáticas para estimar el estado de un sistema dinámico. Se basa en el enfoque de control LQ, que se utiliza para minimizar el error entre el comportamiento deseado del sistema y el comportamiento real del sistema, se utiliza a menudo en aplicaciones donde es importante tener una buena estimación del estado del sistema, especialmente en sistemas con incertidumbres o ruido.

La estimación LQ utiliza una función de coste óptima para calcular la estimación del estado del sistema, se utiliza en sistemas dinámicos donde es importante tener una buena estimación del estado del sistema y minimizar el error entre el comportamiento deseado del sistema y el comportamiento real del sistema. Por ejemplo, puede utilizarse en sistemas de control de procesos industriales, sistemas de control de vehículos autónomos o sistemas de control en robots [6]. Ver Figura (14).

2.2.21.1 Filtro de Kalman

El filtro de Kalman es un algoritmo matemático que se encarga de la Predicción y la Corrección de sistemas, considerando las incertidumbres de estimaciones del modelo matemático en forma precisa y ruidosa. Es muy común en estadísticas y sistemas de control, permite estimar los parámetros de un sistema a partir de mediciones inexactas, permitiendo estimaciones. Radica en su capacidad para estimar parámetros del sistema que no pueden medirse u observarse con precisión [62]. Se basa en dos etapas principales [44] [63]:

➤ Predicción

Se utiliza para predecir el estado futuro del sistema A_k [62].

$$\hat{x}_k = A\hat{x}_{k-1} \quad (68)$$

Se le agrega un vector de modificación llamado vector de control \vec{u} , y el estado \hat{x} es afectado por el vector de control y la matriz que representa la señal de control B_k :

$$\hat{x}_k = A_k\hat{x}_{k-1} + B_k\vec{u}_k \quad (69)$$

Se observa la variación de la matriz de covarianza $cov(\hat{x}_k)$ entre dos estados tal como se muestra en la ecuación (70):

$$cov(\hat{x}_k) = P_k \quad cov(A_k\hat{x}_k) + B_k\vec{u}_k \quad (70)$$

Se suma la covarianza del ruido Q_k :

$$P_k = F_k P_{k-1} A_k^t + Q_k \quad (71)$$

➤ Corrección

Utiliza mediciones recientes para corregir la predicción. Se refiere a la diferencia entre la medición real en el tiempo k (z_k) y la estimación de la medición en el tiempo k [62], basado en el estado anterior ($\hat{x}_{k|k-1}$).

$$\tilde{y}_k = z_k - \hat{z}_{k|k-1} = z_k - H_k\hat{x}_{k|k-1} \quad (72)$$

Innovación o covarianza de Innovación (S_k). se halla mediante la matriz de observación (H_k^T), matriz de covarianza en el tiempo ($P_{k|k-1}$) y el ruido de las mediciones R_k :

$$S_k = H_k P_{k|k-1} H_k^T + R_k \quad (73)$$

Se obtiene la ganancia óptima de Kalman mediante la siguiente ecuación:

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \quad (74)$$

Fórmula utilizada para calcular el estado estimado, después de la corrección.

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \quad (75)$$

Matriz de covarianza estimada:

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (76)$$

Las matrices H_k y F_k se consiguen del filtro de Kalman en sistema extendido no lineal, el cual aproxima linealmente alrededor del estado estimado [62].

$$H_k = \left. \frac{\partial h}{\partial x} \right|_{x_{k-1}} \quad (77)$$

$$F_k = \left. \frac{\partial f}{\partial x} \right|_{x_{k-1}} \quad (78)$$

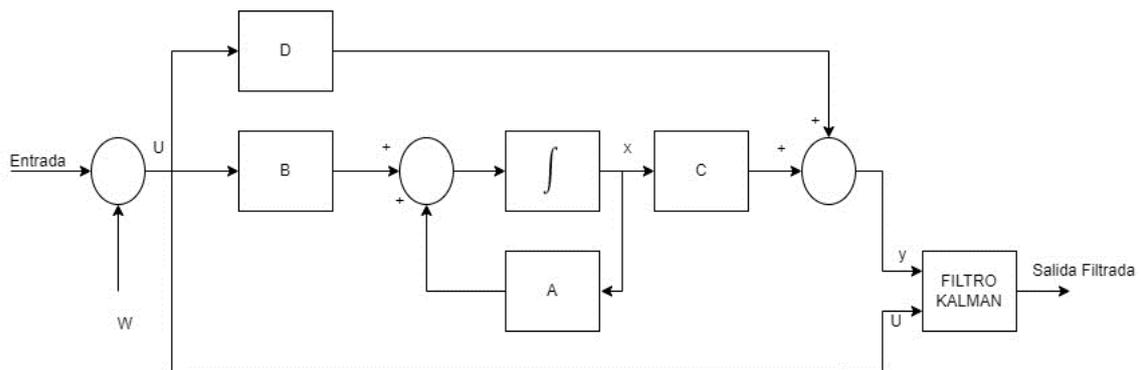


Figura 14 Filtro de Kalman [33].

➤ Filtro de Kalman Bucy

Este filtro no emplea un algoritmo predictor y corrector para las estimaciones de estados, solo se encarga de integrar una ecuación diferencial de Riccati a lo largo del tiempo [33].

2.2.22 Control Lineal Cuadrático Gaussiano (LQG)

Un controlador lineal cuadrático Gaussiano (LQG) es un tipo de sistema de control diseñado para optimizar la respuesta de un sistema dinámico a los cambios en la entrada.

Está basado en el principio de control cuadrático, que se usa para diseñar un sistema con una función de costo cuadrática de entrada y salida. El controlador utiliza una función de costo cuadrática para medir la diferencia entre la entrada deseada y la salida del sistema y luego ajusta los parámetros del sistema para minimizar esta diferencia. El controlador también tiene la capacidad de estimar los parámetros del sistema a través de una función de costo cuadrática. La estimación de los parámetros del sistema se realiza utilizando un filtro de Kalman, que es una técnica de estimación de estado que utiliza observaciones del sistema para estimar el estado interno del sistema. Este controlador es la combinación de un controlador LQR y un Estimador de Kalman es decir un estimador lineal cuadrático LQE, los cuales pueden ser diseñados de forma autónoma, o uno con relación a otro. El modelo del proceso se muestra en la ecuación (79) y (80), donde se representa las ecuaciones de entrada y salida con perturbación. Se considera una herramienta útil para el diseño de sistemas de control automático y se usa ampliamente en campos como la robótica, la aeronáutica y el control de procesos [33].

$$\dot{x}(t) = Ax(t) + Bu(t) + W_d(t) \quad (79)$$

$$y(t) = Cx(t) + Du(t) + W_n(t) \quad (80)$$

Donde:

$W_d =$ Ruido del sistema

$W_n =$ Ruido del estado

En la Figura (15) se muestra el diagrama de bloques del controlador Lineal Cuadrático Gaussiano LQG con los ruidos antes indicado [40].

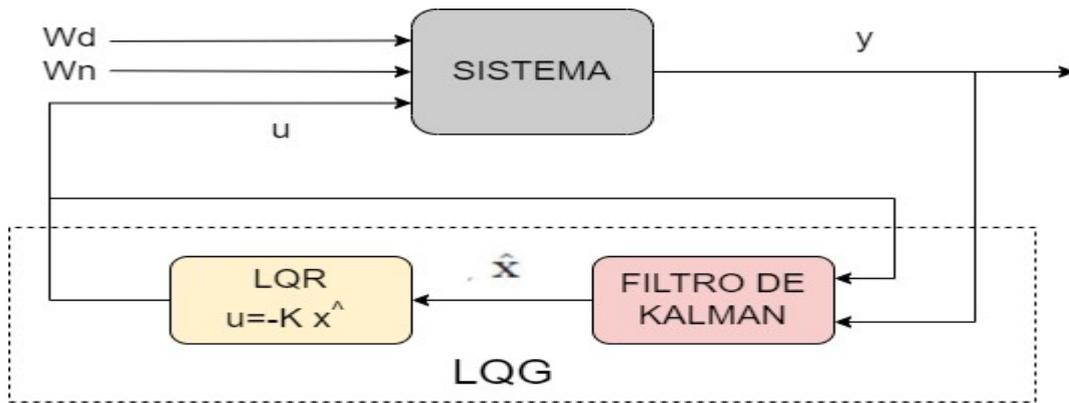


Figura 15 Diagrama del control LQG [40].

Se considera:

L = constante del filtro de Kalman.

K = constante del regulador LQR.

Que ambos combinan la ley de control óptima $u(t) = -Kx(t)$ con la estimación óptima del estado $\hat{x}(t)$. Finalmente la ley de control del controlador LQG es:

$$u(t) = -K\hat{x}(t) \quad (81)$$

Donde:

K es la matriz de ganancias de control obtenida del control LQR.

$\hat{x}(t)$ es la estimación óptima del estado obtenida del estimador LQE.

2.2.22.1 Ventajas y desventajas del control LQG

Se presentan a continuación las ventajas y desventajas del método de control LQG:

Ventajas:

- El LQG es capaz de rechazar perturbaciones y seguir la referencia, resaltando la utilidad en sistemas robustos.
- Minimiza la varianza del error de estimación a partir del análisis de los ruidos.
- Aplicación en sistemas robustos debido a la capacidad de rechazar perturbaciones y el principio de modelamiento interno o seguimiento de referencia.
- Reduce al mínimo la variación del error de estimación al caracterizar los ruidos.

Desventajas:

- Si el modelo utilizado para el control no se ajusta exactamente a los parámetros de la curva, o si el sistema no es realmente lineal, el controlador resultante puede degradarse y el sistema puede volverse inestable [50].
- Cuando se trabaja con estructuras mecánicas móviles, el control LQG presenta limitaciones en términos de estabilización. Para abordar este problema, se implementa un filtro observador y se diseña una matriz de ganancia de realimentación como solución.
- Si hay discrepancias debido a imprecisiones en el modelado de los parámetros de una curva lineal o no linealidades (es decir, si el sistema real no es verdaderamente lineal), el controlador resultante se verá comprometido y el sistema podría incluso volverse inestable. [33], [64], [65].

2.3 Marco teórico

En la sección del marco teórico se presentan una variedad de documentos de investigación que han sido analizados para servir como guía en relación con la propuesta de titulación actual.

En el Repositorio de Tesis de la Universidad Católica de Santa María de la Facultad de Ciencias e Ingenierías Físicas y Formales se encuentra el trabajo: “DISEÑO Y ANÁLISIS COMPARATIVO DE UN SISTEMA PÉNDULO INVERTIDO APLICANDO TECNICAS DE CONTROL LINEAL CUADRÁTICO GAUSSIANO” realizado por Rosa Ramos y Hansem Julián de Arequipa-Perú en el año 2022, recalcan el valioso e importante desarrollo de la investigación de diseño de controladores en espacios de estados realimentados óptimos y robustos en cuanto a ruido y perturbaciones [40]

En artículo de la Revista Saplenza se encuentra el tema “DISEÑO ÓPTIMO DE UN CONTROLADOR LQG PARA UN SISTEMA MECÁNICO DE UN PÉNDULO INVERTIDO”, realizado por Luca Fourieri, en el 2022, en la Jornada Internacional de Estudios Interdisciplinarios de la Universidad Técnica Luis Vargas Torres de Esmeraldas, se comprueba que tanto LQR como LQG son capaces de controlar el éxito de un sistema perturbado, dando resultado satisfactorio [6].

En Actas de Machine Learning Research se encuentra el artículo “SAMPLE COMPLEXITY OF LINEAR QUADRATIC GAUSSIAN (LQG) CONTROL FOR OUTPUT FEEDBACK SYSTEMS” realizado por Zeilinger MN. en la Universidad de British Columbia, Canadá en el año 2021, se estudia una clase de problemas Gaussianos

Cuadráticos Lineales (LQG) parcialmente observados con dinámica desconocida. Una complejidad de muestra de extremo a extremo vinculada al aprendizaje de un controlador LQG robusto para plantas estables de bucle abierto[12].

En la tesis de Maestría en Ingeniería Eléctrica, con el Programa de Posgrado de Ingeniería Eléctrica, “PROJETO E IMPLEMENTAÇÃO DE CONTROLADOR LQG/LTR EM TEMPO REAL PARA UM SISTEMA DE TRÊS TANQUES ACOPLADOS” realizada por Fabio Henrique Dermendjian. en la Universidad Estadual Paulista “Julio de Mesquita Filho” de Sorocaba en el 2020, destaca el uso de controladores óptimos para sistemas acoplados [10] .

En el año 2019, Asfihani Tahiyatul en su artículo International Conference on Mathematics: Pure, Applied and Computation se publica el tema “COMPARISON OF LQG AND ADAPTIVE PID CONTROLLER FOR USV HEADING CONTROL” se enfoca en la comparación de LQG y PID adaptativo para el controlador de rumbo USV. Donde se demuestra que LQG es uno de los mejores controladores para controlar el vehículo de superficie [20].

En el trabajo de titulación “DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR ÓPTIMO LQG, PARA UN SISTEMA DE PÉNDULO INVERTIDO APLICADO EN UN EQUIPO LEGO MINDSTORMS.” Realizada por Carlos German Pillajo Angos, desarrollado en Quito Universidad Politécnica Salesiana, en el 2018 donde plantea aproximación de las constantes mediante algoritmos heurísticos basados en disminuir el error en estado estacionario [33].

CAPÍTULO III

3. DESARROLLO

En este capítulo se desarrolla la propuesta del proyecto, consta el modelo matemático, linealización por espacios de estados, diseño del servo controlador LQG, simulación en MATLAB© Simulink, diseño electrónico del robot y algoritmo de control LQG en lenguajes de programación de alto nivel.

3.1 Componentes de la propuesta

En este apartado se detallan los elementos físicos y lógicos que se utilizan dentro de la propuesta.

3.1.1 Proceso de la planta robótica

En la Figura (16), se muestra la estructura de los principales elementos del Robot Balancín que son sus componentes electrónicos que accionan al debido funcionamiento [9].

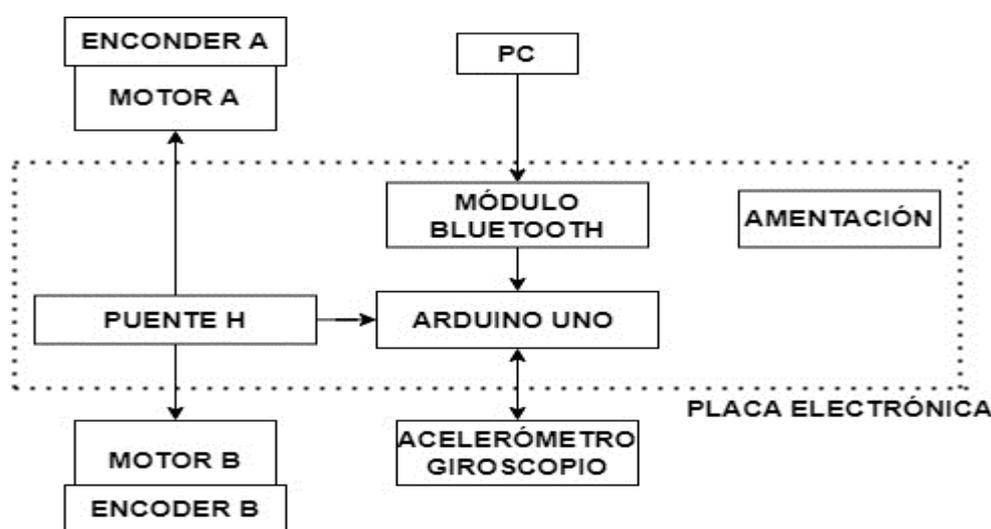


Figura 16 Proceso del trabajo de la planta [9].

3.1.2 Componentes físicos (Hardware)

Los componentes de hardware desempeñan roles fundamentales en el funcionamiento del robot balancín, desde los motores y sensores hasta los controladores y la estructura mecánica. En esta descripción se detallan los elementos necesarios que constituyen la base física del sistema propuesto [9].

3.1.2.1 Giroscopio- Acelerómetro MPU6050

Se utiliza el MPU 6050 que contiene un giroscopio y un acelerómetro de triple eje de sistemas microelectromecánicos (MEMs). Estos sensores de velocidad angular (giroscopio) en ejes X, Y y Z presentan un rango de escala completa configurable por el usuario de: $\pm 250, \pm 500, \pm 1000$ y $\pm 2000^\circ/\text{seg}$. Este dispositivo incluye convertidores analógicos-digitales (ADC) de 16 bits que facilita el muestreo simultáneo de datos del giroscopio, con una corriente de operación de 3.6mA y una corriente de espera de $5\mu\text{A}$ [66].

En la figura (17) se muestra la conexión de los pines del microcontrolador y el módulo MPU6050.

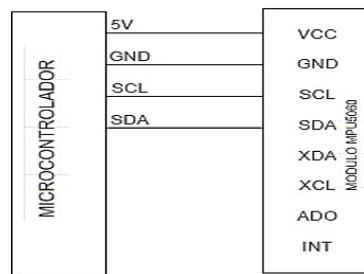


Figura 17 Conexión de pines [66].

3.1.2.2 Encoder double hall switch pulse output

Es un tipo de encoder magnético utilizado para calcular la velocidad y ángulo de rotación en este caso del péndulo, vienen integrados en el sistema de control de motores para el seguimiento preciso de la dinámica [66]. Ver Figura (18).

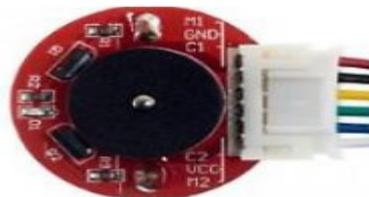


Figura 18 Encoder JGA25-370 [66].

3.1.2.3 Driver IC TBB6612FNG para motor DC

Es un circuito integrado (IC) controlador diseñado específicamente para motores de corriente continua y tiene la capacidad para gestionar eficazmente la dirección y el frenado de un motor DC. Proporciona protección para garantizar un rendimiento fiable y seguro en

diversas aplicaciones, en este caso los motores A y B del robot balancín [66]. Ver Figura (19).

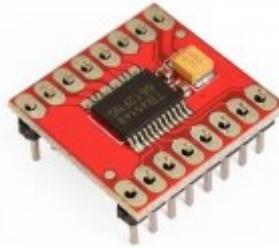


Figura 19 Driver IC TBB6612FNG [66].

3.1.2.4 Motor DC GA37-520

Este motor de corriente continua (Figura 3.6) cuenta con una tensión nominal de 12V DC, exhibe un rango de voltaje operativo que abarca desde 1V hasta 16 V DC, dándole versatilidad a su implementación. Además, el motor presenta una estructura de engranajes con una relación de 1:30, lo que afecta tanto su velocidad de rotación como su torque de salida. El rango de temperatura operativo se extiende desde -10°C hasta $+60^{\circ}\text{C}$, lo que le permite adaptarse a diferentes entornos, y un juego extremo del eje en el intervalo de 0.05 mm a 0.35 mm que asegura precisión en su funcionamiento mecánico. Asimismo, el motor exhibe una velocidad sin carga de $360\text{ rpm} \pm 10\%$ y una corriente de arranque menor o igual a 4^{a} al voltaje nominal [66].

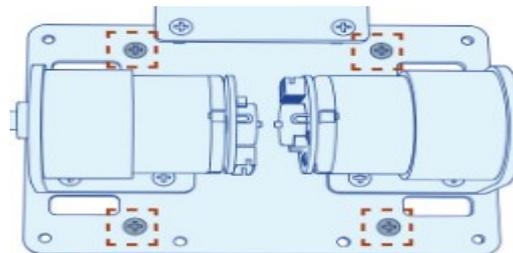


Figura 20 Motor DC GA37-520 [66].

3.1.2.5 Arduino Uno

Es la placa Arduino Uno R3 esencial para la creación de proyectos electrónicos y sistemas interactivos. Sus principales características son: Microcontrolador: ATmega328P. Velocidad de reloj: 16 MHz [43]. El desarrollo y programación del microcontrolador se llevó a cabo utilizando el entorno de desarrollo Arduino IDE, junto con el compilador específico para arquitecturas de 8 bits. La programación del microcontrolador se facilitó a través de la conexión USB directa al puerto de programación del Arduino Uno.

3.1.3 Componentes lógicos (Software)

3.1.3.1 Simulación Numérica

La simulación numérica permite la programación y visualización de un entorno gráfico, cuyos elementos simplifican la caracterización del sistema, generando el espacio de estados que describe el desempeño del control. Su enfoque se centra en la resolución de problemas complejos mediante métodos numéricos y algoritmos avanzados. En este caso se usa programación en MATLAB©-Simulink un lenguaje de programación de alto nivel que permiten la correcta ejecución del algoritmo desarrollado para el control del sistema péndulo invertido [67].

3.1.3.2 Arduino IDE

Arduino IDE es una plataforma integral para programar placas Arduino en proyectos electrónicos y sistemas de control. En el contexto del sistema balancín, facilita la programación del microcontrolador, configuración de pines, lectura de sensores y aplicación de algoritmos de control. Permite mantener el equilibrio del sistema balancín de manera precisa y adaptable, simplificando el proceso de desarrollo y carga de código. En resumen, Arduino IDE es esencial para implementar efectivamente algoritmos de control en sistemas complejos como el balancín [9].

3.1.3.3 Soporte Arduino-Simulink

Esta librería para software numérico establece un nexo técnico crucial entre las placas Arduino y la plataforma de simulación Simulink. Actuando como interfaz, simplifica la representación en bloques de fenómenos físicos y procesos, permitiendo una programación eficiente. En el marco del sistema balancín, esta funcionalidad posibilita el diseño, simulación y ajuste de algoritmos de control en un entorno virtual. Aprovechando las capacidades de Simulink, se pueden analizar y optimizar el rendimiento del sistema antes de su implementación real. Este enfoque respalda la creación de sistemas de control altamente eficientes y precisos mediante una validación rigurosa en simulación antes de su aplicación en el sistema físico [9].

3.2 Diseño de la propuesta

En esta sección se utiliza un enfoque integral que combinan análisis teóricos y experimentales en el desarrollo de la propuesta. Se emplean ecuaciones diferenciales, fórmulas, simulaciones y evaluaciones prácticas para respaldar y validar los resultados obtenidos. Esta combinación de enfoques permite una comprensión profunda del sistema

y garantiza la precisión y confiabilidad de los resultados. Tanto los análisis teóricos como los experimentales serán evaluados para su comparativa y verificar el desempeño crucial en la evaluación y optimización de la propuesta, mediante una base sólida en el diseño y desarrollo del proyecto.

3.2.1 Diseño Electrónico del Robot Balancín

En la figura (21) se muestra el esquema electrónico del robot balancín que se usa como planta de prueba para efectuar el funcionamiento del control. El esquema consta de una placa Arduino Uno que funciona con responsabilidad de procesar la información de los sensores y enviar los comandos al controlador del motor. También consta el MPU, cabe indicar que sus siglas significan Unidad de Medición Inercial y su función está en detectar y medir la inclinación del robot, fundamental para el sistema de equilibrio. El controlador de motor TB6612FNG es el que gestiona los dos motores controlando la velocidad y dirección para que mantenga el equilibrio. El Módulo Bluetooth HC05 permite la comunicación inalámbrica el cual nos sirve para obtener los datos precisos de los sensores [66].

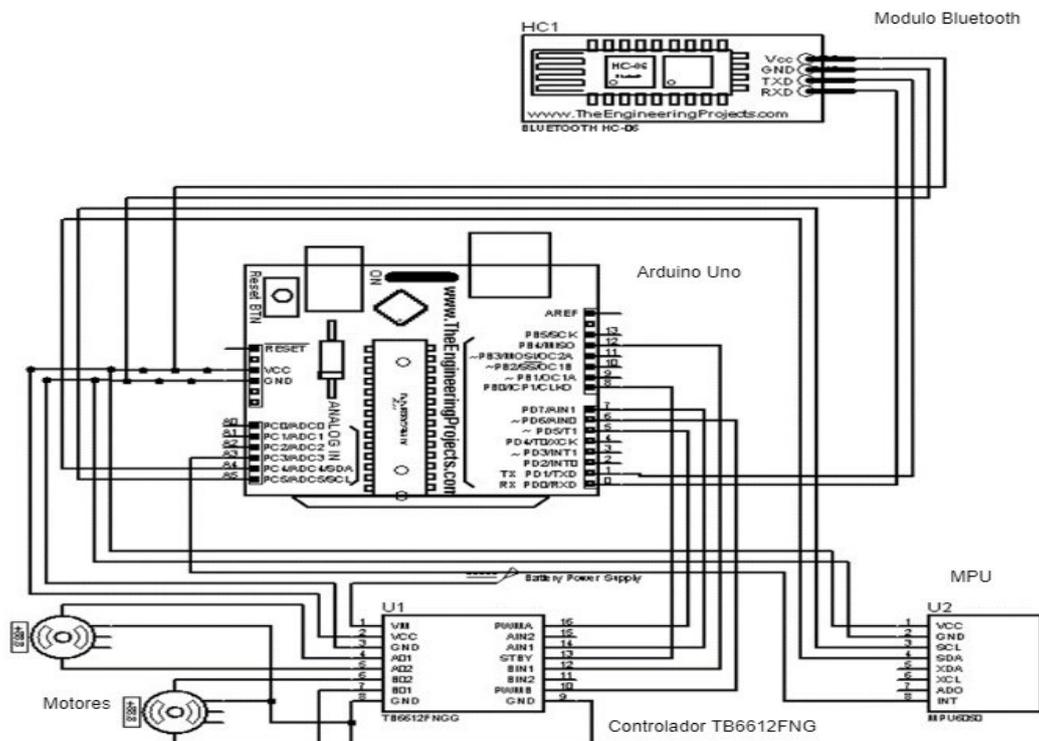


Figura 21 Diseño electrónico del robot balancín.

Realizado por: Yuri Torres, 2024.

3.2.2 Simulador Numérico

Para llevar a cabo las simulaciones de manera adecuada, es esencial contar con datos reales que se obtienen a través de la identificación del sistema. Se utiliza el modelo en el espacio de estados para derivar la ecuación diferencial del sistema. Esto se logra mediante el uso de System Identification de MATLAB© Ver Figura (22), herramienta que permite expresar un modelo específico utilizando los datos tanto de entrada como de salida [67].

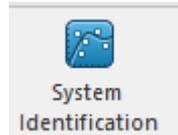


Figura 22 Bloque MATLAB© de System Identification [67].

3.2.2.1 Análisis de la identificación de la planta

Se realizan varias evaluaciones de ejecución para el mejor modelo de identificación en tiempo real como se muestra en la Figura (24), cabe recalcar que no hay uso de un sistema controlado. Por tanto, se realiza una entrada de su máxima velocidad del motor de -255 a 255 de U (PWM) lo que representa su ancho de pulso. Mediante este proceso se identifica el ángulo de inclinación del péndulo en sus distintos valores angulares como salida Y (THETA), tal como indica el literal a) de la Figura (23). La identificación del ángulo de inclinación es de un intervalo de 0.005 segundos, lo que representa 551 muestras como proceso de identificación de la planta como indica el literal b) de la Figura (23).

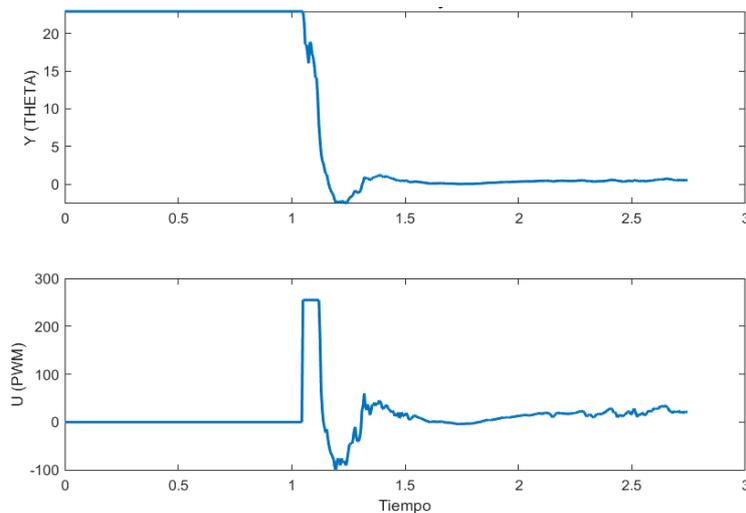


Figura 23 a) Señal de salida b) Señal de entrada.

Elaborado por: Yuri Torre, 2024.

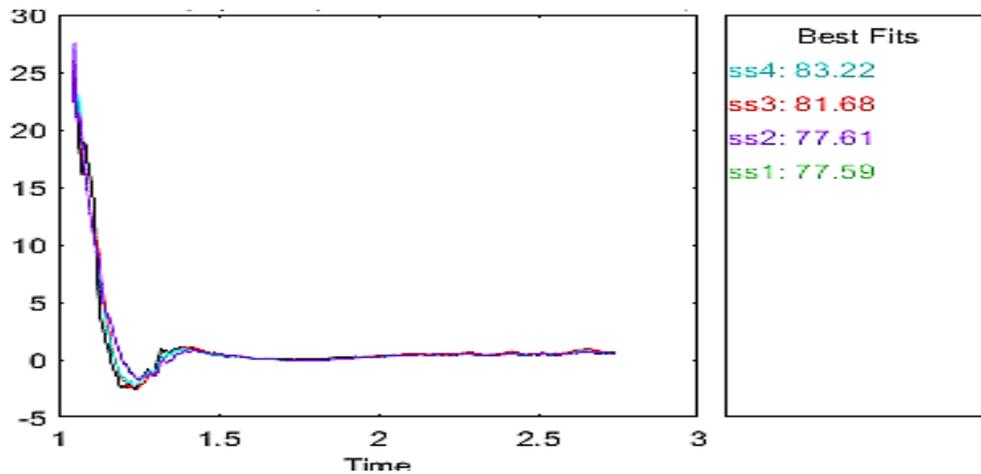


Figura 24 Mejores estimaciones del sistema.

Elaborado por: Yuri Torre, 2024.

En la Tabla (2), se exhiben las aproximaciones obtenidas mediante la formulación en espacio de estados del identificador de sistemas MATLAB®. Estas diferentes perspectivas han sido concebidas a partir de un análisis exhaustivo de las características y propiedades del sistema. Dichas recopilaciones se comparan con el propósito de identificar y seleccionar la formulación que se ajuste mejor a la representación matemática precisa del sistema. Este proceso es esencial para la implementación efectiva de las teorías de control pertinentes, permitiendo la concepción y desarrollo óptimos de estrategias de control que se adecuen a la dinámica y comportamiento del sistema real.

Tabla 2 Identificación de la planta

Modelo	Estimación	Tipo	Aproximación
SS1	Modelo espacio de estado	Orden 4 (Tiempo Continuo)	77.59%
SS2	Modelo espacio de estado	Orden 4 (Tiempo Continuo)	77.61%
SS3	Modelo espacio de estado	Orden 4 (Tiempo Continuo)	81.68%
SS4	Modelo espacio de estado	Orden 4 (Tiempo Discreto)	83.22%

Elaborado por: Yuri Torres, 2024.

Según los datos mejor evaluados, se considera el SS4 con un porcentaje de 83.22%, ideal para una estimación de las matrices a continuación:

A=

	X1	X2	X3	X4
X1	-42.25	-26.02	-48.26	-29.29
X2	3.949	-10.08	-296.5	-144
X3	10.16	151.7	-166.9	-350.4
X4	16.15	194.44	288.6	0.4512

B=

	U (PWM)
X1	0.8411
X2	2.909
X3	-1.172
X4	-3.8

C=

	X1	X2	X3	X4
Y (THETA)	5.099	-0.1551	0.1173	0.03161

D=

	U (PWM)
Y (THETA)	0

Entonces la matriz real de la planta queda de la siguiente manera:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} -42.25 & -26.02 & -48.26 & -29.29 \\ 3.949 & -10.08 & -296.5 & -144 \\ 10.16 & 151.7 & -166.9 & -350.4 \\ 16.15 & 194.44 & 288.6 & 0.4512 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0.8411 & 0.8411 \\ 2.909 & 2.909 \\ -1.172 & -1.172 \\ -3.8 & -3.8 \end{bmatrix} u \quad (82)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = [5.099 \quad -0.1551 \quad 0.1173 \quad 0.03161] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} u \quad (83)$$

➤ Energía cinética de la rueda

Se analiza la masa de la rueda y el cambio de la posición al desplazamiento. Se considera el movimiento angular (φ), dado por la velocidad tangencial de la rueda en movimiento circular (v) y el desplazamiento angular (r) que ha recorrido como se muestra en la ecuación (84).

➤ Energía cinética del péndulo

Se detalla en la ecuación (85), sus componentes resultantes de magnitud y velocidad de la masa puntual mediante elementos espaciales y vectoriales. También se analiza el movimiento rotatorio del mismo.

$$E_{C1} = \frac{1}{2} m_r \dot{x}^2 \quad (84)$$

$$E_{C2} = \frac{1}{2} m_p \dot{x}^2 + m_p \dot{x} \dot{\theta} l \cos(\theta) + \frac{1}{2} m_p \dot{\theta}^2 l^2 \quad (85)$$

Se analiza la energía potencial del sistema partiendo de la ecuación (2): se obtiene la energía potencial en el péndulo considerando nula la energía potencial del carro.

$$E_p = m_p g l \cos(\theta) \quad (86)$$

3.2.2.2 Lagrangiano

Entonces el proceso de Lagrangiano, queda expresado de la siguiente forma: $L = E_{C1} + E_{C2} - E_p$.

$$L = \frac{1}{2} m_r \dot{x}^2 + \frac{1}{2} m_p \dot{x}^2 + m_p \dot{x} \dot{\theta} l \cos(\theta) + \frac{1}{2} m_p \dot{\theta}^2 l^2 - m_p g l \cos(\theta) \quad (87)$$

3.2.2.3 Mecánica de Lagrange

En la Figura (26), se analizan las fuerzas en las ruedas a través del torque generado por el motor Γ_m . Para ello, se emplea la ley de voltaje de Kirchhoff que realiza el análisis de la corriente, teniendo en cuenta la inductancia (L) del motor, la cual suele ser pequeña y se aproxima a cero como se indica en ecuación (88).

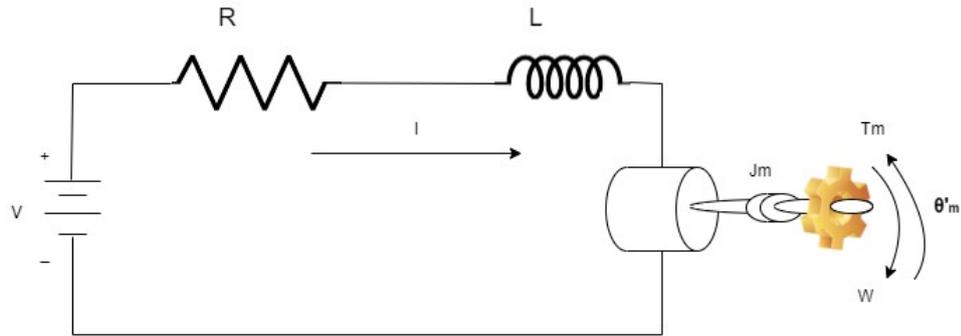


Figura 26 Diagrama del motor DC.

Realizado por: Yuri Torres, 2024.

$$L \approx 0H \quad (88)$$

Se asume que la inductancia L es aproximadamente cero ($L \approx 0H$) y se procede a tomar este valor y reemplazarlo en la ley de voltaje de Kirchhoff en la ecuación (89) para el circuito eléctrico asociado al sistema y queda reducido a la expresión (90).

$$0 = -V + R \cdot i + L \frac{di}{dt} + E_a(t) \quad (89)$$

$$0 = -V + R \cdot i + KE \dot{\theta}(t) \quad (90)$$

Al considerar que L es despreciable, la expresión simplificada de la ley de voltaje nos permite obtener una representación más manejable del comportamiento eléctrico del sistema como se muestra en (91).

$$i = \frac{V}{R} - \frac{KE}{R} \dot{\theta}(t) \quad (91)$$

Se relaciona el voltaje interno generado del motor DC con las constantes respectivas: donde E_a , es el voltaje interno generado por el motor; KE es una constante eléctrica; K_t es la constante de torque; y $\omega(t)$ es la velocidad angular en el instante del tiempo como muestra la expresión (92); luego sustituimos por el método de notación (93) para llegar a la ecuación principal del Torque del motor.

➤ **Relación del voltaje interno generado con la constante eléctrica**

$$E_a(t) = KE \omega(t) \quad (92)$$

$$E_a(t) = K_e \dot{\theta}(t) \quad (93)$$

La ecuación de Lorentz establece que el torque de un motor DC (Γ_m), es proporcional a la corriente que fluye a través de él (i), relacionando así el torque del motor con la constante de torque como define (K_t) como expresa la ecuación (94).

$$\Gamma_m(t) = K_t i(t) \quad (94)$$

En la ecuación (95), se reemplaza la corriente del motor de la ecuación (83) y se obtiene la ecuación de torque del motor (96).

$$\Gamma_m(t) = \frac{V K_t}{R} - \frac{KE K_t \dot{\theta}_m}{R} \quad (95)$$

$$V = K_s u$$

$$\Gamma_m(t) = \frac{K_t K_s}{R} u - \frac{KE K_t \dot{\theta}_m}{R} \quad (96)$$

3.2.2.4 Fuerzas Generalizadas (F)

En este caso se utiliza fuerza del torque, fricción y el amortiguamiento viscoso. Estas fuerzas se pueden expresar en términos de velocidad lineal. Ver expresiones (97) y (98), luego se llevan a términos de desplazamiento x , como indica la fórmula (99).

$$v = r \omega \quad (97)$$

$$v = r \cdot \dot{\varphi} \quad (98)$$

$$\frac{v}{r} = \dot{\varphi} \quad , \quad \varphi = \frac{x}{r} \quad (99)$$

Para obtener la ecuación del torque del motor, se analiza la fuerza de torque tanto en el péndulo como en la rueda. Esto resulta en una diferencia de torque que se plantea en relación con el trabajo realizado (w) en la expresión (100), el cual depende de las fuerzas involucradas (F) por la distancia correspondiente (d).

➤ Trabajo en el torque del motor

$$w = F \cdot d \quad (100)$$

Considerando $\left(\frac{x}{r}\right)$ posición de la rueda y (θ) posición del péndulo, se forma la ecuación trabajo del motor con las diferencias de torques.

$$w_{\Gamma_m} = \Gamma_m \frac{x}{r} - \Gamma_m \theta \quad (101)$$

Se deriva parcialmente con respecto a las dos variables x y θ ; debido a que son muy necesarias al momento de llevarlas a la ecuación Lagrangiano.

Derivada con respecto a 'x' en la expresión (102).

$$\frac{\partial w_{\Gamma_m}}{\partial x} = \frac{\Gamma_m}{r} \quad (102)$$

Luego se derivada con respecto a ‘ θ ’ y queda de la siguiente forma:

$$\frac{\partial w_{\Gamma m}}{\partial \theta} = -\Gamma_m \quad (103)$$

➤ **Trabajo de la fuerza de fricción con el suelo**

Para hallar el trabajo de fricción que existe en el suelo, se consideran las fuerzas en el suelo sobre el eje x.

$$w_{Fs} = F \cdot x \quad (104)$$

La fuerza de fricción estática en el suelo está dada por el coeficiente de fricción del suelo u_s , por la derivada de la posición \dot{x} como indica a continuación:

$$F = u_s \dot{x} \quad (105)$$

En la ecuación (104), se reemplaza con (105), lo que nos permite obtener la ecuación general de la fuerza de fricción con el suelo con su respectiva oposición. Ver ecuación (106).

$$w_{Fs} = -u_s \dot{x} x \quad (106)$$

De la expresión (106), se realiza derivada parcial con respecto a las variables x y θ , términos necesario para la ecuación Euler Lagrange.

$$\frac{\partial w_{Fs}}{\partial x} (-u_s \dot{x} x) = -u_s \dot{x} \quad (107)$$

$$\frac{\partial w_{Fs}}{\partial \theta} (-u_s \dot{x} x) = 0 \quad (108)$$

Se analiza la fuerza de viscosidad, esta se genera debido a la fricción viscosa que actúan en dirección opuesta al movimiento de la rueda.

Por tanto, la ecuación está conformada por: F_v , u_e y θ como se indica en la tabla 1.

$$F_v = u_e \cdot \theta \quad (109)$$

Dado la expresión de fuerza viscosa, reemplazamos por la diferencia de sus componentes de θ .

$$F_v = u_s \cdot (\dot{\phi} - \dot{\theta}) \quad (110)$$

➤ **Trabajo de la fuerza viscosa.**

Se analiza la fuerza de viscosidad, ésta se genera debido a la fricción viscosa que actúan en dirección opuesta al movimiento de la rueda.

Por tanto, la ecuación está conformada por:

$$w_{F_v} = -u_e (\dot{\varphi} - \dot{\theta})(\varphi - \theta) \quad (111)$$

De la ecuación (110) se deriva con respecto a las variables del sistema x y θ : conociendo $\varphi = \frac{x}{r}$ y se obtiene las relaciones matemáticas (112) y (113).

$$\frac{\partial F_v}{\partial x} = -\frac{u_e}{r} (\dot{\varphi} - \dot{\theta}) \quad (112)$$

$$\frac{\partial F_v}{\partial \theta} = u_e (\dot{\varphi} - \dot{\theta}) \quad (113)$$

3.2.2.5 Euler Lagrange

Para las coordenadas generalizadas de la expresión (12), se acoplan las derivada de las fuerzas con respecto a 'x'; ecuación (102), (107) y (112) resultando la forma expresada en (114).

$$\frac{\Gamma_m}{r} - \dot{x} \left(u_s + \frac{u_e}{r^2} \right) + \frac{u_e}{r} \dot{\theta} \quad (114)$$

Para el siguiente proceso se considera las derivadas con respecto a ' θ '; ecuación (103) y (113), componentes de Euler Lagrange, formando la ecuación (115)

$$-\Gamma_m + u_e \left(\frac{\dot{x}}{r} - \dot{\theta} \right) \quad (115)$$

Se efectúa la derivación parcial de la expresión (87) respecto a las variables $x, \dot{x}, \theta, \dot{\theta}$, considerando el contexto del sistema dinámico y se obtiene la formulación (116), (117), (118) y (119)

$$\frac{\partial L}{\partial x} = 0 \quad (116)$$

$$\frac{\partial L}{\partial \dot{x}} = (m_r + m_p) \dot{x} + m_p \dot{\theta} l \cos(\theta)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) = (m_r + m_p) \ddot{x} + m_p \ddot{\theta} l \cos(\theta) - m_p l \sin(\theta) (\dot{\theta})^2 \quad (117)$$

$$\frac{\partial L}{\partial \theta} = -m_p l \dot{\theta} \dot{x} \sin(\theta) + m_p g l \sin(\theta) \quad (118)$$

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

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) = m l \ddot{x} \cos(\theta) - m_p l \dot{x} \sin(\theta) \dot{\theta} + m_p l^2 (\ddot{\theta}) \quad (119)$$

Una vez derivado la primera coordenada generalizada (\dot{x}) del Lagrangiano en la expresión (117), se reemplaza con la ecuación (12) igualando a las fuerzas generalizadas (F) menos el coeficiente de fricción del péndulo por la velocidad.

$$(m_r + m_p) m_p \ddot{\theta} l \cos(\theta) - m_p l \sin(\theta) (\dot{\theta})^2 = F - u e \cdot \dot{x} \quad (120)$$

Se sustituye la ecuación (118) y (119) en la ecuación (12) con las representaciones de la segunda coordenada generalizada en función (θ) igual a fricción de péndulo (us) por la variación del ángulo obteniendo las siguientes expresiones:

$$m_p l \ddot{x} \cos(\theta) - m_p l \dot{x} \sin(\theta) \dot{\theta} + m_p l^2 (\ddot{\theta}) + m_p l \dot{\theta} \dot{x} \sin(\theta) - m_p g l \sin(\theta) = -us \cdot \dot{\theta} \quad (121)$$

$$m_p l \ddot{x} \cos(\theta) + m_p l^2 (\ddot{\theta}) - m_p g l \sin(\theta) + us \cdot \dot{\theta} = 0 \quad (122)$$

Las ecuaciones que describen el sistema dinámico son (120) y (122):

$$(m_r + m_p) \ddot{x} + m_p \ddot{\theta} l \cos(\theta) - m_p l \sin(\theta) (\dot{\theta})^2 = F - u e \cdot \dot{x}$$

$$m_p l \ddot{x} \cos(\theta) + m_p l^2 (\ddot{\theta}) - m_p g l \sin(\theta) + us \cdot \dot{\theta} = 0$$

Se despeja \ddot{x} y $\ddot{\theta}$ de la ecuación (120) y (122) para optimizar el proceso.

$$\left\{ \begin{array}{l} \ddot{x} = -\frac{m_p l \cos(\theta)}{(m_r + m_p)} \ddot{\theta} - \frac{m_p l \sin(\theta)}{(m_r + m_p)} \dot{\theta}^2 + \frac{1}{(m_r + m_p)} F - \frac{u e}{(m_r + m_p)} \dot{x} \\ \ddot{\theta} = \frac{\cos(\theta)}{l} \ddot{x} - \frac{g \sin(\theta)}{l} + \frac{us}{m_p l^2} \dot{\theta} \end{array} \right. \quad (123)$$

$$\left\{ \begin{array}{l} \ddot{x} = -\frac{m_p l \cos(\theta)}{(m_r + m_p)} \ddot{\theta} - \frac{m_p l \sin(\theta)}{(m_r + m_p)} \dot{\theta}^2 + \frac{1}{(m_r + m_p)} F - \frac{u e}{(m_r + m_p)} \dot{x} \\ \ddot{\theta} = \frac{\cos(\theta)}{l} \ddot{x} - \frac{g \sin(\theta)}{l} + \frac{us}{m_p l^2} \dot{\theta} \end{array} \right. \quad (124)$$

En la ecuación (124), la variable de la expresión (123) es sustituida para facilitar el proceso dinámico del sistema y se obtiene la ecuación (125). Ver Anexo (4)

$$\ddot{x} = -\frac{m_p l \cos(\theta)}{(m_r + m_p)} \ddot{\theta} - \frac{m_p l \sin(\theta)}{(m_r + m_p)} \dot{\theta}^2 + \frac{1}{(m_r + m_p)} F - \frac{u e}{(m_r + m_p)} \dot{x} \quad (125)$$

$$\ddot{\theta} = -\frac{m_p \operatorname{sen}(\theta) \cos(\theta)}{(m_r + m_p) + m_p \cos(\theta)^2} + \frac{\cos(\theta)}{l((m_r + m_p) + m_p \cos(\theta)^2)} F \quad (125)$$

$$- \frac{ue \cos(\theta)}{l((m_r + m_p) + m_p \cos(\theta)^2)} \dot{x} - \frac{g \operatorname{sen}(\theta)(m_r + m_p)}{((m_r + m_p) + m_p \cos(\theta)^2)}$$

$$+ \frac{us(m_r + m_p)}{m_p l^2 ((m_r + m_p) + m_p \cos(\theta)^2)} \dot{\theta}$$

3.3 Linealización por espacios de estados

En este paso, se lleva a cabo la derivación parcial de las ecuaciones que describen la dinámica del sistema (123) y (125), lo que implica la determinación del jacobiano. Pero antes se expresan en variables de estados que se detallan en la Tabla (3). Considerando la identificación en la Tabla (21). Ver ecuaciones en términos de estados con aproximaciones del ángulo $\operatorname{sen}(\theta) \approx \theta$, $\cos(\theta) \approx 1$ y $\theta \approx 0$. Expresiones (127) y (129).

Tabla 3 Simbologías de variables de estado

$x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 0, F = u$		
$x_1 = x$	$\dot{x}_1 = \dot{x} = x_2$	$x_1 = 0$
$x_2 = \dot{x}$	$\dot{x}_2 = \ddot{x}$	$x_2 = 0$
$x_3 = \theta$	$\dot{x}_3 = \dot{\theta}$	$x_3 = 0$
$x_4 = \dot{\theta}$	$\dot{x}_4 = \ddot{\theta}$	$x_4 = 0, u = 0$

Realizado por: Yuri Torres, 2024.

$$\dot{x}_1 = x_2 \quad (126)$$

$$\dot{x}_2 = -\frac{m_p l}{(m_r + m_p)} x_4 + \frac{u}{(m_r + m_p)} - \frac{ue}{(m_r + m_p)} x_2 \quad (127)$$

$$\dot{x}_3 = x_4 \quad (128)$$

$$\dot{x}_4 = \frac{ue}{m_r l} x_2 - \frac{(m_r + m_p)us}{m_r m_p l^2} x_4 + \frac{(m_r + m_p)g}{m_r l} x_3 - \frac{u}{m_r l} \quad (129)$$

Se procede hallar las matrices Jacobinas y mediante el análisis anterior de los puntos de equilibrio. Ver desarrollo en Anexo (5).

$$A = \begin{bmatrix} 0 & \frac{1}{ue} & 0 & \frac{0}{m_p l} \\ 0 & -\frac{1}{(m_r + m_p)} & 0 & -\frac{0}{(m_r + m_p)} \\ 0 & \frac{0}{m_r l} & \frac{0}{(m_r + m_p)g} & \frac{1}{(m_r + m_p)us} \\ 0 & \frac{ue}{m_r l} & \frac{(m_r + m_p)g}{m_r l} & \frac{(m_r + m_p)us}{m_r m_p l^2} \end{bmatrix} \quad (130)$$

$$B = \begin{bmatrix} 0 \\ 1 \\ \frac{0}{(m_r + m_p)} \\ 0 \\ \frac{1}{-m_r l} \end{bmatrix} \quad (131)$$

Las matrices derivadas del análisis de jacobiano y los puntos de equilibrio se pueden encontrar en la sección de resolución de anexos.

Con las matrices jacobianas linealizada en sus puntos de equilibrio construimos el modelo en espacio de estados (132) y en la salida tenemos (133), cabe recalcar que u es el vector de entrada que representa la fuerzas aplicadas y la matriz D presenta valor de cero.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{ue} & 0 & \frac{0}{m_p l} \\ 0 & -\frac{1}{(m_r + m_p)} & 0 & -\frac{0}{(m_r + m_p)} \\ 0 & \frac{0}{m_r l} & \frac{0}{(m_r + m_p)g} & \frac{1}{(m_r + m_p)us} \\ 0 & \frac{ue}{m_r l} & \frac{(m_r + m_p)g}{m_r l} & \frac{(m_r + m_p)us}{m_r m_p l^2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ -\frac{1}{m_r l} \end{bmatrix} u \quad (132)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \quad (133)$$

3.4 Entornos para el diseño del control

Primero se verifica la dinámica del sistema, luego se determina si es controlable u observable en el software MATLAB© - Simulink. Posteriormente, se discretiza con un período de muestreo adecuado para obtener el modelo discreto en espacio de estados. Una vez resuelto, se generan gráficos que mostrarán el comportamiento de la planta.

El modelo matemático linealizado en espacios de estados se compone mediante:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (134)$$

$$y(t) = Cx(t) + Du(t) \quad (135)$$

Se hace un llamado a las matrices A, B, C y D de las expresiones: (130), (131) y (133) a conformar el modelo matemático.

$$A = \begin{bmatrix} 0 & \frac{1}{ue} & 0 & \frac{0}{m_p l} \\ 0 & -\frac{1}{(m_r + m_p)} & 0 & -\frac{0}{(m_r + m_p)} \\ 0 & \frac{0}{m_r l} & \frac{(m_r + m_p)g}{m_r l} & \frac{1}{(m_r + m_p)} \\ 0 & \frac{ue}{m_r l} & \frac{(m_r + m_p)us}{m_r m_p l^2} & \frac{1}{m_r m_p l^2} \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \\ (m_r + m_p) \\ 0 \\ -\frac{1}{m_r l} \end{bmatrix} \quad (136)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (137)$$

Mediante el software MATLAB© se elabora el código con los parámetros y especificaciones para ver las matrices en espacio de estados con los valores respectivos en sistema continuo (**sys**) como indican las siguientes expresiones.

Donde:

model_2

A=

	x1	x2	x3	x4
x1	0	1	0	0
x2	0	-0.2966	0	-0.58
x3	0	0	0	1
x4	0	4.236	140.1	0.5259

B=

	u1
x1	0
x2	1.187
x3	0
x4	-16.94

C=

	x1	x2	x3	x4
y1	1	0	0	0
y2	0	0	1	0

D=

	u1
y1	0
y2	0

Una vez realizado el análisis de variables de estados en tiempo continuo del sistema dinámico, se procede a la discretización del modelado (*sys_d*) con un periodo de muestreo de $T=0.1s$ mediante el Mapeo Bilineal en relación con variables continuas y discretas.

Model_2d

A=

	x1	x2	x3	x4
x1	1	0.09759	-0.03179	-0.004539
x2	0	0.9518	-0.6358	-0.09078
x3	0	0.03315	2.113	0.1588
x4	0	0.663	22.25	2.177

B=

	u1
x1	0.009635
x2	0.1927
x3	-0.1326
x4	-2.652

C=

	x1	x2	x3	x4
y1	1	0.0488	-0.0159	-0.002269
y2	0	0.01658	1.556	0.07942

D=

	u1
y1	0.004817
y2	-0.0663

Una vez completado los pasos fundamentales de modelo en tiempo continuo y discreto en espacio de estados, se procede al análisis de estabilidad y comportamiento dinámico del sistema.

3.4.1 Evaluación de la estabilidad del sistema

Mediante análisis de Rudolf E. Kalman, se define ' a, b, c, d ' coeficientes calculados a partir de los elementos de la matriz "A" de la ecuación (32) y (33) para obtener los autovalores propios de la matriz. El proceso de desarrolla en el software de simulación MATLAB© que permite ver al menos un valor propio con parte real positiva siendo un motivo suficiente para determinar una INESTABILIDAD.

Eig(A)

$$\det (S I - A) = \begin{bmatrix} 0 \\ 11.9999 \\ -0.3019 \\ -11.4688 \end{bmatrix} \quad (138)$$

3.4.2 Respuesta ante el impulso del sistema

En la Figura (21) muestra el comportamiento del sistema completamente divergente demostrando inestabilidad, y también como reacciona el péndulo ante los impulsos y condiciones iniciales:

$$\begin{bmatrix} 0 \\ 0 \\ pi/18 \\ 0 \end{bmatrix} \quad (139)$$

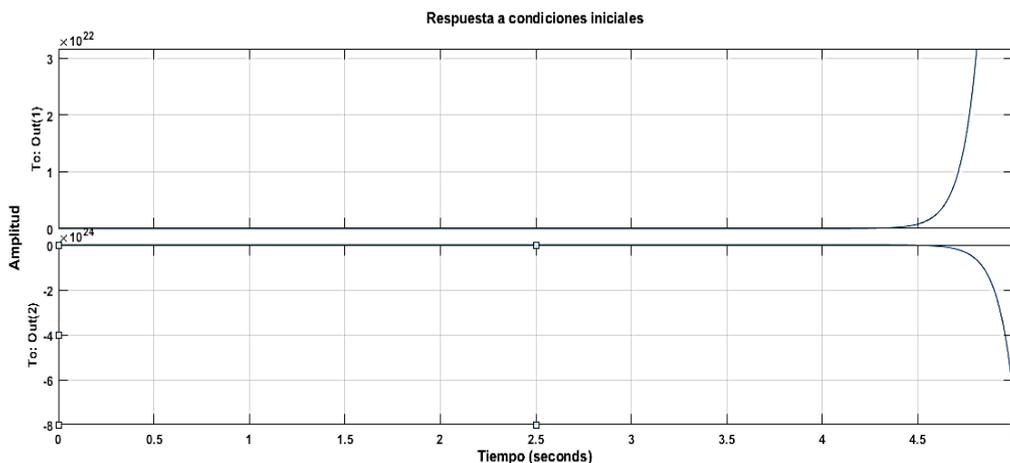


Figura 27 Respuesta al impulso en tiempo continuo.

Realizado por: Yuri Torres, 2024.

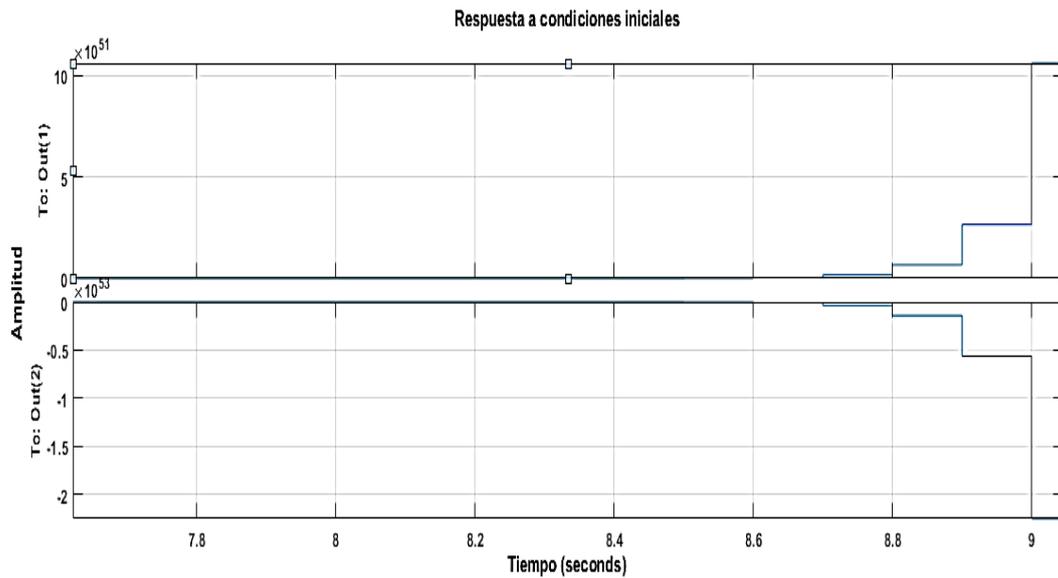


Figura 28 Respuesta al impulso en tiempo discreto.

Realizado por: Yuri Torres, 2024.

3.4.3 Respuesta al escalón unitario

La Figura (29), muestra la respuesta del sistema a un escalón unitario para la primera salida. Es decir, cómo el sistema responde en cuanto a la robustez cuando se le aplica una entrada de referencia. Se puede observar cómo el sistema se comporta en términos de estabilidad y cómo alcanza su estado estacionario.

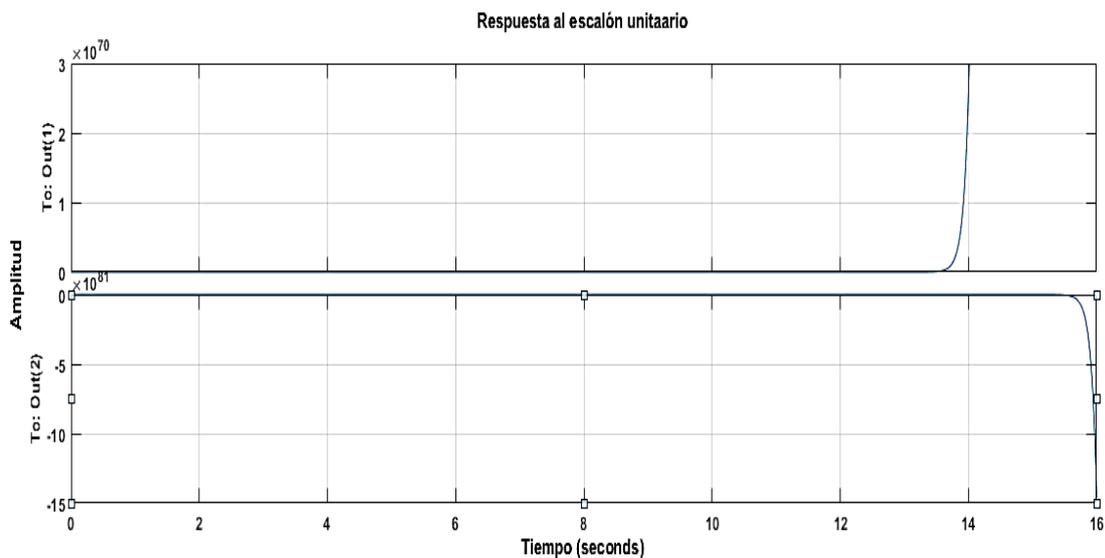


Figura 29 Respuesta al escalón unitario.

Realizado por: Yuri Torres, 2024.

3.4.4 Análisis de controlabilidad

Mediante análisis de evaluaciones anteriores se trabaja con el modelo de espacios de estados en tiempo continuo y se procede a la comprobación de la controlabilidad; haciendo uso de la ecuación (34), que verifica si el rango de $P =$ igual a n y verificando que el sistema es completamente CONTROLABLE, demostrado con los datos del simulador MATLAB© que se presentan a continuación:

P=ctrb (model_2)

P=

1.0e+03*

$$\begin{bmatrix} 0 & 0.0012 & 0.0095 & -0.0006 \\ 0.0012 & 0.0095 & -0.0006 & 1.3547 \\ 0 & -0.0169 & -0.0039 & -2.3354 \\ -0.0169 & -0.0039 & -2.3354 & -1.7745 \end{bmatrix} \quad (140)$$

Rank(P)

Ans = 4

3.4.5 Análisis de observabilidad

De la misma forma se establece la matriz de observabilidad O de la ecuación (35) y se comprueba que es completamente OBSERVABLE porque el rango de sus valores es igual al número de las variables de estados, entonces se puede decir que todas las variables de estado pueden ser estimadas a partir de las salidas del sistema. A continuación, la matriz de observabilidad comprobada en el software MATLAB©.

O=obsv (model_2)

O=

$$\begin{bmatrix} 1.000 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 1.0000 \\ 0 & -0.2966 & 0 & -0.5800 \\ 0 & 4.2359 & 140.0859 & 0.5800 \\ 0 & -2.3687 & -81.2458 & -0.1329 \\ 0 & 0.9710 & 73.6650 & 137.9057 \end{bmatrix} \quad (141)$$

Rank= 4

3.5 Diseño del servo-controlador (LQG)

Antes del diseño del control Lineal Cuadrático Gaussiano, es necesario modelar con un seguimiento de referencia que ingrese al sistema y permita seguir la posición de la planta $x(t)$ y mediante la acción de control gestione la posición del ángulo $\theta(t)$ como indican las expresiones (43) y (44). Se caracteriza por las matrices extendidas $[A_a, B_a, C_a]$ que diseñan el control, las ecuaciones (142) y (143) definen el sistema, donde \dot{e} , representa el error referido entre la salida real y la salida de referencia:

$$\dot{e} = A_a e + B u_e \quad (142)$$

$$y = C_a e \quad (143)$$

Para encontrar el vector de retroalimentación K_a , es necesario analizar ecuaciones de un servo-sistema como indica la expresión (42) para el vector de realimentación extendida.

$$K_a = -[K - k_1] \quad (144)$$

Las matrices extendidas mediante el software MATLAB© son:

A_amp

	x1	x2	x3	x4	x5
x1	0	1.0000	0	0	0
x2	0	-0.2966	0	-0.5800	0
x3	0	0	0	1.0000	0
x4	0	4.2359	140.0859	0.5259	0
x5	-1.0000	0	0	0	0

B_amp=

	x1
x1	0
x2	1.1865
x3	0
x4	-16.9434
x5	0

C_amp=

	x1	x2	x3	x4	x5
y1	1	0	0	0	0

D_amp=

	u1
y1	0

Las matrices extendidas, son esenciales para generar el modelo en espacios de estados extendido que son generadas en el mismo software.

>> sys_a=ss (A_amp, B_amp, C_amp, D_amp)

A

	x1	x2	x3	x4	x5
x1	0	1	0	0	0
x2	0	-0.2966	0	-0.58	0
x3	0	0	0	1	0
x4	0	4.236	140.1	0.5259	0
x5	-1	0	0	0	0

B=

	x1
x1	0
x2	1.187
x3	0
x4	-16.94
x5	0

C=

	x1	x2	x3	x4	x5
y1	1	0	0	0	0

D=

	u1
y1	0

Luego del análisis anterior, se comprueba la controlabilidad del sistema y se observa que el rango de la matriz P que cumple con el número de variables y se confirma que el sistema es completamente controlable.

>> P= [Aa Ba; -Ca 0]

rangP=rank(P)

P=

0	1.0000	0	0	0	0
0	-0.2966	0	-0.5800	0	1.1865
0	0	0	1.0000	0	0
0	4.2359	140.0859	0.5259	0	-16.9434
-1.0000	0	0	0	0	0
-1.0000	0	0	0	0	0

Rank(P) =

5

3.5.1 Control LQR sin ruido ni perturbación

El objetivo principal es encontrar la ley de control $u(t)$ que minimice J . Este indicador se suele definir como una mezcla ponderada del error cuadrático entre el estado deseado y el estado actual del sistema, junto con el esfuerzo aplicado al control. De aquí depende la existencia de la retroalimentación de estados, es decir, la ley de control óptima $u(t) = -K(t) x(t)$.

La ecuación (47) minimiza la función de costo cuadrática.

$$J = \int_0^{\infty} (x^T Q_x + u^T R_u) dt \quad (145)$$

Mediante la dinámica

$$\dot{x} = Ax + Bu \quad (146)$$

$$u(0) = x_0 \quad (147)$$

Entonces las matrices ponderadas en este desarrollo están dadas por Q y R las cuales deben ser mayor o igual a cero para la respectiva energía de control. Una viable solución a este asunto es la optimización de la ley de control como explica el desarrollo de la ecuación (54).

$$K = -R^{-1} B.P \quad (148)$$

Donde P se encuentra a través de la ecuación matemática (53) de Riccati, igual a cero.

$$A^T P(t) + P A + Q - P B R^{-1} B^T P = 0 \quad (149)$$

Mediante la optimización del proceso anterior, se realiza las ponderaciones respectivas de Q y R siendo la parte mas fundamental del caso. A continuación, las matrices elegidas:

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 50 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 20 \end{bmatrix} \quad (150)$$

$$R = [1] \quad (151)$$

Para la simulación y respuesta a la matriz de retroalimentación óptima de estados (K) se hace uso del software MATLAB© con los siguientes resultados:

$Q_lqr =$

10.0000	0	0	0	0
0	3.0000	0	0	0
0	0	50.0000	0	0
0	0	0	3.0000	0
0	0	0	0	20.0000

$k_lqi =$

-8.0879	-6.4454	-32.0464	-2.9784	4.4721
---------	---------	----------	---------	--------

$S =$

39.2678	17.6605	32.4068	1.7141	-27.7068
17.6605	11.5030	22.7860	1.1859	-10.8400
32.4068	22.7860	70.5687	3.4870	-19.0686
1.7141	1.1859	3.4870	0.2588	-1.0231
-27.7068	-10.8400	-19.0686	-1.0231	36.1701

$e =$

-1.2075	+1.1508i
-1.2075	-1.1508i
-1.4955	+0.0000i
-5.3622	+0.0000i
-33.3140	+0.0000i

Para el vector de control óptimo se obtiene:

$$k_{lqr} = \begin{bmatrix} -8.0879 & -6.4454 & -32.0464 & -2.9784 \end{bmatrix}$$

$$k_i = \begin{bmatrix} 4.4721 \end{bmatrix}$$

También se obtuvo el vector de control de realimentación K_a para el sistema extendido.

$$K_a = [-8.0879 \quad -6.4454 \quad -32.0464 \quad -2.9784 \quad 4.4721] \quad (152)$$

En base a estos desarrollos se logra valores respectivos para K y K_l .

$$K = [-8.0879 \quad -6.4454 \quad -32.0464 \quad -2.9784 \quad 4.4721] \quad (153)$$

$$K_l = [4.4721] \quad (154)$$

Los valores de la matrices extendidas obtienen sus autovalores mediante cálculos simulados y se evidencian a continuación:

$sys_r =$

A=

	x1	x2	x3	x4	x5
x1	0	1	0	0	0
x2	9.596	7.351	38.02	2.954	-5.306
x3	0	0	0	1	0
x4	-137	-105	-402.9	-49.94	75.77
x5	-1	0	0	0	0

B=

	u1
x1	0
x2	10187
x3	0
x4	-16.94
x5	0

C=

	x1	x2	x3	x4

y1	1	0	0	0
-----------	---	---	---	---

D=

	u1
y1	0

Los autovalores del sistema se calculan mediante el comando “eig” de MATLAB©.

$$eig = (A_{amp} - B_{amp} * K_a) = \begin{bmatrix} -33.3140 & +0.0000i \\ -5.3622 & +0.0000i \\ -1.4955 & +0.0000i \\ -1.2075 & +1.1508i \\ -1.2075 & -1.1508i \end{bmatrix} \quad (155)$$

En este caso se verifica que en la ecuación de la matriz (155) existe una estabilidad con seguimiento a la referencia.

El diseño del control se presenta en MATLAB© - Simulink, como se muestra en la Figura (30), con el modelo en espacios de estados y las respectivas entradas de control.

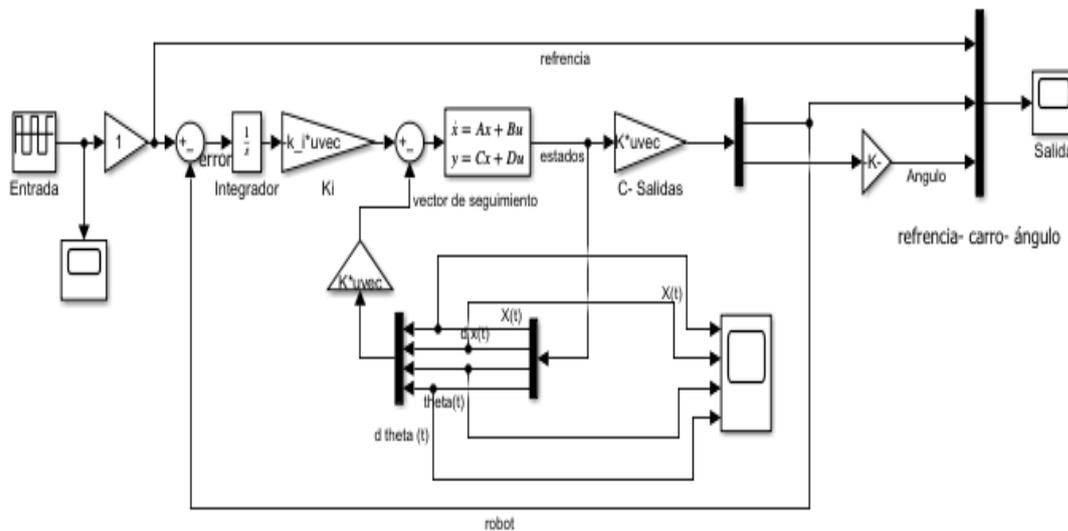


Figura 30 Control LQR sin perturbación ni ruido.

Realizado por: Yuri Torres, 2024.

En la Figura antes descrita del programa MATLAB© - Simulink se aplica una señal de referencia en la entrada $r(t)$ con escalón de -0.80m y 0.80m en un periodo de $T = 15$ s

por un tiempo de 150 segundos para observar el comportamiento de la señal aplicada como indica la Figura (31).

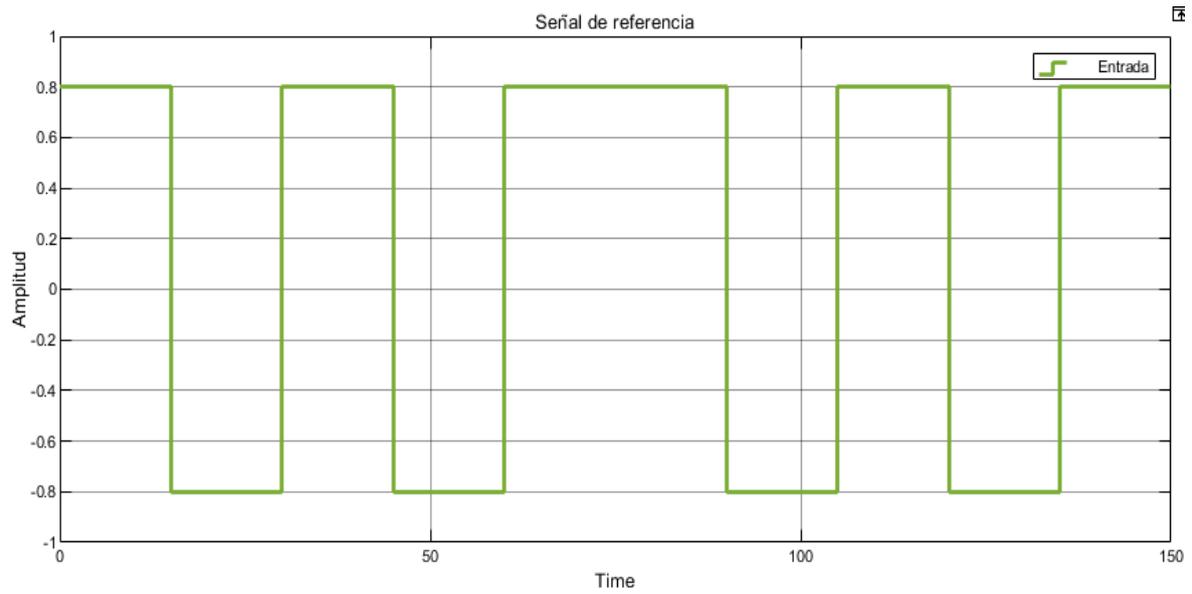


Figura 31 Señal de referencia.

Realizado por: Yuri Torres, 2024.

Los estados regulados mediante el controlador LQR se muestran en la imagen (32).

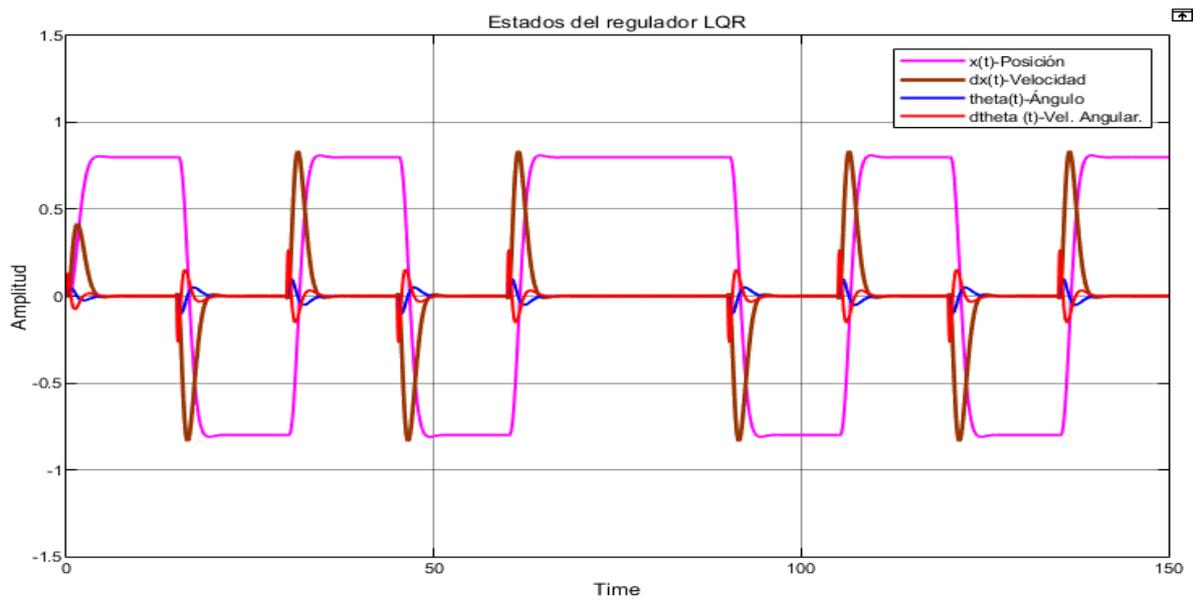


Figura 32 Estados del sistema regulado con LQR

Realizado por: Yuri Torres, 2024.

En la figura (33), se muestran las salidas del controlador LQR.



Figura 33 Salidas LQR

Realizado por: Yuri Torres, 2024.

Con las imágenes antes descritas (32) y (33), se observa que el control sigue la referencia sin perturbaciones, y la estabilización de ángulo también se comporta aceptable. En base estos criterios se procede hallar la respuesta en el tiempo de: establecimiento, sobre impulso máximo y parámetros de ángulo y a la señal de control.

3.5.1.1 Respuesta del tiempo de estabilización (t_{ss})

La Figura (34), muestra la gráfica de MATLAB© - Simulink con seguimiento a la referencia para el tiempo de estabilización en $t_{ss} = 5.782$ s en un 2%.

$$y_{max} = 0.80 + 0.02(0.80) = 0.816 \quad (156)$$

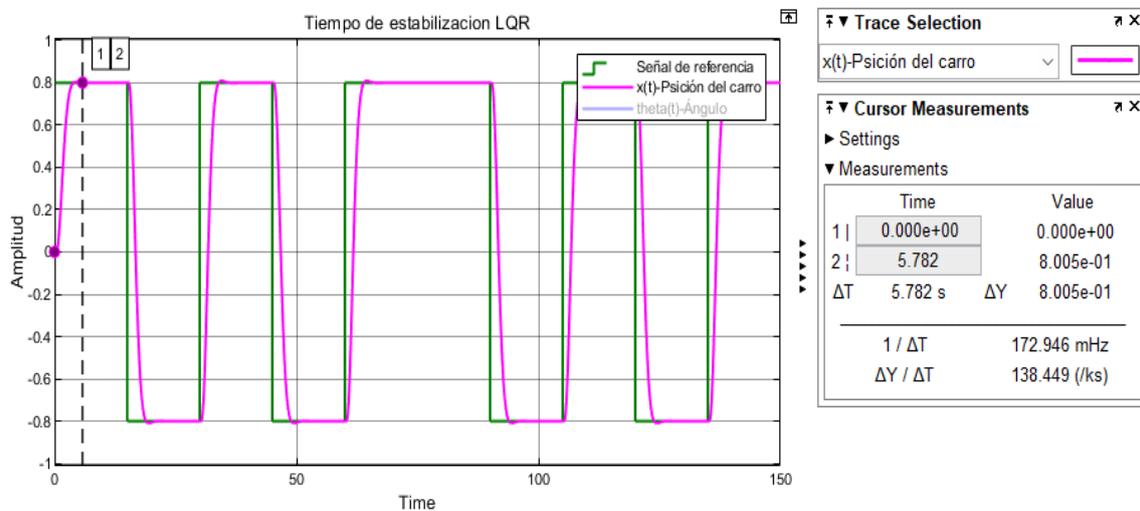


Figura 34 Tiempo de estabilización LQR.

Realizado por: Yuri Torres, 2024.

3.5.1.2 Respuesta del sobre impulso máximo (M_p)

En la Figura (35), se muestra la gráfica para el sobre impulso versus la referencia.

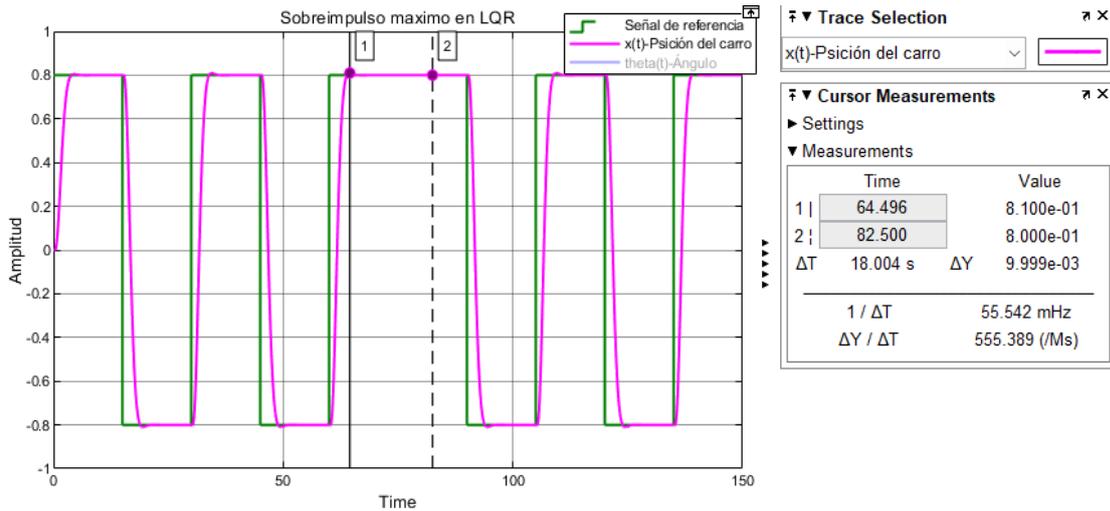


Figura 35 Máximo impulso LQR.

Realizado por: Yuri Torres, 2024.

Para hallar el (M_p) se procede a la ecuación siguiente:

$$M_p = \frac{0.8100 - 0.80}{0.80} = 0.0125 \quad (157)$$

$$M_p = 1.25\% \quad (158)$$

3.5.1.3 Respuesta del ángulo de inclinación (θ)

En la Figura (36), se muestra el ángulo controlado desde su instante inicial en 10° grados sexagesimales con una oscilación en el rango de (5.719 y -5.714) con un desplazamiento angular previsto:

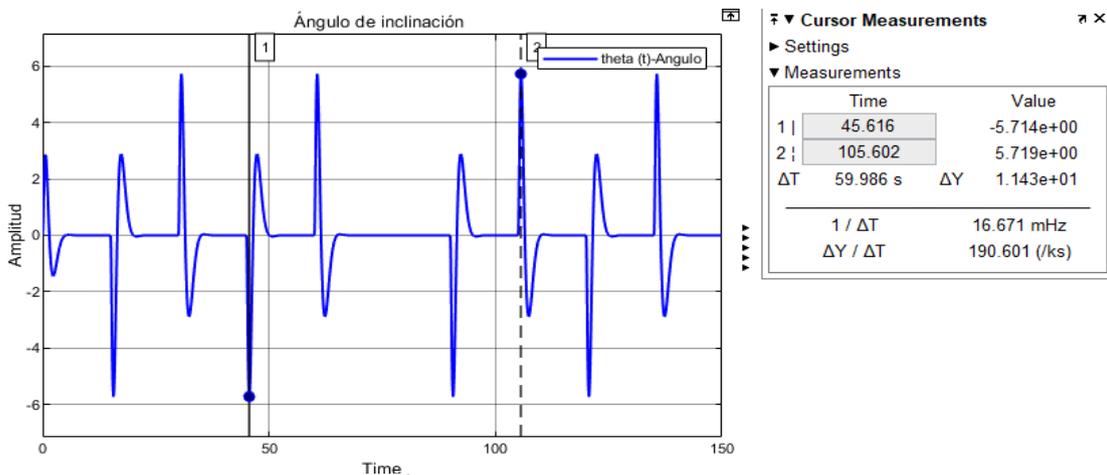


Figura 36 Ángulo de inclinación en el control LQR.

Realizado por: Yuri Torres, 2024.

$$\theta_{max} = 5.719 \text{ y } \theta_{min} = -5.714$$

En la imagen (37) se muestra el desplazamiento del ángulo desde su instante inicial en 10° .

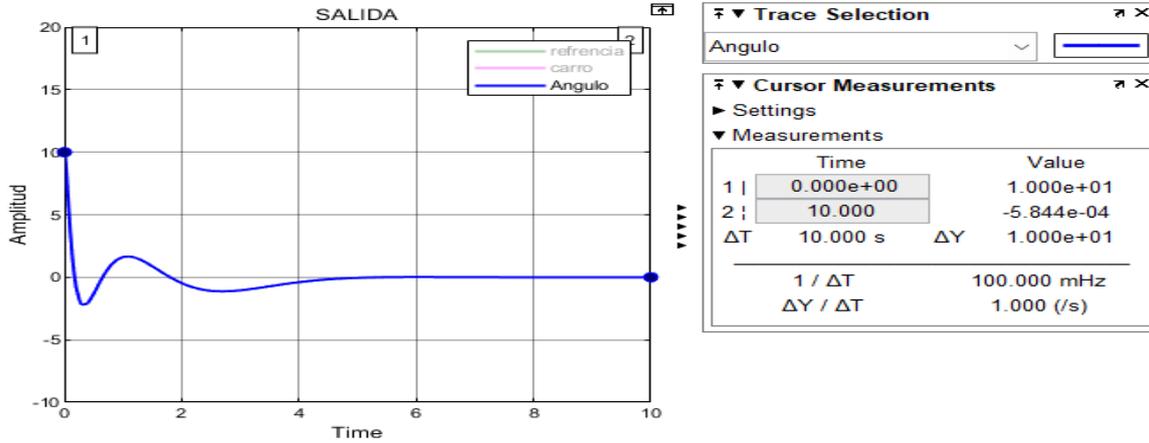


Figura 37 Regulación del ángulo en LQR

Realizado por: Yuri Torres, 2024.

3.5.1.4 Respuesta de la señal de control $u(t)$

En la Figura (38) se evidencia la señal de control inicia en 0V y obtiene valores máximos 5.593V y valor mínimo de -0.4461V.

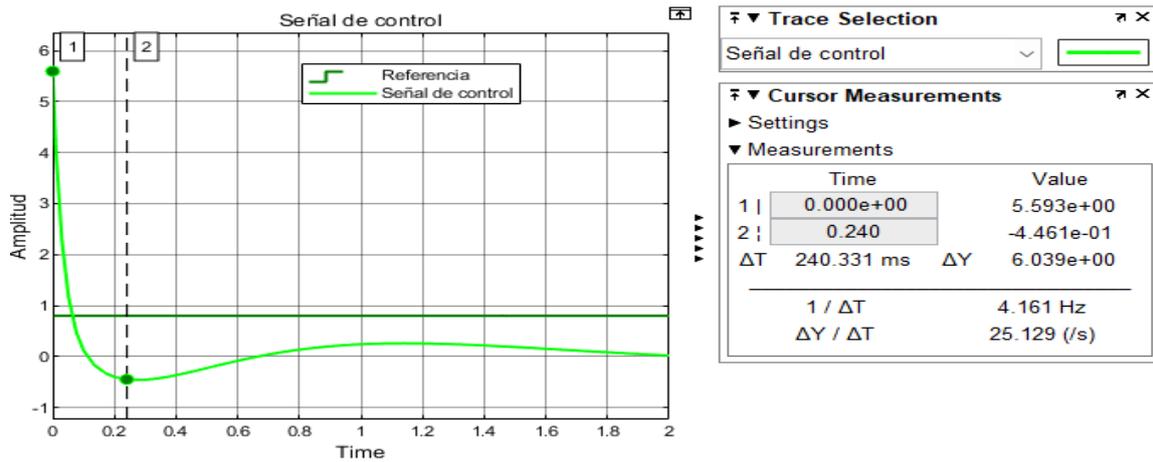


Figura 38 Valor inicial en señal de control en LQR.

Realizado por: Yuri Torres, 2024.

En la Figura (39), se observa que existen un intervalo de valores en la señal de control de:

$$-0.1121 \leq u(t) \leq 0.9416$$

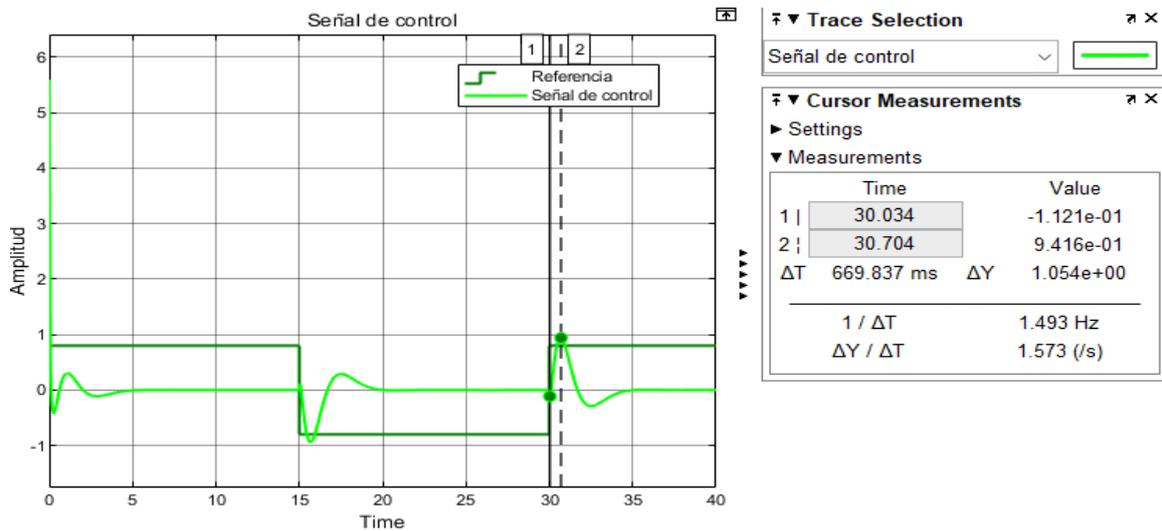


Figura 39 Señal de control en LQR.

Realizado por: Yuri Torres, 2024.

3.5.2 Control LQR con ruido y perturbación

Concluido el diseño del control LQR con seguimiento a referencias, se procede a introducir ruidos y perturbaciones como especifica la imagen (40).

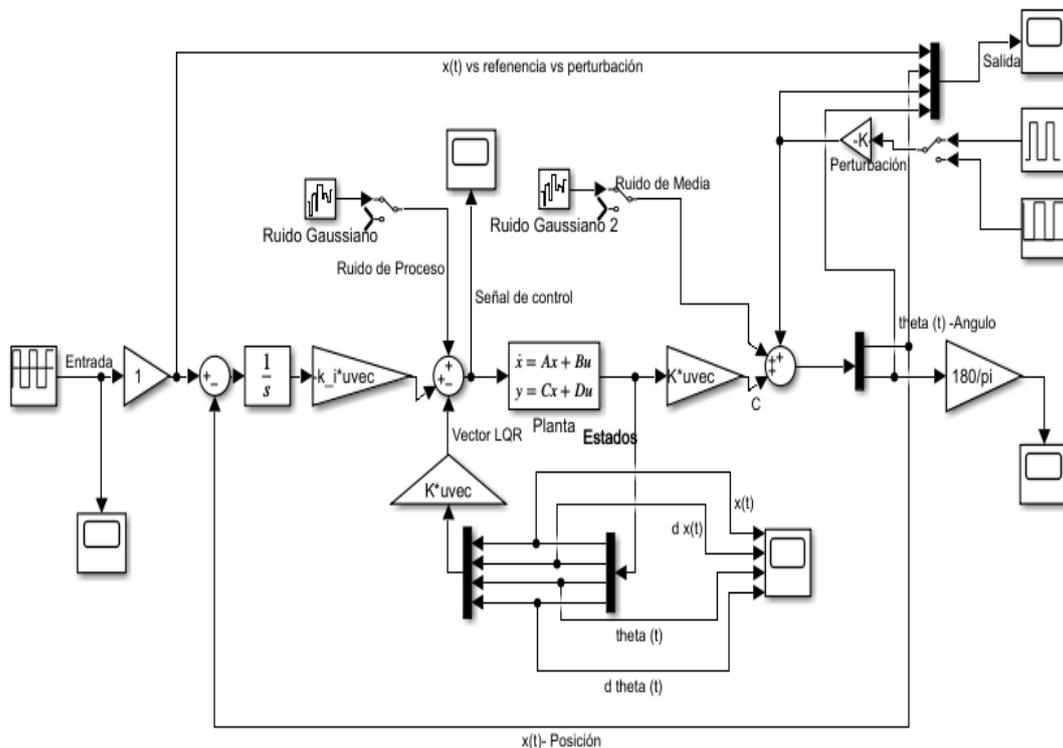


Figura 40 Control LQR con ruidos y perturbaciones.

Realizado por: Yuri Torres, 2024.

En la gráfica del control LQR con ruidos y perturbaciones (40), consta de ruidos blancos gaussianos produciendo señales estocásticas mediante ruido de media que puede ser causa de los sensores a la salida del sistema con densidad espectral de potencia y ruido de proceso que refiere a la parte inherente como ruido en el modelo matemático.

El ruido de Media con distribución gaussiana se presenta en los sensores y afecta directamente la salida del sistema y posee las siguientes características en densidad espectral de potencia (PSD) y voltaje, como se aprecia en la Figura (41).

$$(PSD) = 1 * 10^{-4} \frac{W}{hz} \quad (159)$$

$$Vmax = 120 mV \quad (160)$$

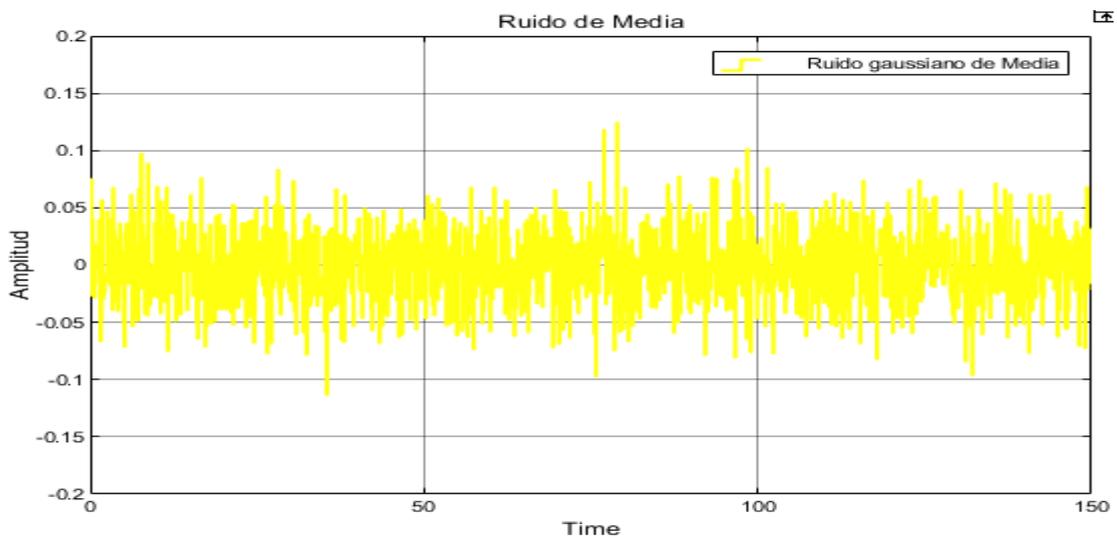


Figura 41 Ruido de media del sistema controlador LQR.

Realizado por: Yuri Torres, 2024.

El ruido de proceso con distribución gaussiana consiste en las alteraciones que sufren las variables del modelo matemático, a continuación, los parámetros permitidos en la imagen:

$$(PSD) = 1 * 10^{-7} \frac{W}{hz} \quad (161)$$

$$Vmax = 4 mV \quad (162)$$

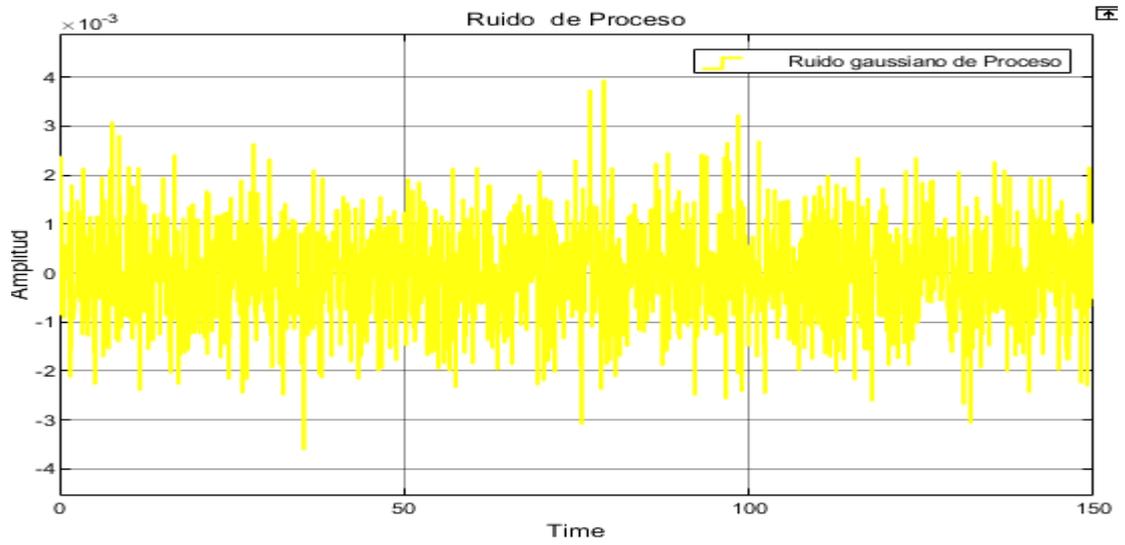


Figura 42 Ruido de Proceso del sistema del controlador LQR

Realizado por: Yuri Torres, 2024.

Se utiliza una perturbación externa que representa mínimos impulsos de $T=5$ segundos con amplitud de $F=0.16$, que conduce directamente a la salida del sistema después de la estabilización. Ver figura (43).

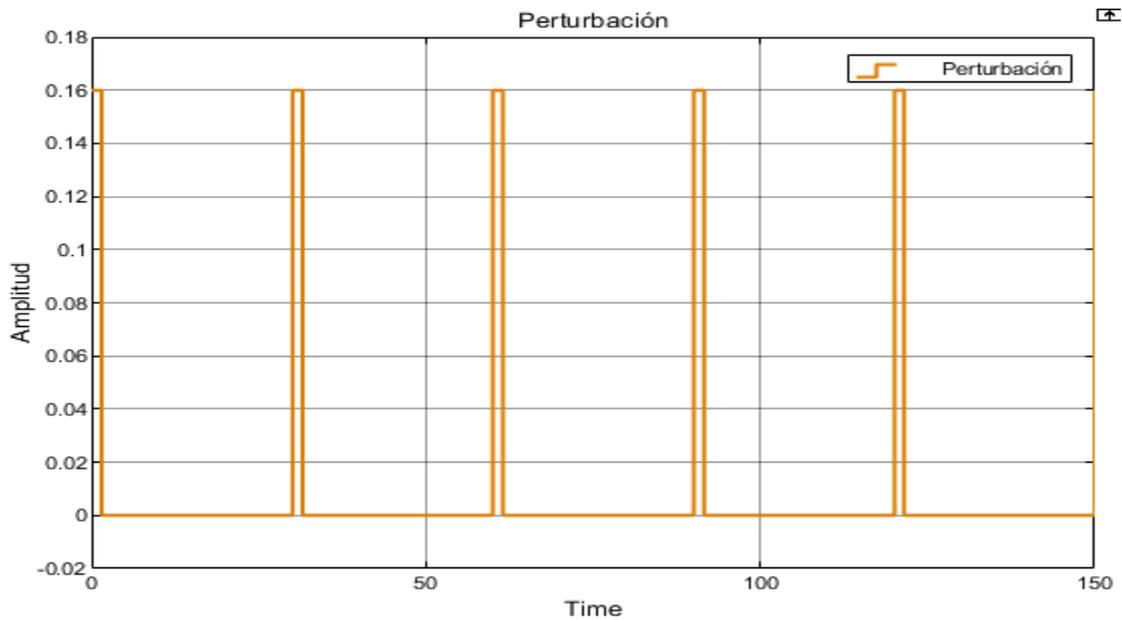


Figura 43 Perturbación externa en el control LQR.

Realizado por: Yuri Torres, 2024.

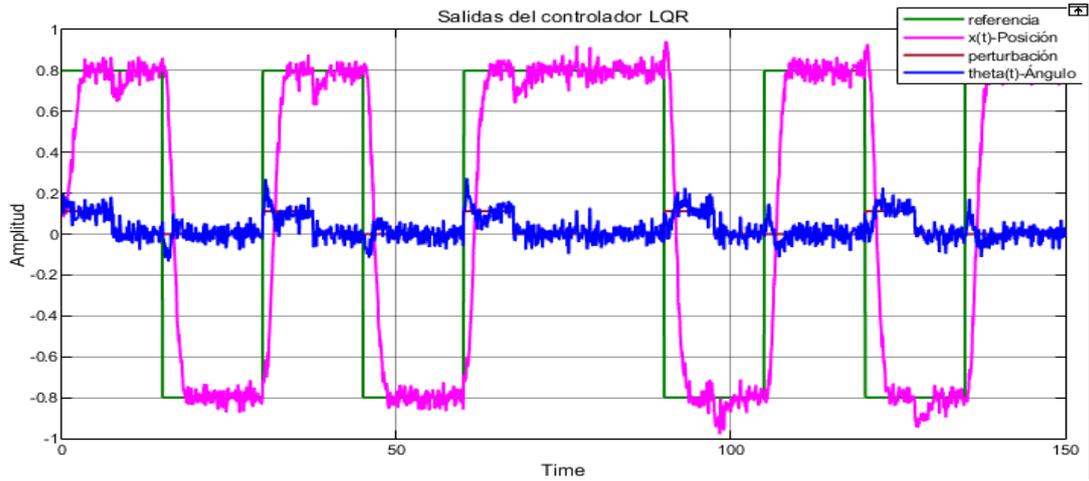


Figura 44 Salidas LQR con perturbación y ruido.

Realizado por: Yuri Torres, 2024.

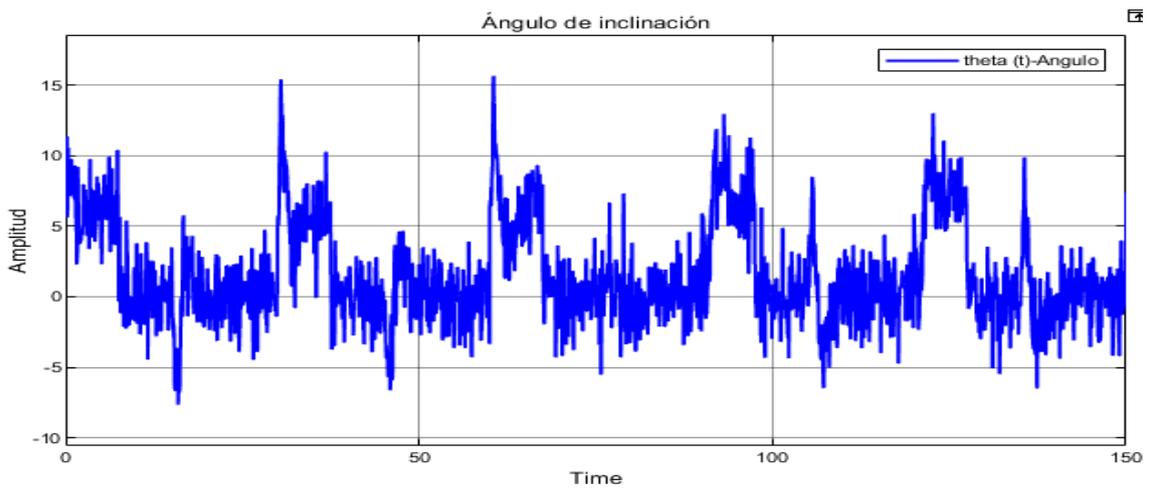


Figura 45 Contaminación del Ángulo del sistema en el control LQR.

Realizado por: Yuri Torres, 2024.

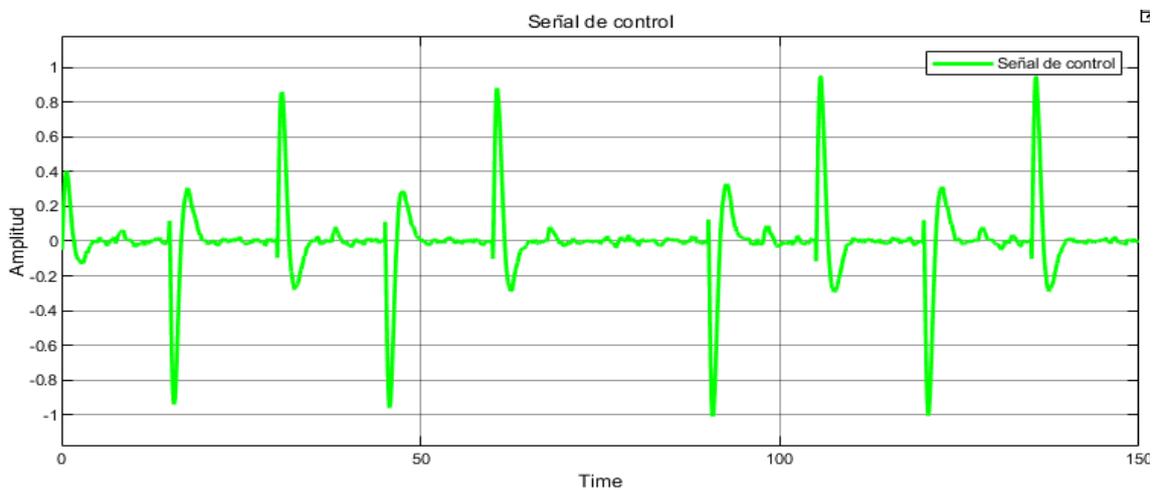


Figura 46 Señal de control contaminada en LQR.

Realizado por: Yuri Torres, 2024.

En las figuras (44), (45) y (46) se constata que existen afectaciones directas al sistema por causa de ruido y perturbación; por tal motivo se necesita un filtro de Kalman que reduzca los efectos antes descritos y permita estimar eficientemente los estados.

3.5.2.1 Respuesta del tiempo de estabilización (t_{ss})

La Figura (47), del tiempo de estabilización muestra el tiempo de estabilización en $t_{ss} = 4.360s$ en criterio del 2%.

$$y_{max} = 0.50 + 0.02(0.50) = 0.51$$

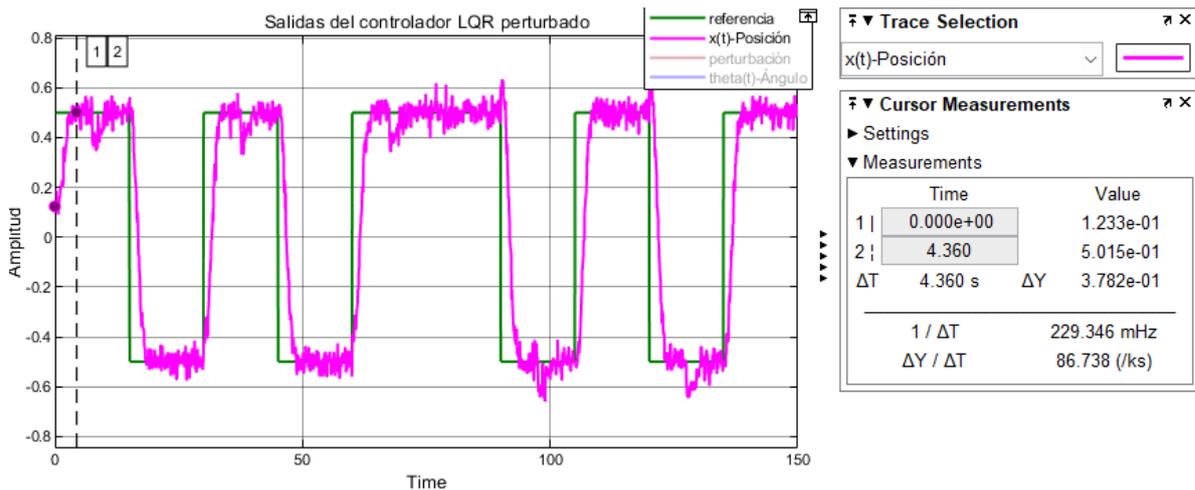


Figura 47 Tiempo de estabilización LQR perturbado.

Realizado por: Yuri Torres, 2024.

3.5.2.2 Respuesta del sobre impulso máximo (M_p)

En la figura (48), se muestra la gráfica para el sobre impulso versus la referencia. Para hallar el (M_p) se procede a la ecuación siguiente:

$$M_p = \frac{0.5661 - 0.50}{0.50} = 0.1322 \quad (163)$$

$$M_p = 13.22\% \quad (164)$$

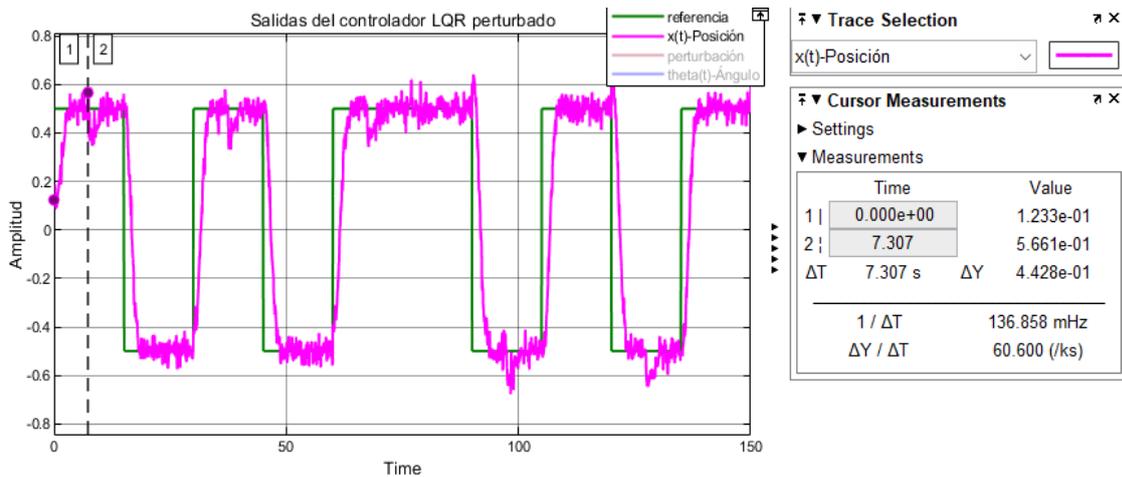


Figura 48 Mximo impulso LQR perturbado.

Realizado por: Yuri Torres, 2024.

3.5.2.3 Respuesta del ngulo de inclinacin (θ)

En la Figura (49), se muestra el ngulo controlado desde su instante inicial en 0° grados sexagesimales:

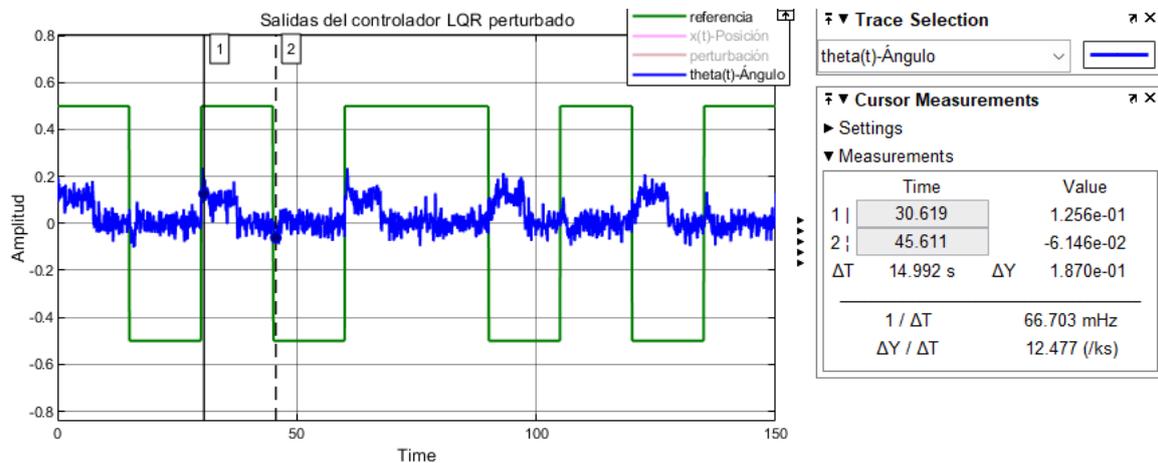


Figura 49 ngulo de inclinacin en el control LQR perturbado.

Realizado por: Yuri Torres, 2024.

$$\theta_{max} = 0.1256^\circ \text{ y } \theta_{min} = -0.06146^\circ$$

3.5.2.4 Respuesta de la seal de control (u)

Se evidencia la seal de control en regulacin con el LQR y se observa que la energa de control oscila entre los $-0.5785 \leq u(t) \leq 0.5024$. Figura (50).

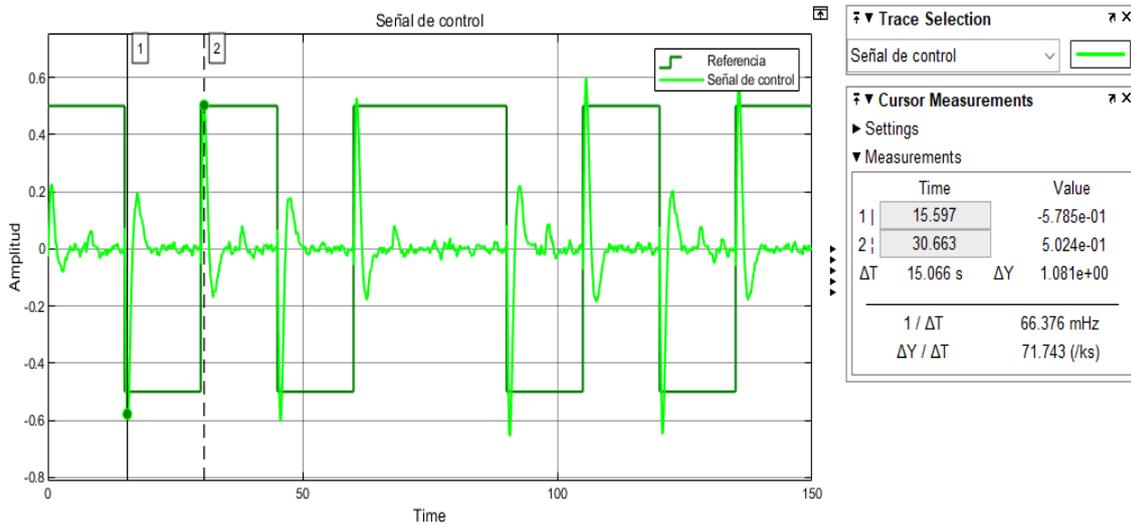


Figura 50 Valor inicial en señal de control en LQR perturbado.

Realizado por: Yuri Torres, 2024.

También se evidencia que empieza en 0V y alcanza valores máximo 0.2271 V y valor mínimo -0.0818V, como indica el apartado siguiente:

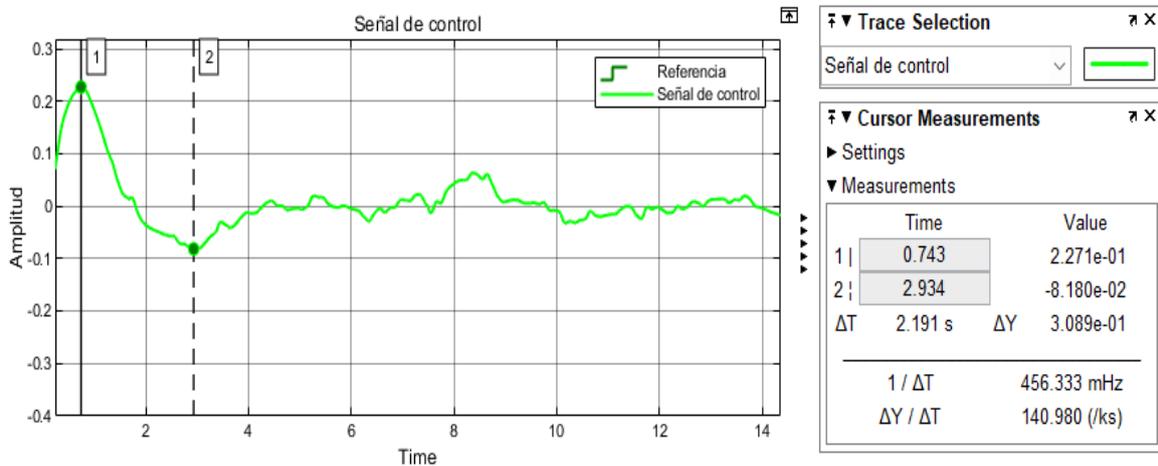


Figura 51 Señal de control en LQR perturbado.

Realizado por: Yuri Torres, 2024.

Después del diseño del filtro de Kalman, se implementa la conexión entre el control óptimo y el estimador óptimo para cumplir con el control Lineal cuadrático gaussiano y cumplir la misión del estudio.

3.5.3 Diseño del Filtro de Kalman.

En esta sección se aborda el diseño de dos filtro de Kalman: uno para el estado estacionario y otro para el tiempo continuo, ambos con fines diferentes: el filtro en tiempo continuo es quien estima en tiempo dinámico con variables que cambian constantemente y el filtro de estado estacionario estima variables que son constantes debido

a su estado. Ambos buscan la manera de minimizar la matriz de covarianza del error de estimación.

3.5.3.1 Filtro de Kalman Bucy

Lase ecuaciones (165) y (167) representan el sistema dinámico estocástico en espacios de estados con ruido, donde $\delta(\tau)$ es la expresión del impulso, (v) y (w) son ruidos con promedios de cero y se distribuyen de acuerdo con la campana de Gauss.

En la expresión (166) se representa el ruido que afecta la dinámica del sistema y deja de ser nula cuando $t = s$

En la siguiente ecuación (166) representa la fórmula necesaria para mostrar el ruido que afecta las mediciones u observaciones del sistema y deja de ser nula cuando $t = s$

$$\dot{x}(t) = Ax(t) + Bu(t) + Fv(t) \quad (165)$$

$$E\{v(s)v^T(t)\} = Q_k(t) \delta(t - s) \quad (166)$$

$$y(t) = Cx(t) + w(t) \quad (167)$$

$$E\{w(s)w^T(t)\} = R_k(t) \delta(t - s) \quad (168)$$

El filtro de Kalman para estados estacionarios debe encontrar la estimación $\hat{x}(t)$ que minimice el error en los estados de medios cuadrados como indica la expresión siguiente:

$$E\{(x(t) - \hat{x}(t)) (x(t) - \hat{x}(t))^T\} \quad (169)$$

Se utiliza el rango en $y(\tau)$ para valores de τ entre 0 y t dando solución al problema de optimización en la ecuación (36) y (37). Donde $L = PC^T R^{-1}$ llamado ganancia de filtro de Kalman es considerada solución simétrica positiva. La ecuación (170) proviene de la expresión de Riccati (53)

$$AP + PA^T - PC^T R^{-1} CP + FQF^T = 0 \quad (170)$$

Para encontrar el error de la salida medida y estimada se presenta la forma matemática (169), donde $y(t)$ es la salida medida, $C\hat{x}(t)$ es la salida estimada, $r(t)$ es el residuo y $\hat{x}(t)$ parte de la ecuación (36).

$$r(t) = y(t) - C\hat{x}(t) \quad (171)$$

El modelo en espacios de estados queda expresado de la siguiente manera:

Donde u es la entrada al sistema y w es el ruido blanco de proceso.

$$\dot{x} = Ax + Bu + Gw \quad (172)$$

$$y = Cx + Dx + Hw + v \quad (173)$$

De las ecuaciones antes descritas (160) y (161), se establecen las matrices aumentadas para la implementación en el software MATLAB©.

$$A_k = A \quad B_k = [B \ G] \quad C_k = C \quad D_k = [D \ H]$$

Matrices aumentadas en espacios de estados.

$sys_k =$

$A_k =$

	x1	x2	x3	x4
x1	0	1	0	0
x2	0	-0.2966	0	-0.58
x3	0	0	0	1
x4	0	4.236	140.1	0.5259

$B_k =$

	u1	u2
x1	0	1
x2	1.187	1
x3	0	1
x4	-16.94	1

$C_k =$

	x1	x2	x3	x4
y1	1	0	0	0

$D_k =$

	u1	u2
y1	0	0

Matrices de Covarianza

Las matrices de covarianza Q_k , representa intensidad del ruido de proceso del sistema y R_k intensidad de ruido de media en las salidas del sistema; la importante contribución de estas matrices define el filtrado del ruido gaussiano y el efecto positivo en el comportamiento del sistema. Se utiliza los valores para $Q_k = 0.75$ y $R_k = 1.5$, en base

a las matrices de covarianza del error, se obtiene el sistema de matrices extendidas para el filtro de Kalman que se generan con el comando MATLAB© - Simulink, dando matrices en espacios de estados:

A=

	x1_e	x2_e	x3_e	x4_e
x1_e	-25.51	1	0	0
x2_e	-325.2	-0.2966	0	-0.58
x3_e	576	0	0	1
x4_e	6912	4.236	140.1	0.5259

B=

	u1	y1
y1_e	0	25.51
x2_e	1.187	325.2
x3_e	0	-576
x4_e	-16.94	-69.12

C=

	x1_e	x2_e	x3_e	x4_e
y1_e	1	0	0	0
x1_e	1	0	0	0
x2_e	0	1	0	0
x3_e	0	0	1	0
x4_e	0	0	0	1

D=

	u1	y2
y1_e	0	0
x1_e	0	0
x2_e	0	0
x3_e	0	0
x4_e	0	0

También se generan las matrices de ganancia del filtro L y covarianza del error P .

$$L = 1.0e + 03 *$$

$$\begin{bmatrix} 0.0255 \\ 0.3252 \\ -0.5760 \\ -6.9120 \end{bmatrix}$$

$P = 1.0e+06^*$

$$\begin{bmatrix} 0.0000 & 0.0015 & -0.0003 & -0.0104 \\ 0.0005 & 0.0066 & 0.0011 & -0.1401 \\ -0.0009 & -0.0117 & 0.1088 & 0.2488 \\ -0.0104 & -0.2488 & -0.4350 & 2.9858 \end{bmatrix}$$

Luego de este proceso ya se efectúa el filtro de Kalman y se utiliza el bloque de MATLAB© -Simulink 'Kalman Filter'. Ver Figura (52).

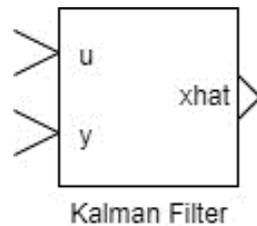


Figura 52 Bloque del Filtro de Kalman.

Realizado por: Yuri Torres, 2024.

3.5.3.2 Filtro de Kalman

El algoritmo de estimación antes descrito, cumple el propósito de estimación necesaria en sistemas lineales de tiempo continuo y discreto. En la figura (53) se muestra un entorno de bloques de MATLAB© - Simulink donde se evidencia la entrada (u) y las medidas (y) encargadas de las estimaciones (\hat{x}). Cabe recalcar que también se puede observar la salida (\hat{y}) real del sistema.

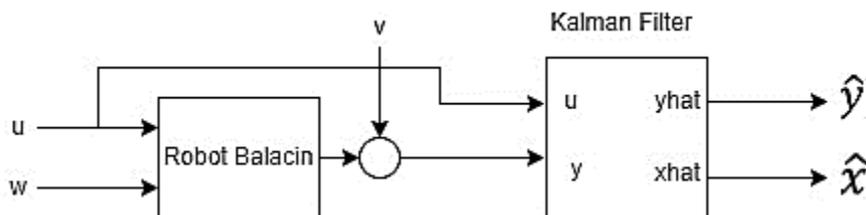


Figura 53 Kalman Estimador [40].

El bloque en la imagen del Filtro de Kalman en estado estacionario contiene matrices $A(t), B(t), C(t), D(t), G(t), H(t)$ y covarianzas $Q(t), R(t), N(t)$ constantes,

también utiliza una matriz P constante para minimizar la covarianza del error resuelta mediante la ecuación de Riccati en tiempo continuo:

$$P = \lim_{n \rightarrow \infty} E\{(x(t) - \hat{x}(t))(x(t) - \hat{x}(t))^T\} \quad (174)$$

Ahora se procede a la configuración e ingreso de valores en el bloque Filtro de Kalman. Luego de establecer el algoritmo matemático del Filtro de Kalman, se hace la implementación en el MATLAB© - Simulink con los ajustes necesarios para estimador óptimo de los estados y la eliminación del ruido gaussiano. A continuación, la Figura (54) muestra la configuración necesaria: Ingreso de los valores de matrices A, B, C y D.

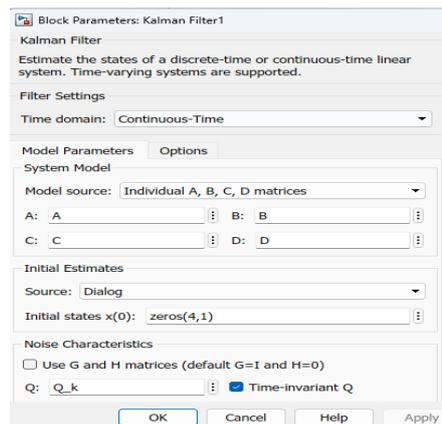


Figura 54 Configuración de bloque Kalman Filter [67].

En este paso de la figura (54) del bloque “Filter Kalman” seleccionamos la opción “Time domain”, “Continuous-time”, “Model Parameters”, “Individual A, B, C, D matrices” y luego se agregan los términos de las constantes ya establecidas del modelo dinámico.

Ahora se escoge el recuadro de “Initial Estimates”, “Source”, Dialog”, “Initial States x (0)” y se escribe “zeros (4,1)”. Esto ayuda a entender que los estados inician en 0.

Para “Noise Characteristics” se realizan las siguientes acciones:

Dejar en blanco la casilla “Use G and H matrices”, escribimos las variables Q y R matrices de covarianza de ruido, selecciona “Time Invariant Q” y “Time Invariant R”, Casilla N en 0 y se marca la opción “Time Invariant N”.

Finalmente, para las salidas estimadas es seleccionado “Additional Outports”, “Output estimated model output y”. Esto provoca observar las salidas estimadas óptimas en el bloque del filtro como indica la figura (50).

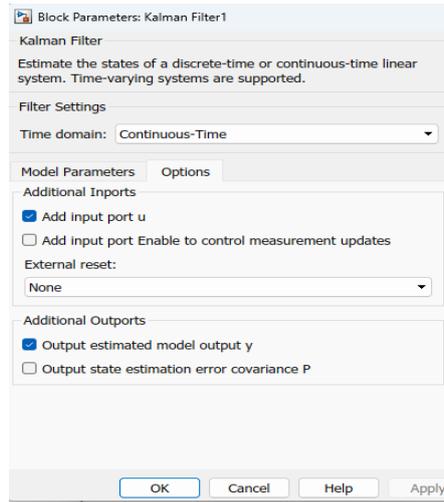


Figura 55 Configuración de Opciones Kalman Filter [67].

Luego de las configuraciones antes descritas, se procede a conexiones de bloques en el software MATLAB© - Simulink sin el control óptimo de seguimiento y sin perturbaciones externas, solo con ruidos blancos gaussianos donde se podrá verificar el rendimiento del Filtro. Se hace observaciones que en la Figura (56), solo está el diseño y demostración de las capacidades del Filtro.

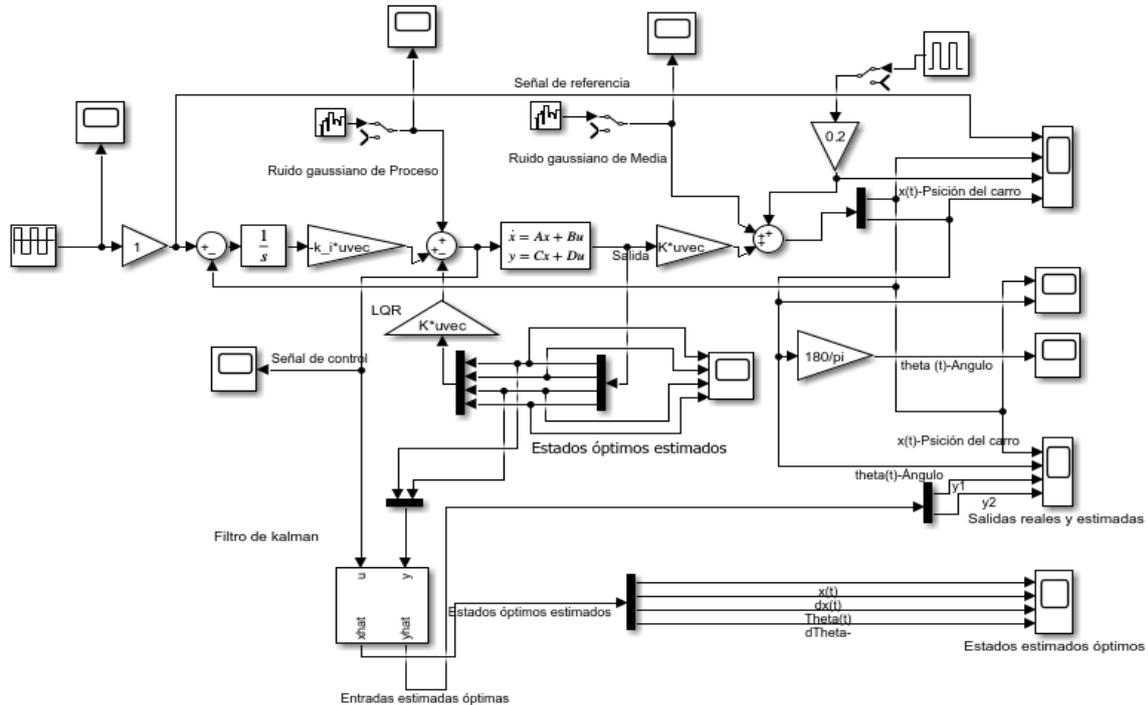


Figura 56 Sistema de control y Filtro de Kalman.

Realizado por: Yuri Torres, 2024.

Una vez ejecutado el algoritmo del Filtro de Kalman sin el sistema controlado, se procede a examinar resultados como las estimaciones de estados de entrada y salida en comparación con los valores reales.

En la Figura (57) se muestra el resultado de la simulación en MATLAB© - Simulink de los estados estimados óptimos, donde el ruido blanco gaussiano es eliminado proporcionando estimaciones próximas a los estados reales, casi listos para la conexión del control.

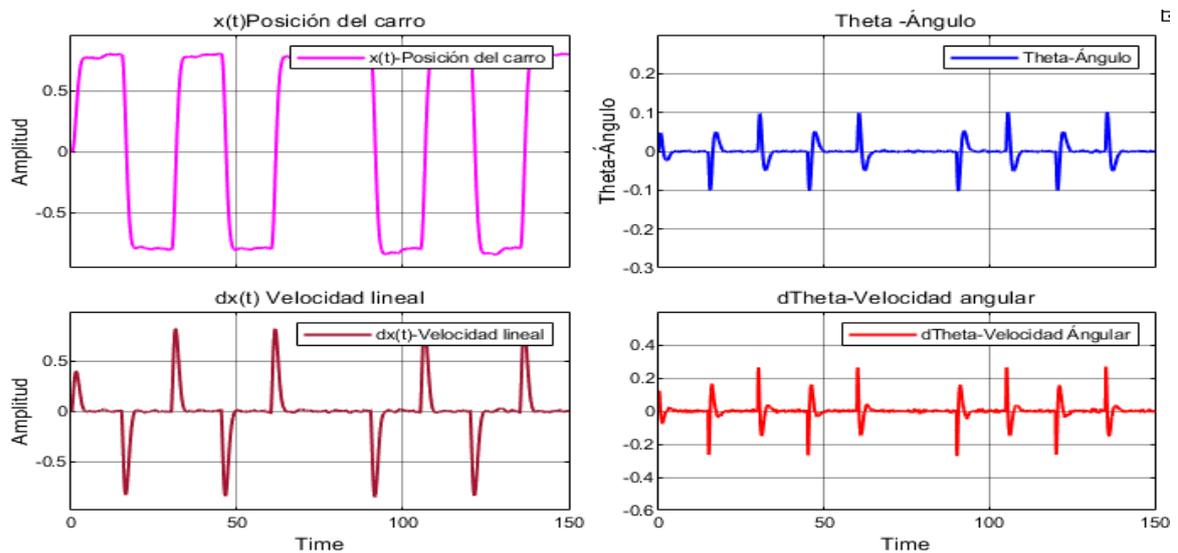


Figura 57 Salidas estimadas óptimas.

Realizado por: Yuri Torres, 2024.

En la Figura (58) se presentan las estimaciones de las salidas óptimas, con evidencias de salidas reales ruidosas y estimaciones limpias.

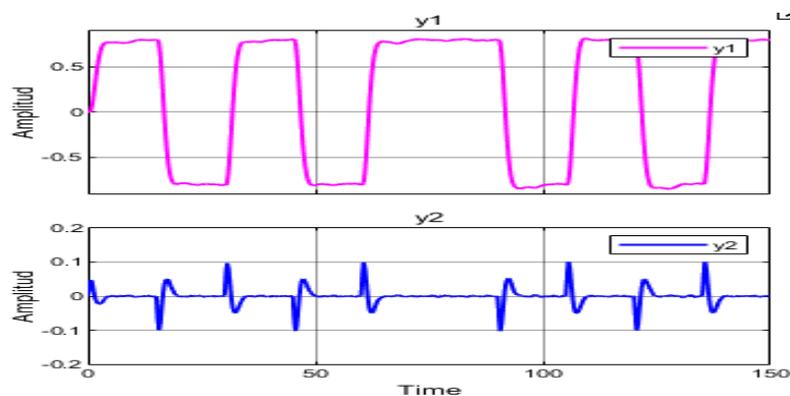


Figura 58 Salidas óptimas estimadas.

Realizado por: Yuri Torres, 2024.

Se analiza la gráfica (59) y se observa un contraste entre las salidas reales y filtradas con estimaciones óptimas.

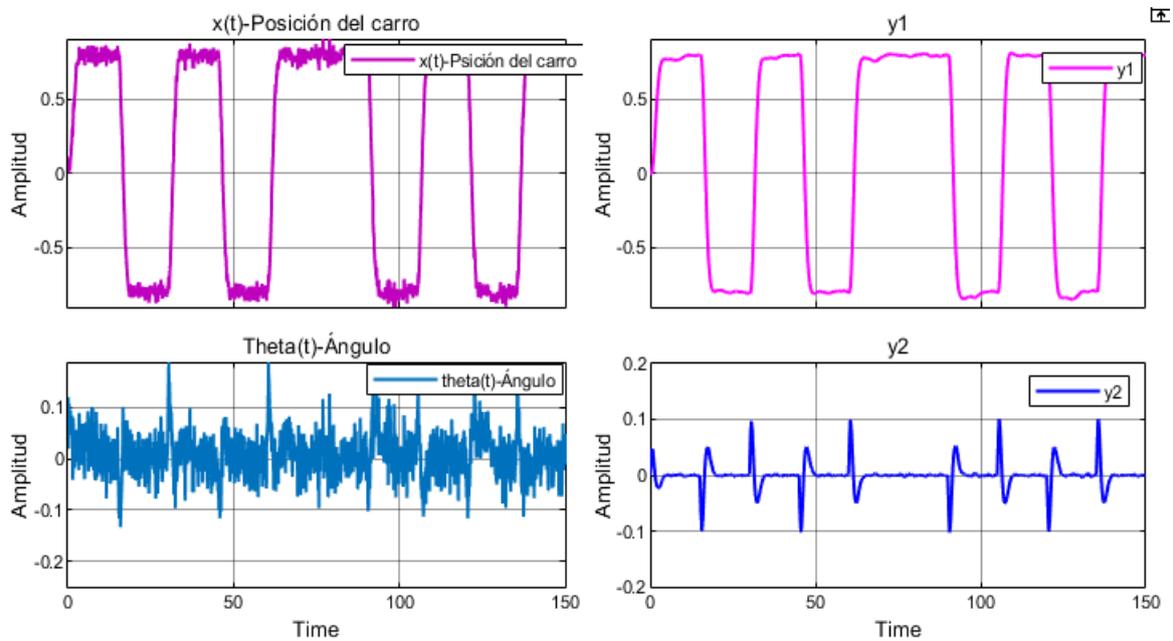


Figura 59 Salidas reales y salidas filtradas.

Realizado por: Yuri Torres, 2024.

Para finalizar el filtro de Kalman, se puede decir que realiza estimaciones precisas entre las señales de salidas reales y las estimadas filtrando al máximo el ruido.

También se ha observado la forma en que rechaza el ruido estocástico en las salidas, dando respuestas óptimas al sistema, y es consecuencia es considerado el mejor estimador óptimo lineal. Luego de este filtrado se procede a conectar al sistema de control, con los estados óptimos ya estimados en el controlador lineal cuadrático LQR, y a su vez modelar un controlador servo-lineal cuadrático gaussiano. A continuación, se desarrolla el proceso.

3.5.4 Servo-controlador LQG

En esta sección, se diseña el servo-controlador lineal cuadrático gaussiano, sistema óptimo para lidiar con incertidumbres y ruidos en el proceso. Este controlador se compone de un Filtro de Kalman y un controlador LQR resultando un sistema de control robusto resistente a ruidos y perturbaciones. En la ecuación (175) se muestra el modelo en espacios de estados del controlador LQG y en la figura (60) el diagrama de bloques MATLAB© - Simulink para las conexiones respectivas para el controlador de dos grados de libertad.

$$\hat{x} = [A - LC - (B - LD)K]\hat{x} + Ly \quad (175)$$

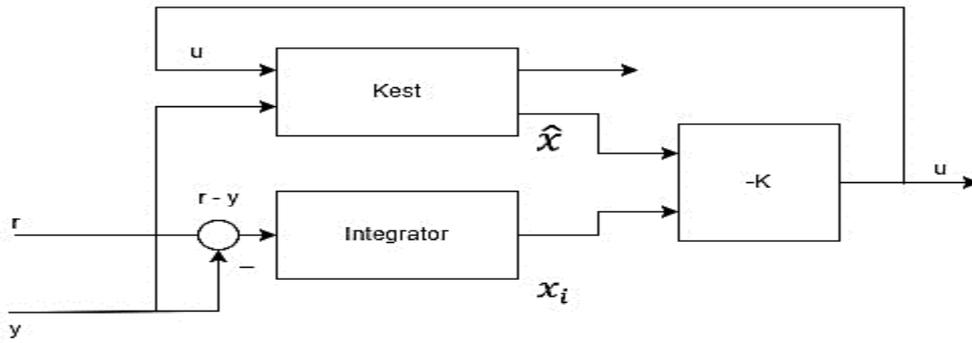


Figura 60 Servo controlador LQG [68].

El servo-controlador lineal cuadrático gaussiano, tiene un modelo en espacios de estados donde K_x y K_i son ganancias de retroalimentación para x y x_i .

$$\begin{bmatrix} \hat{x} \\ x_i \end{bmatrix} = \begin{bmatrix} A - BK_x - LC + LDK_x & -BK_i + LD K_i \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ x_i \end{bmatrix} + \begin{bmatrix} 0 & L \\ I & -I \end{bmatrix} \begin{bmatrix} r \\ y \end{bmatrix} \quad (176)$$

$$u = [-K_x - K_i] \begin{bmatrix} \hat{x} \\ x_i \end{bmatrix} \quad (177)$$

En base a las ecuaciones de estados descritas en el apartado anterior, se implementa el servo-controlador en MATLAB© utilizando el comando “lqgtrack”, proporcionando el argumento necesario del modelo en espacios de estados del filtro de Kalman “kest” y la ganancia del controlador lineal cuadrático del sistema de seguimiento “k_lqi”. Estos resultados se guardan en la variable “K_lqg” que en este caso es el modelo del controlador LQG.

Mediante los comandos “K_lqg=lqgtrack (kest, K_lqi)”, se obtiene el modelo en espacios de estados del servo-controlador LQG en el software MATLAB©.

K_lqg

A =

	x1_e	x2_e	x3_e	x4_e	xi1
x1_e	-25.51	1	0	0	0
x2_e	-315.6	7.351	38.02	2.954	-5.306
x3_e	576	0	0	1	0
x4_e	6775	-105	-402.9	-49.94	75.77
xi_e	0	0	0	0	0

B=

	r1	y1
x1_e	0	25.51
x2_e	0	325.2
x3_e	0	-576
x4_e	0	-6912
xi1	1	-1

C=

	x1_e	x2_e	x3_e	x4_e	xi1
u1	8.088	6.445	32.05	2.978	-4.472

D=

	u1	u2
y1	0	0

En el modelo de espacios antes indicado se presentan los estados “ x1_e, x2_e, x3_e, x4_e” y la variable que sigue la referencia “ xi1”, esto indica que el servo-controlador utiliza las estimaciones optimas de los estados hallados por el filtro de Kalman.

El modelo de espacios de estados del servo-controlador LQG, se implementa en diagrama de bloques MATLAB© - Simulink con la señal de referencia, y las dos variables de salida (retroalimentadas para el seguimiento) como entrada y la señal de control como única salida debido al modelo del servo-controlador. Además es importante resaltar que el modelo en bloques de MATLAB© - Simulink debe realizarse por medio del principio de separación, es decir no implementar un solo bloque de “Space state” debido a respuestas inexactas [40], por tal motivo se realiza por separado utilizando bloques “Kalman Filter-Bucy” en estado estacionario y bloques de ganancia para el controlador LQR.

3.5.5 Control LQG diseño en solo bloque

En la figura (61) se muestra el diagrama de bloques con el principio de separación para evitar inconsistencias en los resultados.

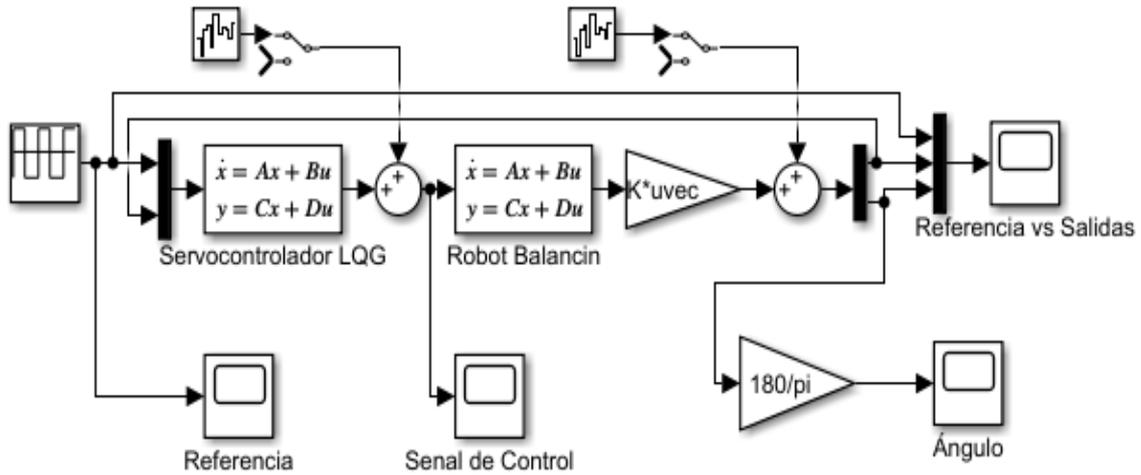


Figura 61 Controlador LQG en un solo bloque.

Realizado por: Yuri Torres, 2024.

La gráfica (61) del servo-controlador LQG mediante conexiones independientes en el diagrama de bloque MATLAB© - Simulink, se obtienen las respuestas del ruido, perturbaciones, salidas del sistema, estados estimados, comparaciones entre las salidas observadas y estimadas y la señal de control o señales del servo-sistema.

En la figura (62), se observa el Ruido de Media del servo-sistema con los parámetros de las siguientes expresiones:

$$(PSD) = 1 * 10^{-10} \frac{W}{hz} \quad (178)$$

$$Vmax = 0.1247 m V. \quad (179)$$

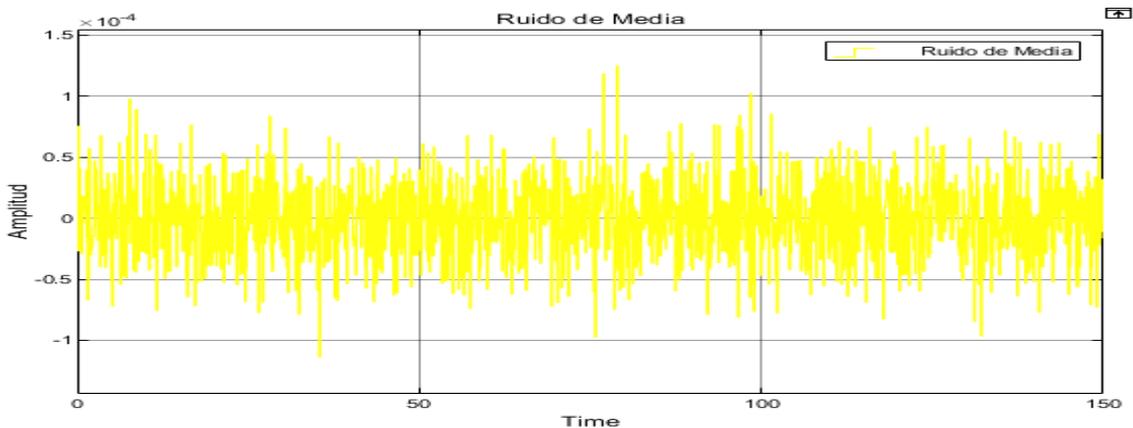


Figura 62 Ruido de Media en el sistema LQG.

Realizado por: Yuri Torres, 2024.

En la gráfica (63), se muestra el ruido de proceso del servo-sistema con los siguientes parámetros que definen el Ruido Gaussiano Blanco con la densidad espectral de potencia y la amplitud de voltaje.

$$(PSD) = 1 * 10^{-9} \frac{W}{hz} \quad (180)$$

$$V_{max} = 0.3944 \text{ mV} \quad (181)$$

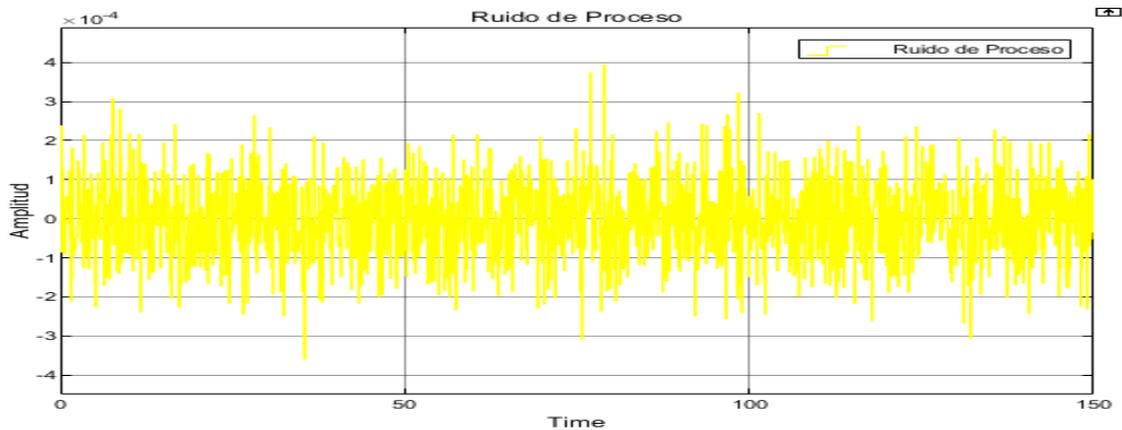


Figura 63 Ruido de Proceso en el sistema LQG.

Realizado por: Yuri Torres, 2024.

En la figura (64), (65) y (66) se evidencia el desempeño y el funcionamiento del sistema en la señal de referencia y en los ruidos de proceso y media. Claro está que el sistema no trabaja de manera correcta debido a los ruidos aplicados y la implantación de un solo bloque de controlador LQG donde está el Filtro de Kalman y el controlador LQR, esto demuestra que se limitan la demostración de sus capacidades y no es la forma correcta de hacerlo.

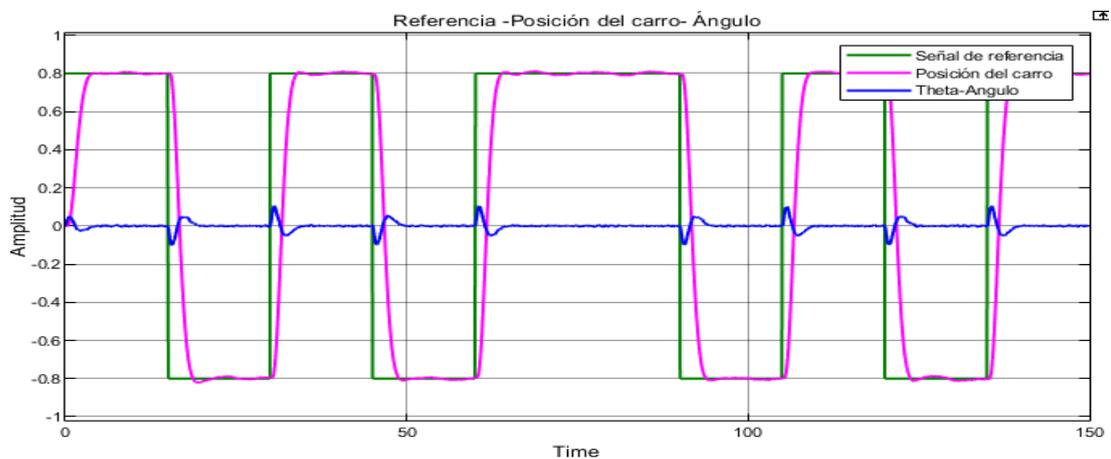


Figura 64 Referencia- Salidas LQG.

Realizado por: Yuri Torres, 2024.

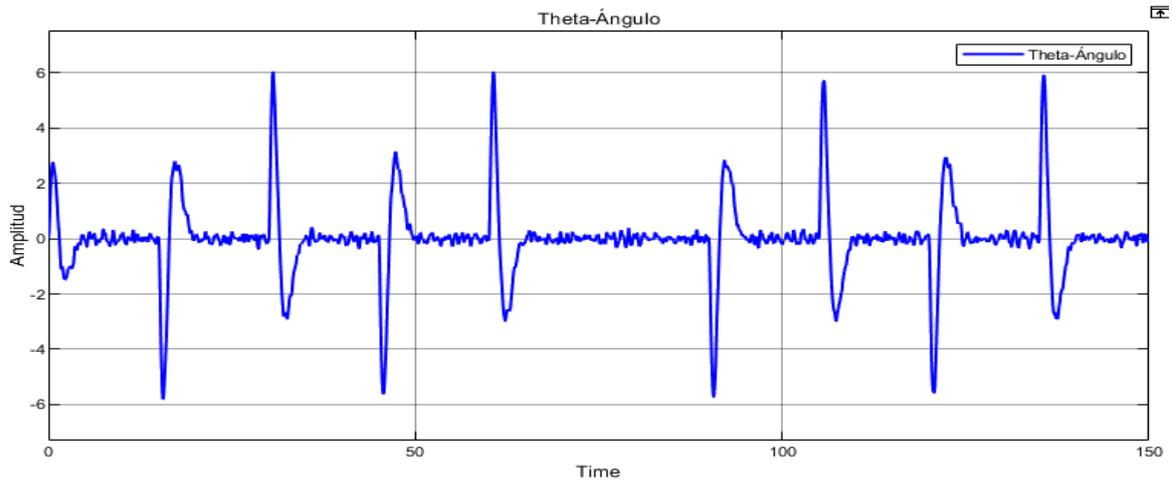


Figura 65 Ángulo del servosistema LQG.

Realizado por: Yuri Torres, 2024.

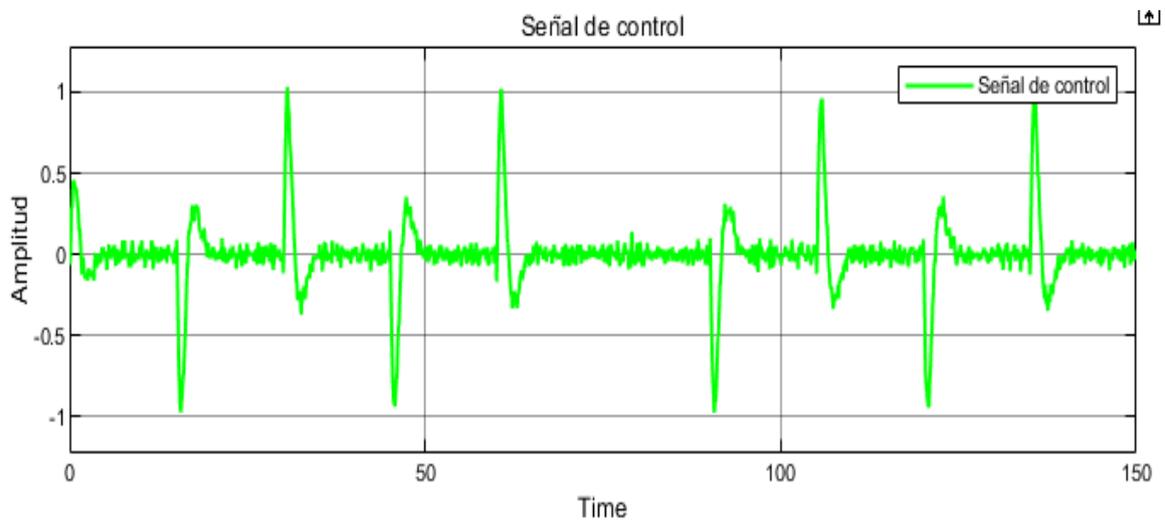


Figura 66 Señal de control del servo LQG.

Realizado por: Yuri Torres, 2024.

Para mejor presentación del trabajo; se implementa el servo controlador LQG separado por bloques específicos para “Kalman-Filter” en Kalman Bucy de estados estacionarios y bloques de ganancias para el controlador LQR. Mediante este principio de separación se conecta los estados estimados del Filtro de Kalman al vector óptimo de control, para evaluar con detalles los parámetros del sistema.

3.5.6 Diseño del servo controlador LQG por bloques separados

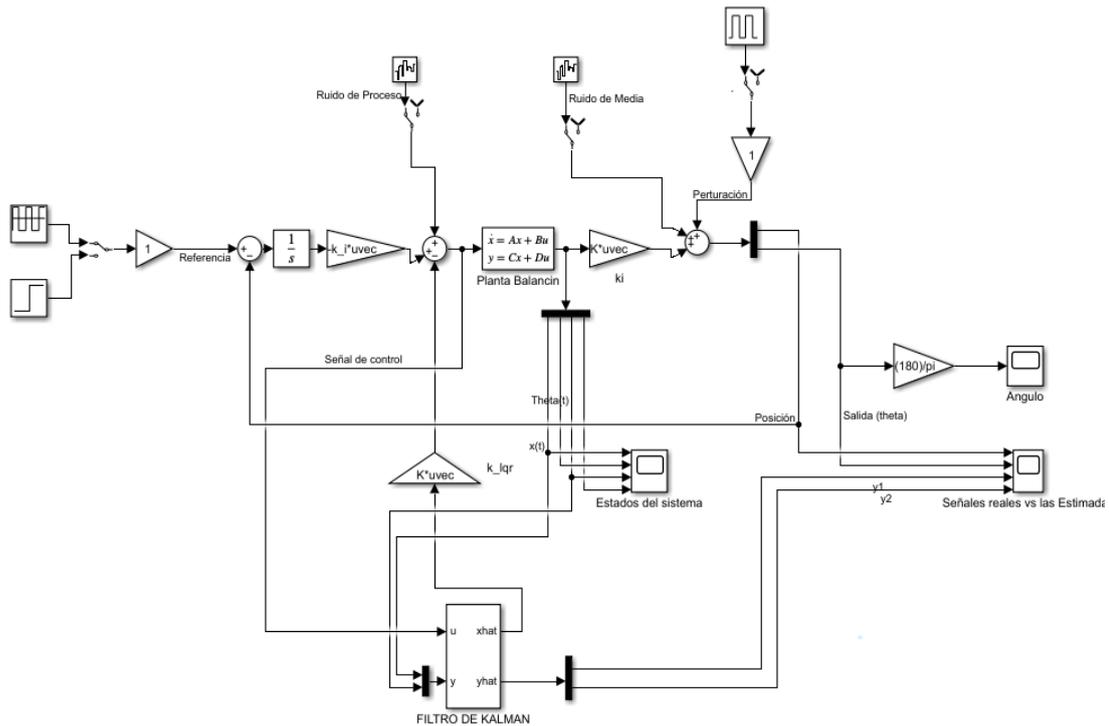


Figura 67 Diseño del servosistema LQG por separado.

Realizado por: Yuri Torres, 2024.

En la figura (67), se implementó el diseño por el principio de separación para cada elemento del controlador LQG. Con este diseño se podrá evaluar detalladamente el desempeño general del sistema mediante el análisis del ruido, perturbaciones, salidas, estados estimados, comparaciones entre salidas estimadas y reales y la señal de control.

La Figura (68) muestra el ruido de media del servosistema con diseño por boques separados.

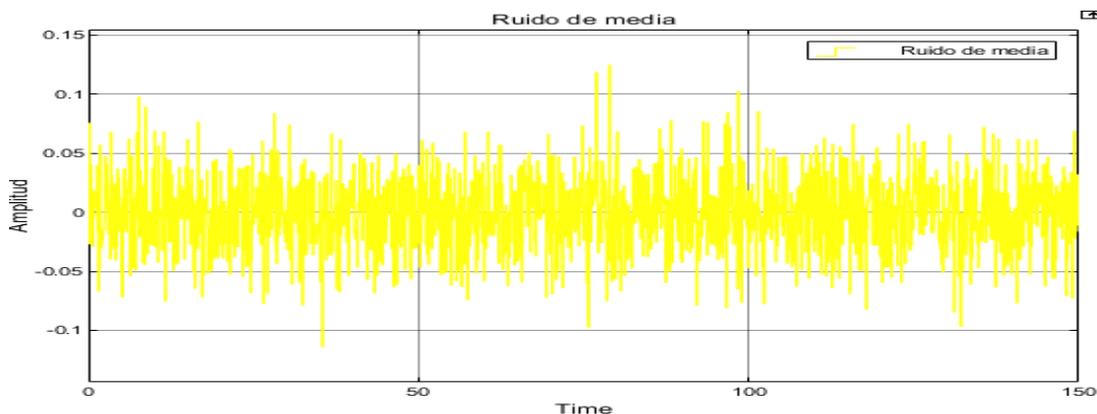


Figura 68 Ruido de Media en el servosistema LQG, por bloques separados.

Realizado por: Yuri Torres, 2024.

$$(PSD) = 1 * 10^{-4} \frac{W}{hz} \quad (182)$$

$$V_{max} = 125 \text{ mV} \quad (183)$$

En la siguiente grafica (69), se analiza el ruido de proceso con distribución Gaussiana.

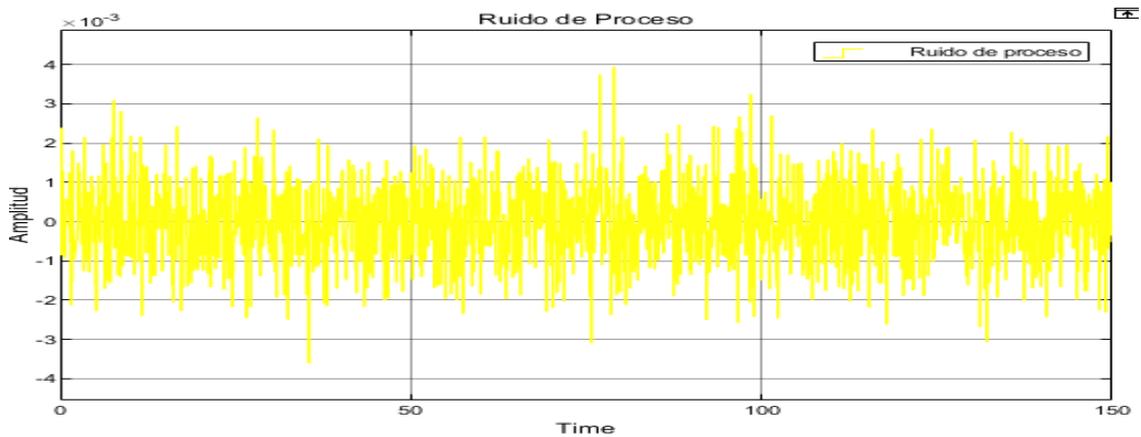


Figura 69 Ruido de Proceso en el servosistema LQG, por bloques separados.

Realizado por: Yuri Torres, 2024.

$$(PSD) = 1 * 10^{-7} \frac{W}{hz} \quad (184)$$

$$V_{max} = 3.95 \text{ mV} \quad (185)$$

Analizando la gráfica (70), se observa la perturbación que ingresa la servosistema por bloques separados que actúa en el servo controlador LQG.

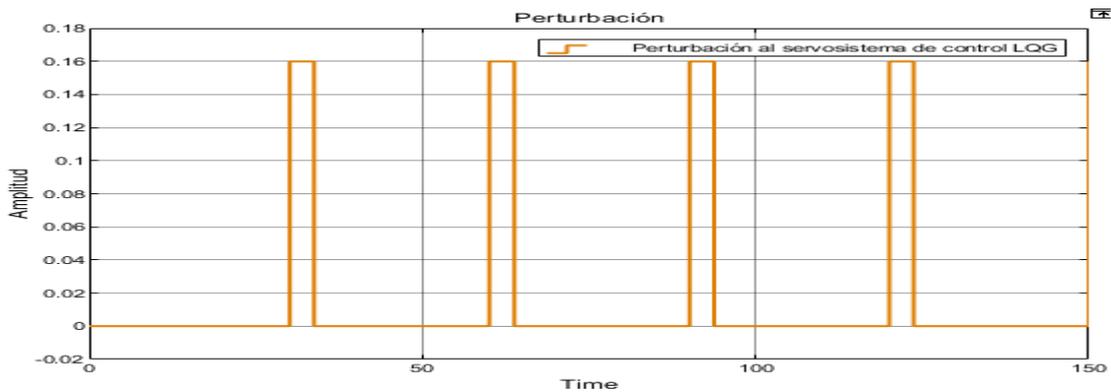


Figura 70 Perturbación al sistema por bloques separados.

Realizado por: Yuri Torres, 2024.

Los estados óptimos estimados, aplicados al sistema controlado con reducción de ruido y una cercanía a los estados reales del sistema con $x = 0$ y $\theta = 0^\circ$. Ver Figura (71).

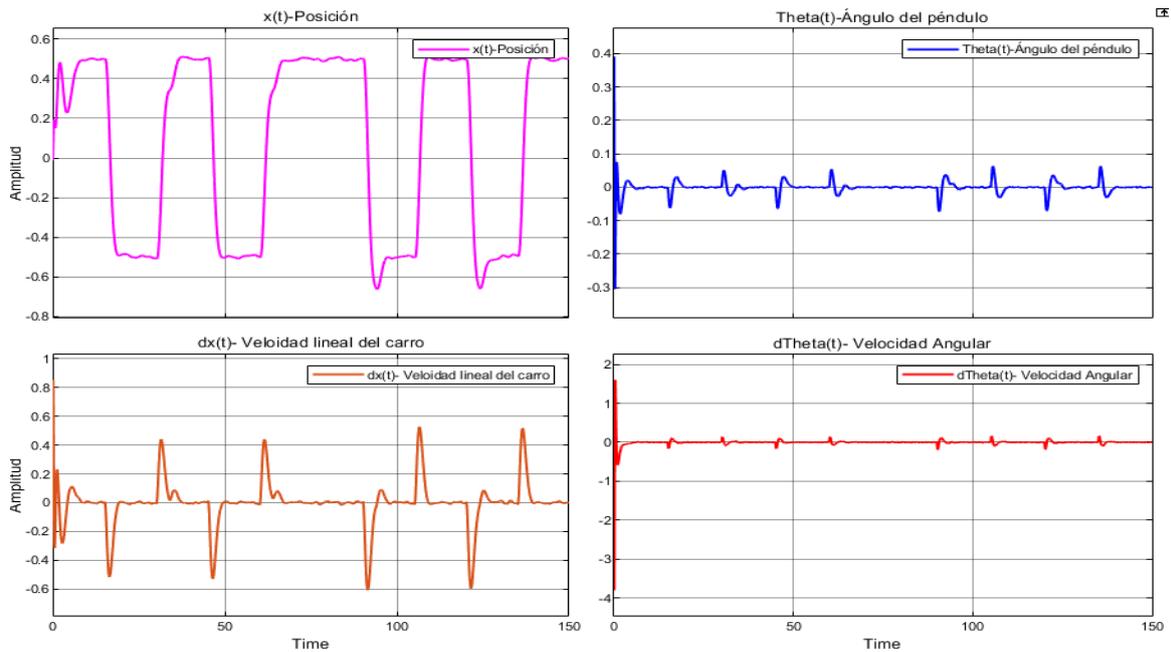


Figura 71 Estados óptimos estimados del servo sistema LQG diseñado por separado.

Realizado por: Yuri Torres, 2024.

En la Figura (72) se presentan las señales de salida reales en comparación con las señales de salidas filtradas y estimadas por el filtro de Kalman denotando carencia de ruido en las salidas estimadas.

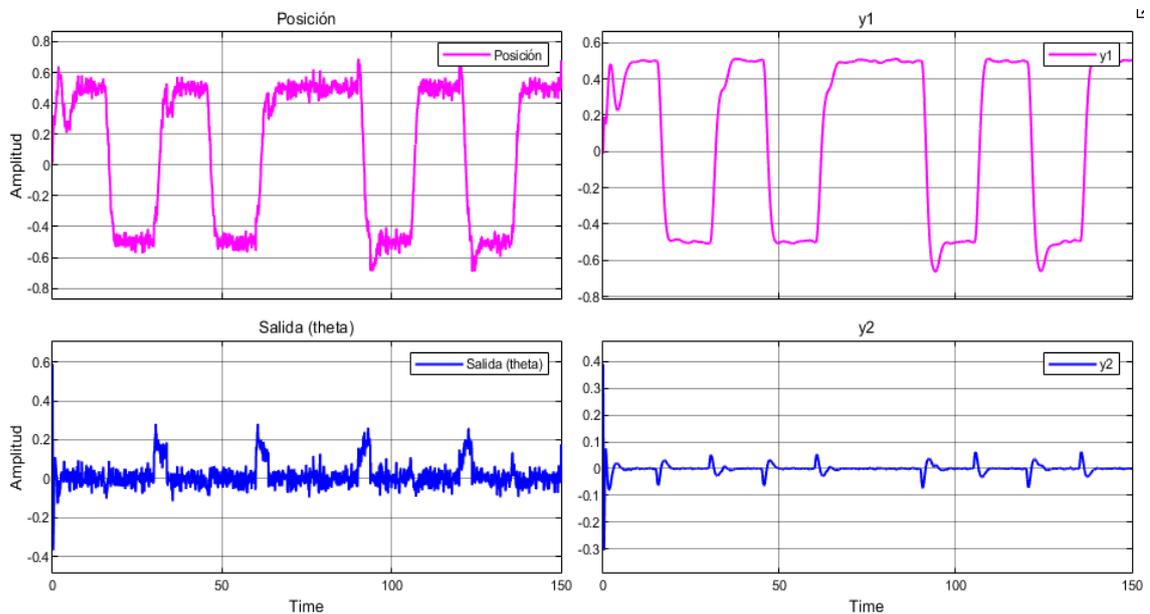


Figura 72 Salidas reales y salidas estimadas óptimas.

Realizado por: Yuri Torres, 2024.

Como se observa en la Figura (73) el servo-sistema de control LQG, elimina la mayoría de ruido y perturbaciones de manera óptima alcanzando la señal de referencia sin mayores inconvenientes sin tantos sobre impulsos, gracias a los aportes del servo controlador.

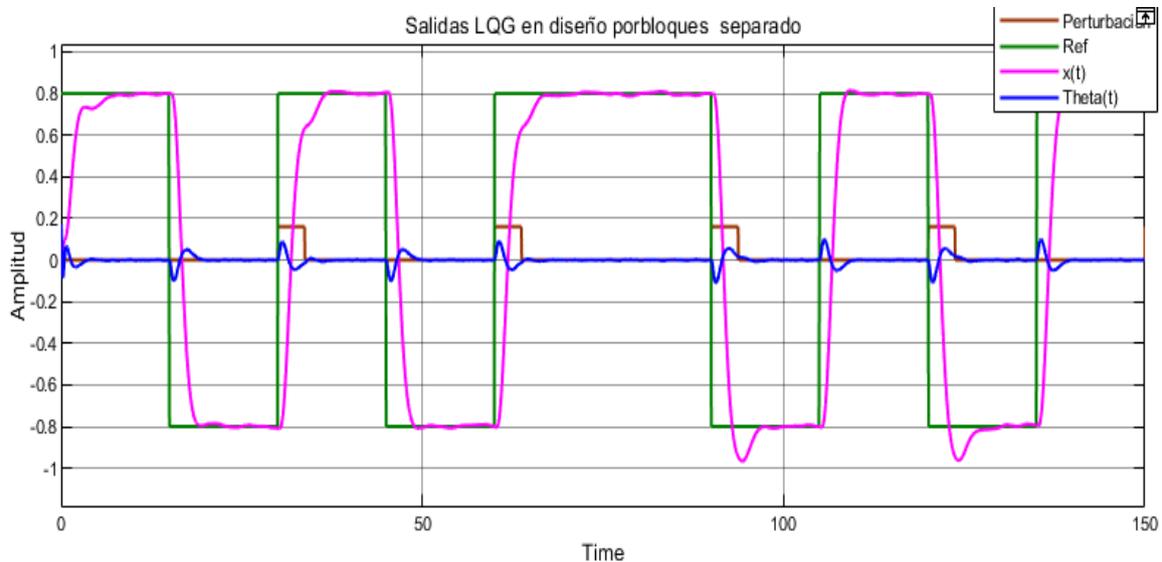


Figura 73 Salidas LQG en bloques separados.

Realizado por: Yuri Torres, 2024.

3.5.6.1 Respuesta del tiempo de estabilización (t_{ss})

La Figura (74), muestra la gráfica de MATLAB© - Simulink con seguimiento a la referencia para el tiempo de estabilización en $t_{ss} = 3.715 s$ en un 2%. Cabe indicar que las condiciones iniciales son $x = 0$ y $\theta = 0^\circ$.

$$y_{max} = 0.50 + 0.02(0.49) = 0.509 \quad (186)$$

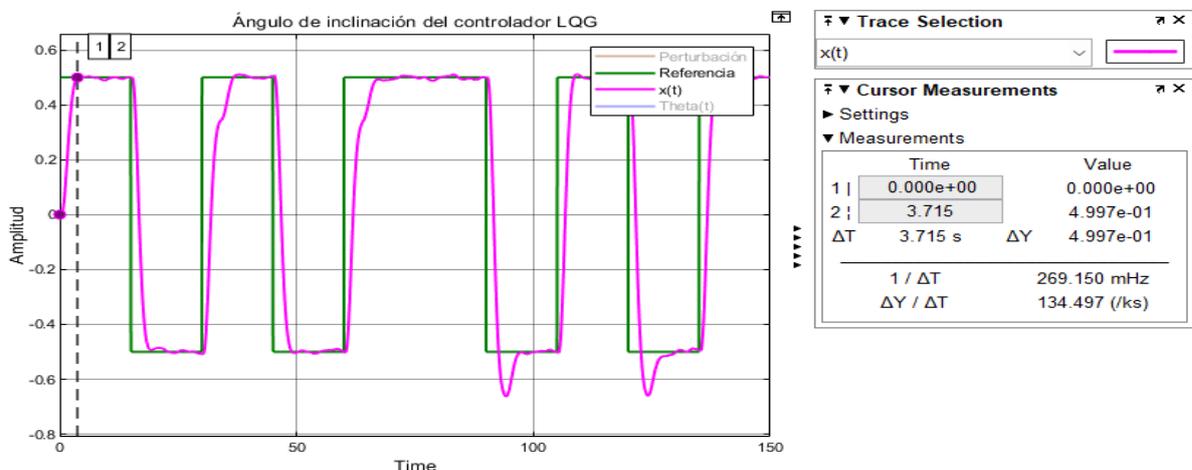


Figura 74 Tiempo de estabilización (t_{ss}) del control LQG.

Realizado por: Yuri Torres, 2024.

3.5.6.2 Respuesta del sobre impulso máximo (M_p)

En la figura (75), se muestra la gráfica para el sobre impulso versus la referencia.

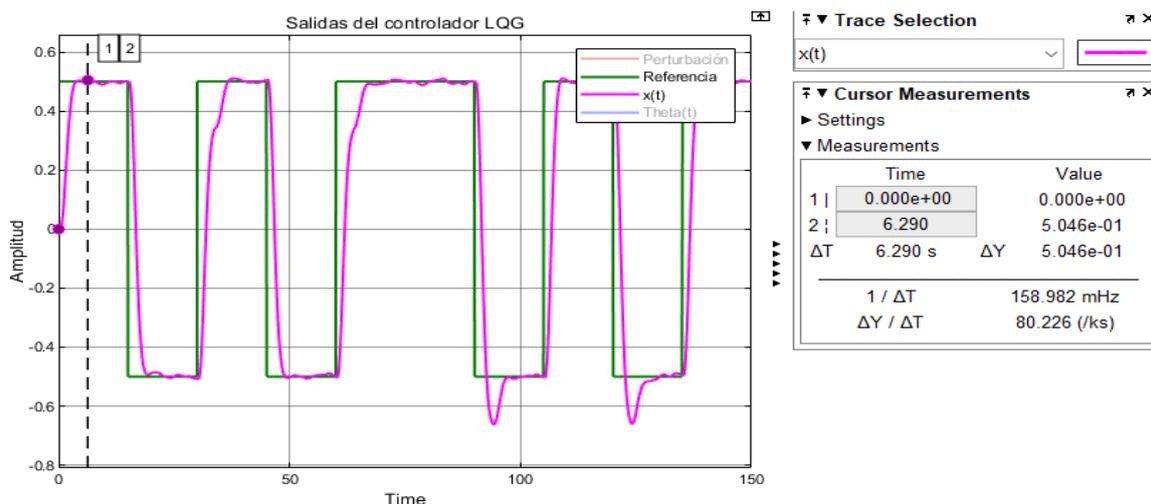


Figura 75 Máximo sobre impulso (M_p) en LQG.

Realizado por: Yuri Torres, 2024.

Para hallar el (M_p) se procede a la ecuación siguiente:

$$M_p = \frac{0.5046 - 0.50}{0.50} = 0.0092 \quad (187)$$

$$M_p = \mathbf{0.92\%} \quad (188)$$

3.5.6.3 Respuesta del ángulo de inclinación (θ)

En la Figura (76), se muestra el ángulo controlado desde su instante inicial 0° , tal como muestra la imagen, después llega a mínimos de -0.05143° para luego estabilizarse.

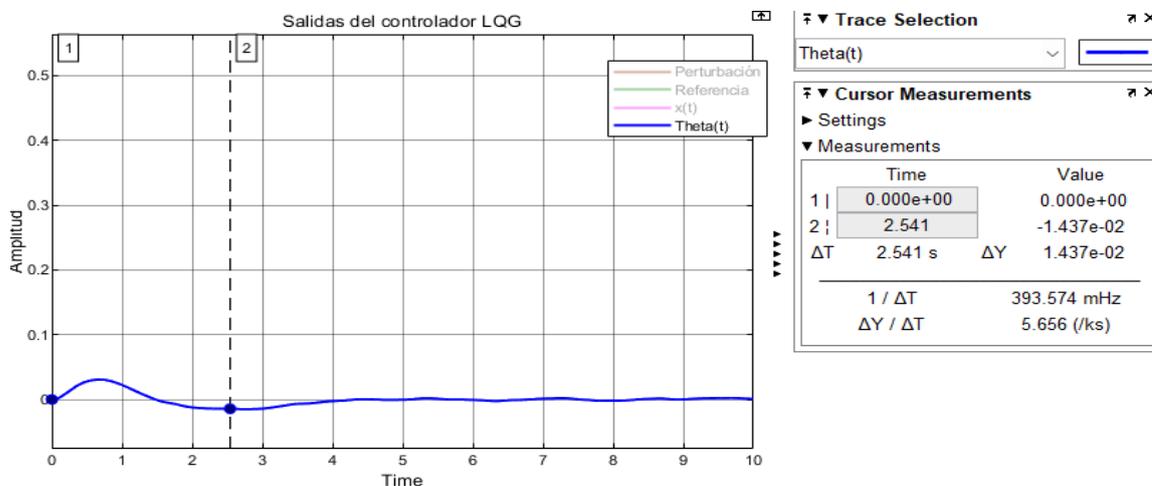


Figura 76 Ángulo de inclinación en el control LQG.

Realizado por: Yuri Torres, 2024.

En la imagen (77) se muestra el intervalo de valores en radianes del ángulo estabilizado.

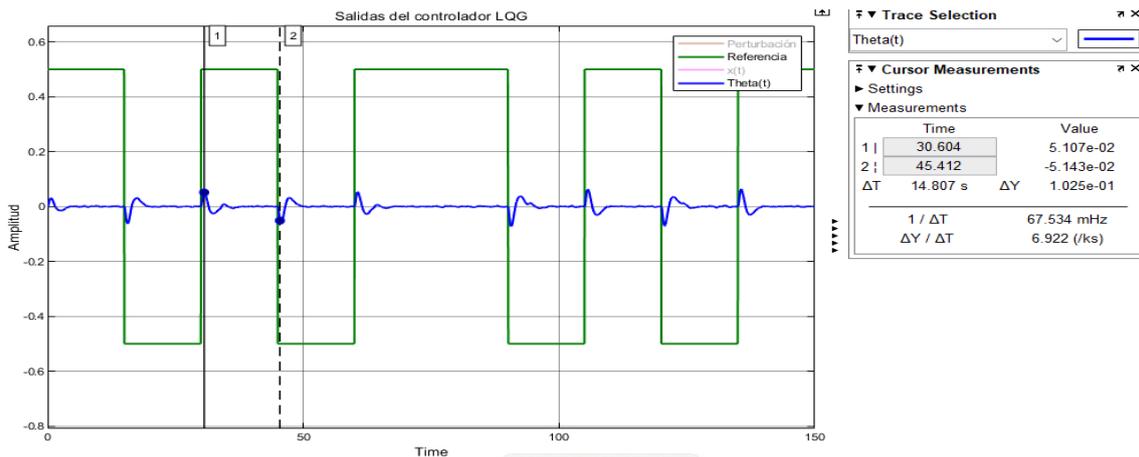


Figura 77 Regulación del ángulo en LQG.

Realizado por: Yuri Torres, 2024.

$$-0.05143^\circ \leq \theta \leq 0.05107^\circ$$

➤ **Valores Máximos y mínimos del ángulo de inclinación.**

Los valores máximos y mínimos del ángulo de inclinación son:

$$\theta_{max} = 0.05107^\circ \text{ y } \theta_{min} = -0.05143^\circ$$

3.5.6.4 Respuesta de la señal de control

En la Figura (78) y (79), se aprecia el efecto del servo-controlador LQG, desde sus valores iniciales de 0 V, valor máximo 0.2925 V, valor mínimo -0.03567 V y resaltan notables los cambios en la evolución de la señal de control.

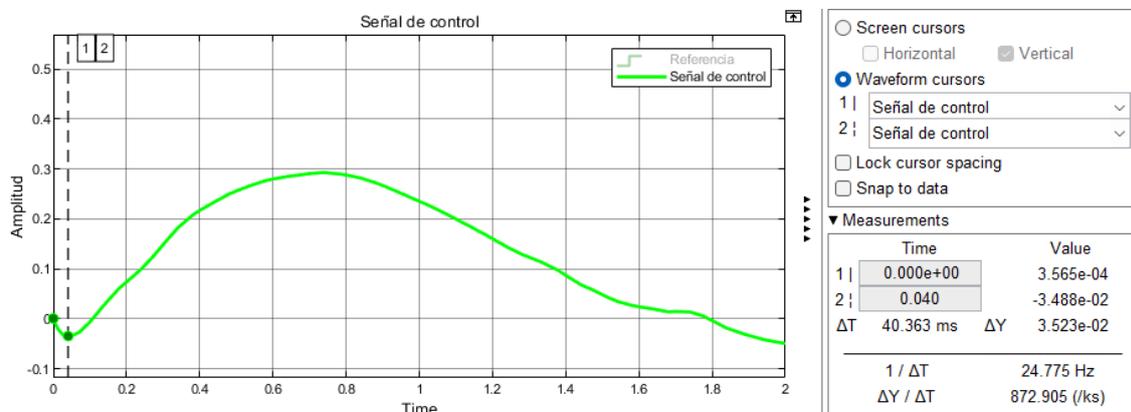


Figura 78 Señal de control (u) en el servo controlador LQG diseño de bloques separados.

Realizado por: Yuri Torres, 2024.

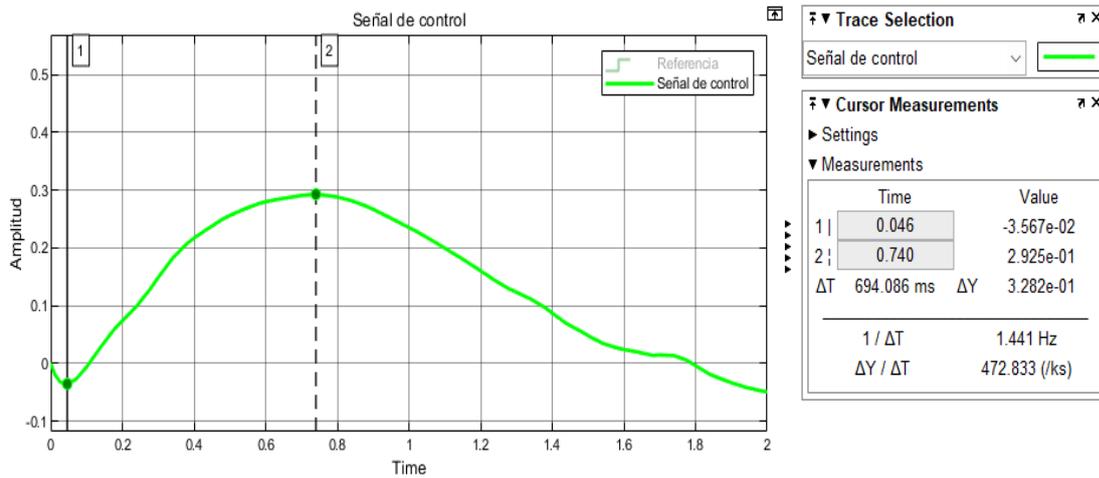


Figura 79 Señal de control con sus valores máximo y mínimo para el control LQG.

Realizado por: Yuri Torres, 2024.

➤ **Intervalos de la señal de energía de control del servo sistema LQG.**

En la figura (80) se muestra como la señal de control cambia en los rangos de valores de entrada a la referencia:

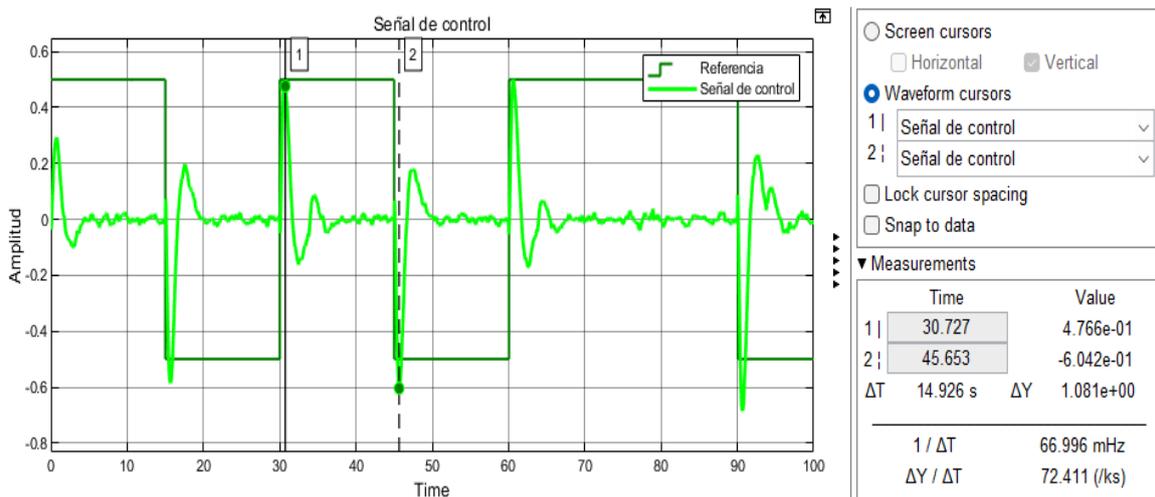


Figura 80 Intervalos de la señal de control para el control LQG.

Realizado por: Yuri Torres, 2024.

$$-0.6042 \leq u(t) \leq 0.4766$$

3.7 Diseño del controlador LQG en Arduino IDE

En este paso se resume la implementación del código en el software Arduino IDE, para programar el microcontrolador Arduino Uno y cumplir con la comunicación de la

planta robótica. Ver Anexo (10) donde está el detalle de la codificación con las librerías creadas por el autor:

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

#include "Motor.h"
#include "LQG.h"
Temporizador Temporizador;
extern Mpu6050 Mpu6050;
extern Motor Motor;
extern LQR LQR;
LQR::LQR(){

/*    theta, theta_dot, x_dot, x      ,    omega    , omega_dot*/
    K[0] = 67, K[1] = 0.7, 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();
    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[0]);
    }
}
```

3.9 Comunicación Arduino IDE - Robot balancín

Los pasos de comunicación son:

- Conectar la placa Arduino Uno a la computadora mediante cable USB.
- Abrir el Arduino IDE.

- Verificar el código y dar solución en caso de errores.
- Supervisar la compilación que no existan errores.
- Cargar el código
- Desconectar la placa Arduino y verificar el proceso implementado en la planta robótica de forma inalámbrica.



Realizado por: Yuri Torres, 2024.

Figura 81 Carga del código al microcontrolador.

3.8 Comunicación del módulo Bluetooth - Robot balancín

La Figura (82), muestra la comunicación inalámbrica de la planta con el módulo HC-05 permitiendo obtener la información de los encoder de las llantas del robot.

En la figura (85), se muestra el MATLAB© - Simulink, diseñado para la recepción de los datos, en este caso el ángulo inclinación de la Figura (83) y el giro en X Y Z de la aceleración angular del robot como indica la Figura (84).



Figura 82 Conexión inalámbrica de la planta.

Realizado por: Yuri Torres, 2024.

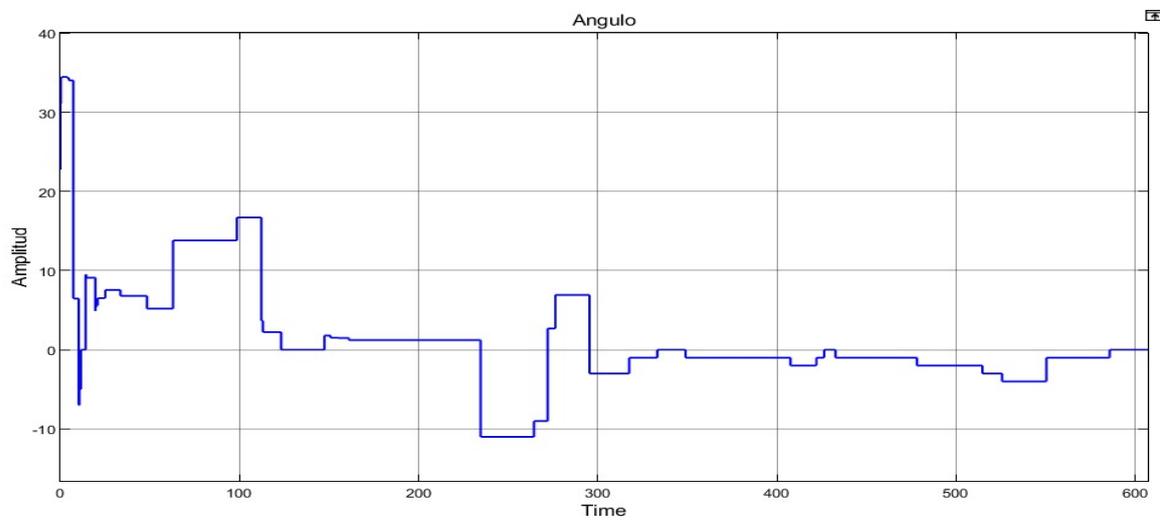


Figura 83 Ángulo del robot en tiempo real.

Realizado por: Yuri Torres, 2024.

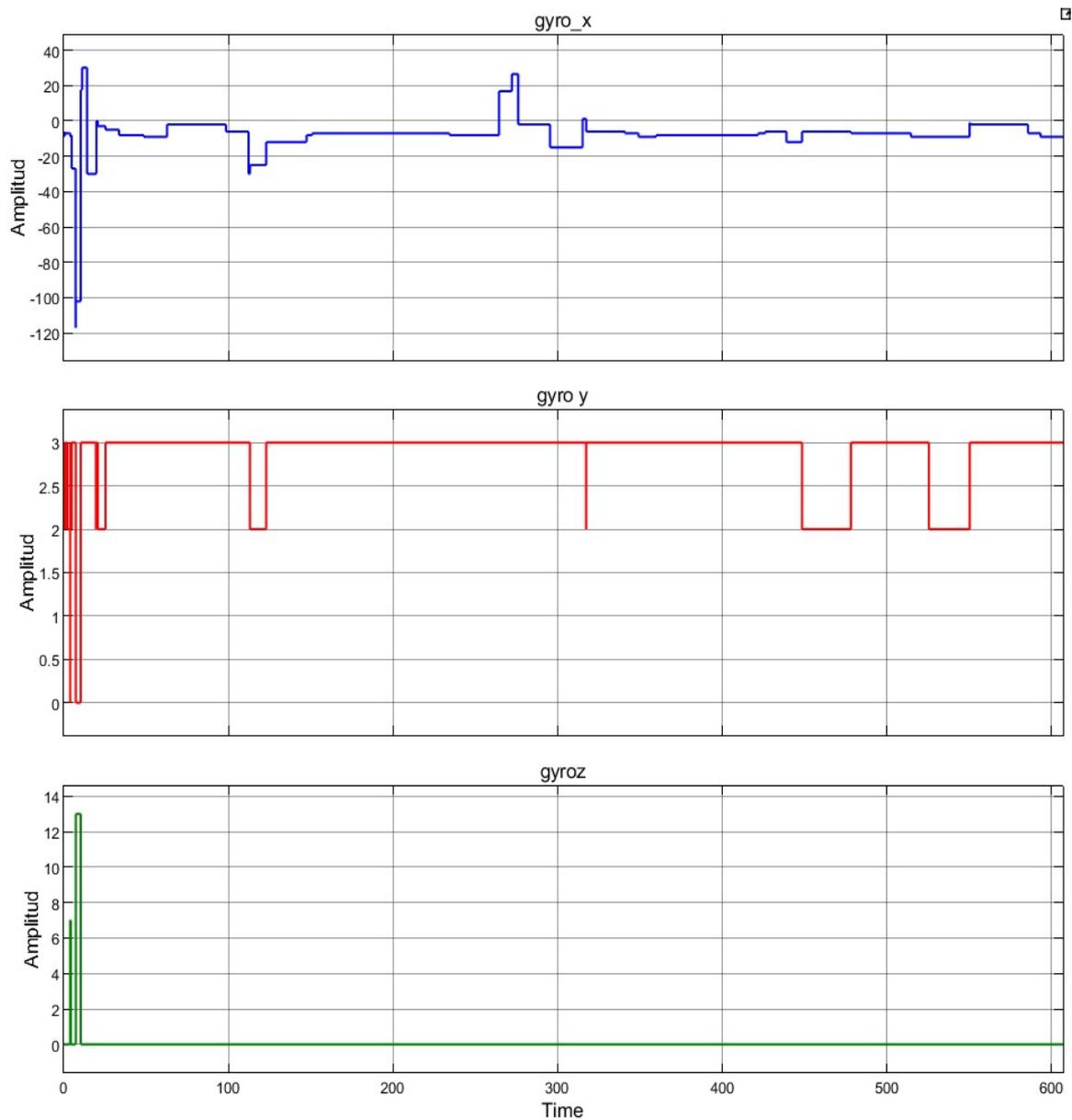


Figura 84 Aceleración angular del robot en tiempo real.

En el Anexo (6), se presenta el desarrollo del código en Arduino, el cual ha sido implementado al código del diseño del servo controlador LQG. Cabe indicar que la comunicación en tiempo real ayuda a obtener los datos remoto del robot, mediante el pin (RX) de recepción mientras el PWM controla la velocidad y dirección de los motores mediante la modulación del ancho de pulso. Ver Figura (85). Cabe resaltar que la señal PWM proporciona las directrices específicas sobre la velocidad y dirección del robot para mantener el equilibrio. En la misma figura también se observan los estados de la planta: posición, velocidad lineal, ángulo, y variación del ángulo.

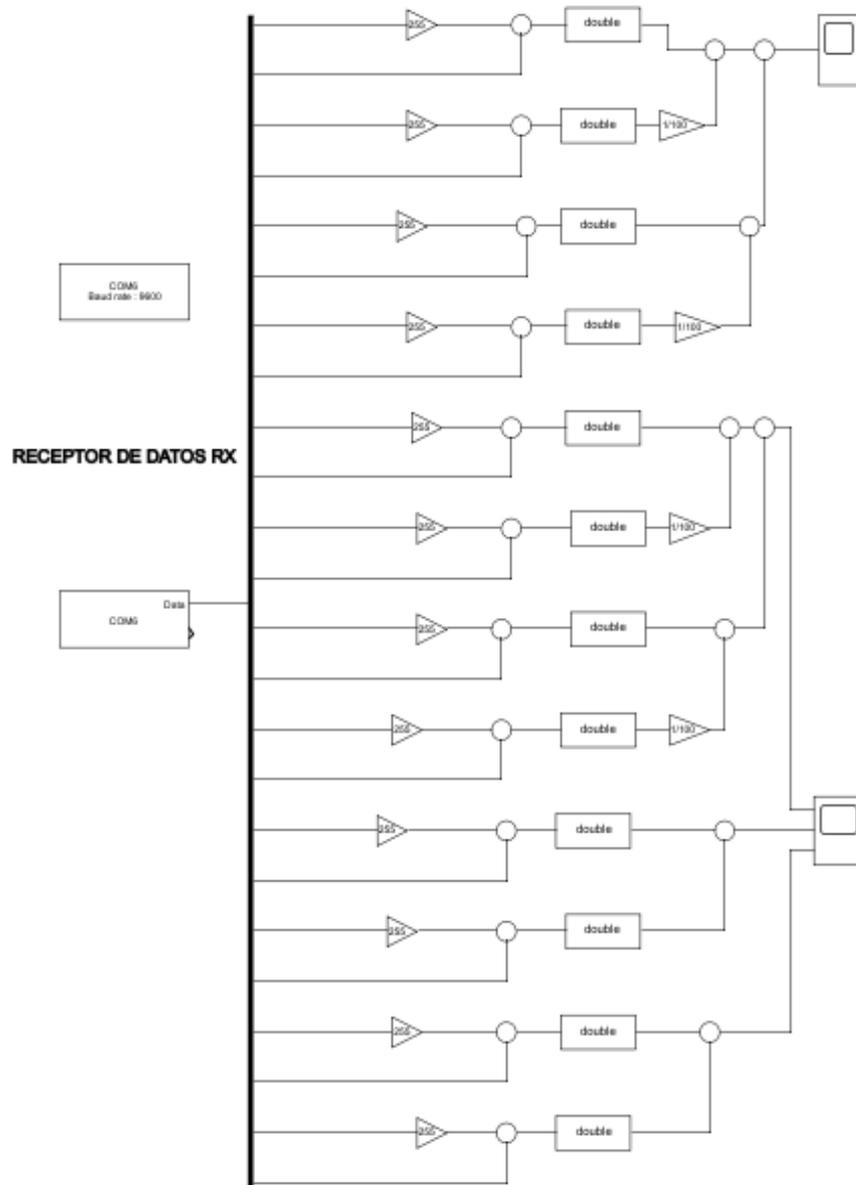


Figura 85 Comunicación Bluetooth.

Realizado por Yuri Torres, 2024.

CAPÍTULO IV

4. EVALUACIONES Y RESULTADOS

Este capítulo, presenta evaluaciones numéricas y comparativas del control LQR y LQG con análisis en diversas condiciones iniciales.

4.1. Evaluaciones

El estudio se realiza con las evaluaciones analíticas del entorno gráfico proporcionado por el software MATLAB© - Simulink para ejecución y análisis en tiempo real del proceso. Se utilizaron diferentes bloques, como "Scope" y "Display", para visualizar y examinar las señales medidas. Herramientas gráficas que permiten una representación más intuitiva y efectiva de los resultados obtenidos durante las evaluaciones, que facilitan la comprensión y valoración del comportamiento del controlador LQR perturbado y el controlador LQG. También se considera el control LQG con diferentes condiciones iniciales para respectivas evaluaciones

4.1.1 Comparativa del control LQR vs LQG

El servo-sistema de control por bloques separados LQG implementado en el robot balancín demuestra ser el mejor control al compararse con el control LQR especificando los siguientes parámetros:

- Tiempo de estabilización (t_{ss}).
- Máximo sobre impulso (M_p).
- Error de estado estacionario (e_{ss}).

También se considera comparar la variable de control y variable a controlar:

- Ángulo máximo del péndulo invertido (θ_{max}).
- Señal de control máxima (u_{max}).

4.1.1.1 Desempeño del sistema de control LQG

En la Tabla 4, se especifican los parámetros hallados mediante la simulación respectiva.

Tabla 4 Especificaciones del control LQG.

Especificaciones	Parámetros del sistema
Tiempo de estabilización (criterio 2%)	$t_{ss} = 3.71 \text{ s}$
Sobre impulso máximo	$M_p = 0.92\%$
Ángulo de inclinación máximo	$\theta_{max} = 0.05107^\circ$
Ángulo de inclinación mínimo	$-\theta_{min} = -0.05143^\circ$
Valores máximos de la señal de control	$u_{max} = 0.2925 \text{ V}$
Valores mínimos de la señal de control	$u_{min} = -0.0357 \text{ V}$
Rango de valores de la señal de control.	$-0.6042 \text{ V} \leq u(t) \leq 0.4766 \text{ V}$
Error de estado estacionario	$e_{ss} = 0.00813$

Realizado por: Yuri Torres, 2024.

4.1.1.2 Desempeño del sistema de control LQR

En la Tabla 5, se muestran los datos numéricos del controlador LQR detallados en el capítulo III.

Tabla 5 Especificaciones del control LQR.

Especificaciones	Parámetros del sistema
Tiempo de estabilización (criterio 2%)	$t_{ss} = 4.360 \text{ s}$
Sobre impulso máximo	$M_p = 13.22\%$
Ángulo de inclinación máximo	$\theta_{max} = 0.1256^\circ$
Ángulo de inclinación mínimo	$\theta_{min} = -0.06146^\circ$
Valores máximos de la señal de control	$u_{max} = 0.2271 \text{ V}$
Valores mínimos de la señal de control	$u_{min} = -0.0818 \text{ V}$
Rango de valores de la señal de control	$-0.5785 \leq u(t) \leq 0.5024$
Error de estado estacionario	$e_{ss}=0.03350$

Realizado por: Yuri Torres, 2024.

El ruido blanco gaussiano y las perturbaciones, que se mencionan en la tabla 6, fueron utilizadas en todas las evaluaciones del controlador.

Tabla 6 Especificaciones del Ruido Gaussiano blanco y perturbaciones.

	Ruido Blanco Gaussiano (Ruido de Proceso)	Ruido Blanco Gaussiano (Ruido de Media)

Densidad Espectral de Potencia (PSD).	$1 * 10^{-7} \left[\frac{W}{Hz} \right]$	$1 * 10^{-4} \left[\frac{W}{Hz} \right]$
Perturbación		
$F = 0.16, T = 5s$		

Realizado por: Yuri Torres, 2024.

4.1.2 Evaluaciones del servo-control LQG

En la tabla (7), se muestran los parámetros del servo-sistema LQG para el respectivo análisis de comparación en diferentes aspectos.

Tabla 7 Evaluaciones y comparaciones del control LQG

Servo-control LQG	t_{ss}	M_p	θ_{max}	e_{ss}	u_{max}	Referencia	Condiciones iniciales: (x, θ)
Evaluación 1	7.956 s	0.225%	5.051°	0.00813	3.396 V	0.80m	(0, 10°)
Evaluación 2	9.076 s	1.162%	27.746°	0.00160	3.579 V	0.80.m	(3, 30°)
Evaluación 3	3.715 s	0.92%	0.0510°	0.00386	0.292V	0.50m	(0, 0°)
Evaluación 4	9.584 s	0.3%	2.927°	0.00726	0.128V	0.50m	(10, 15°)

Realizado por: Yuri Torres, 2024.

4.1.2.1 Análisis de Evaluación 1: $x = 0$ y $\theta = 10^\circ$

En las gráficas (86)-(90) se comprueban los análisis comparativos de la tabla (7), donde se constatan los resultados de la simulación en MATLAB© - Simulink.

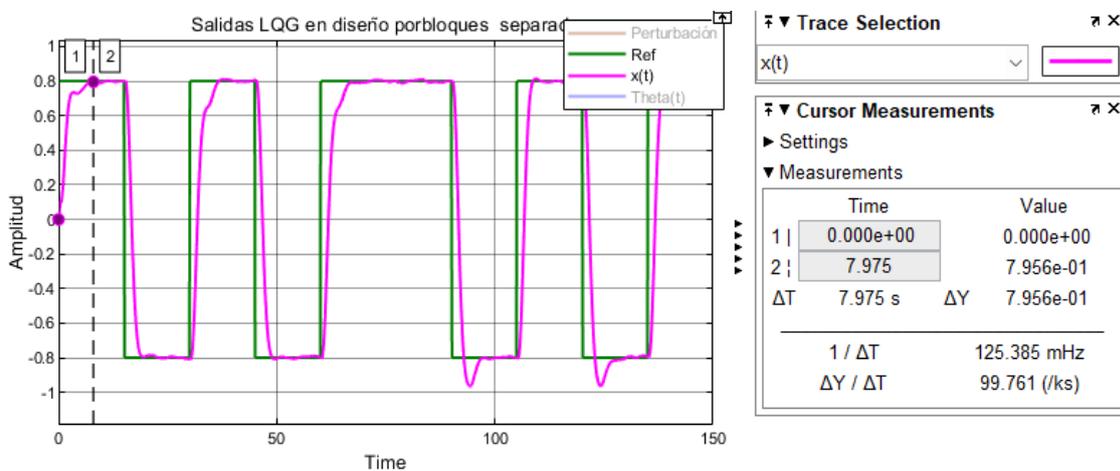


Figura 86 Tiempo de estabilización LQG para Evaluación 1.

Realizado por: Yuri Torres, 2024.

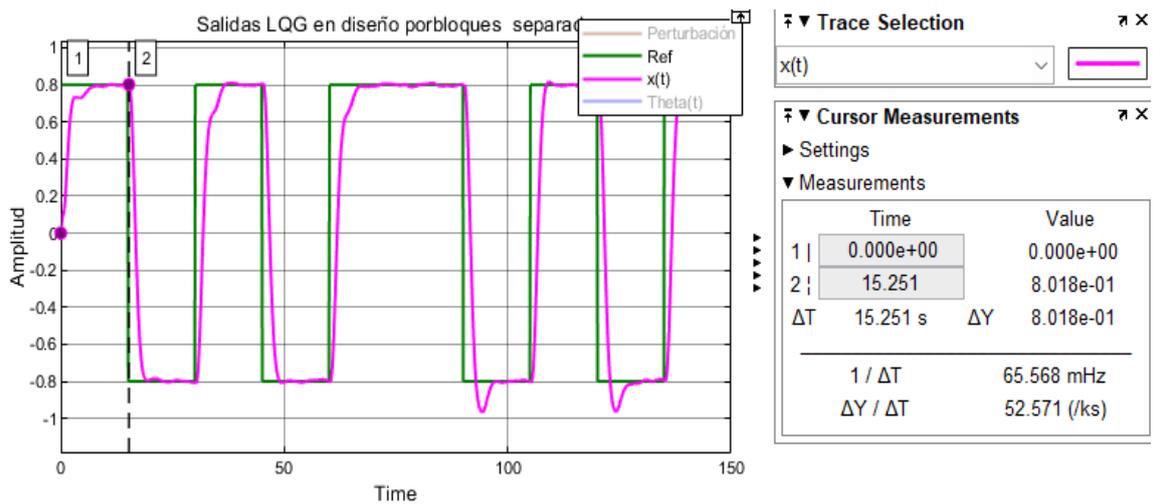


Figura 87 Sobre impulso máximo (M_p) en controlador LQG para Evaluación 1.

Realizado por: Yuri Torres, 2024.

$$M_p = \frac{0.8018 - 0.80}{0.80} = 0.00225 \quad (189)$$

$$M_p = 0.225\% \quad (190)$$

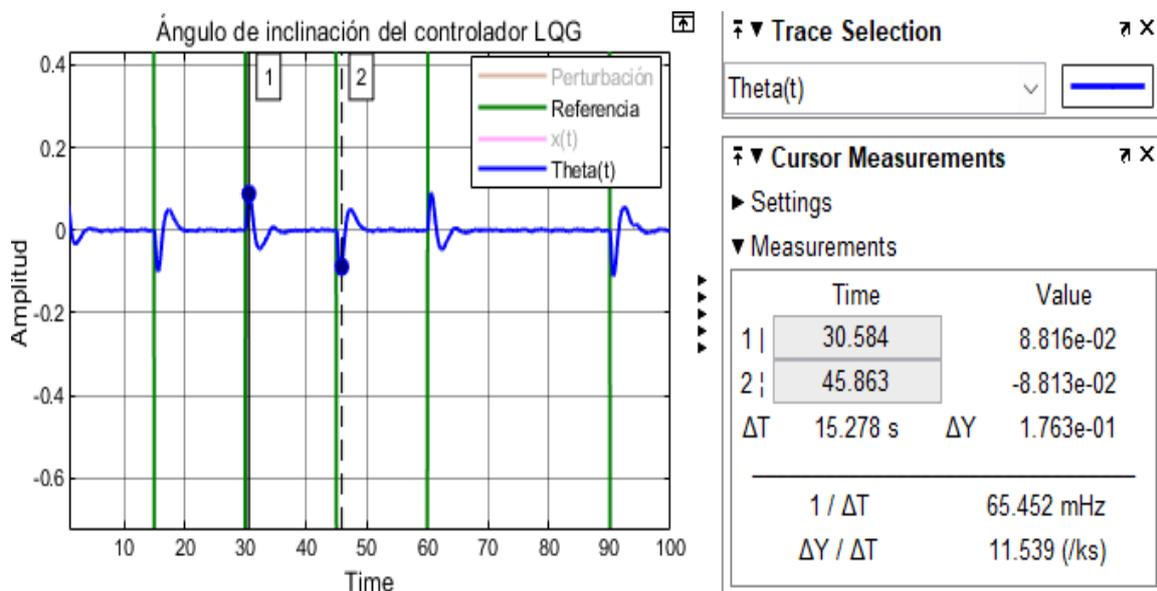


Figura 88 Ángulo de inclinación en el controlador LQG en la Evaluación 1.

Realizado por: Yuri Torres, 2024.

$$-0.08813 \text{ rad} \leq \theta \leq 0.08816 \text{ rad}$$

$$-5.049^\circ \leq \theta \leq 5.051^\circ$$

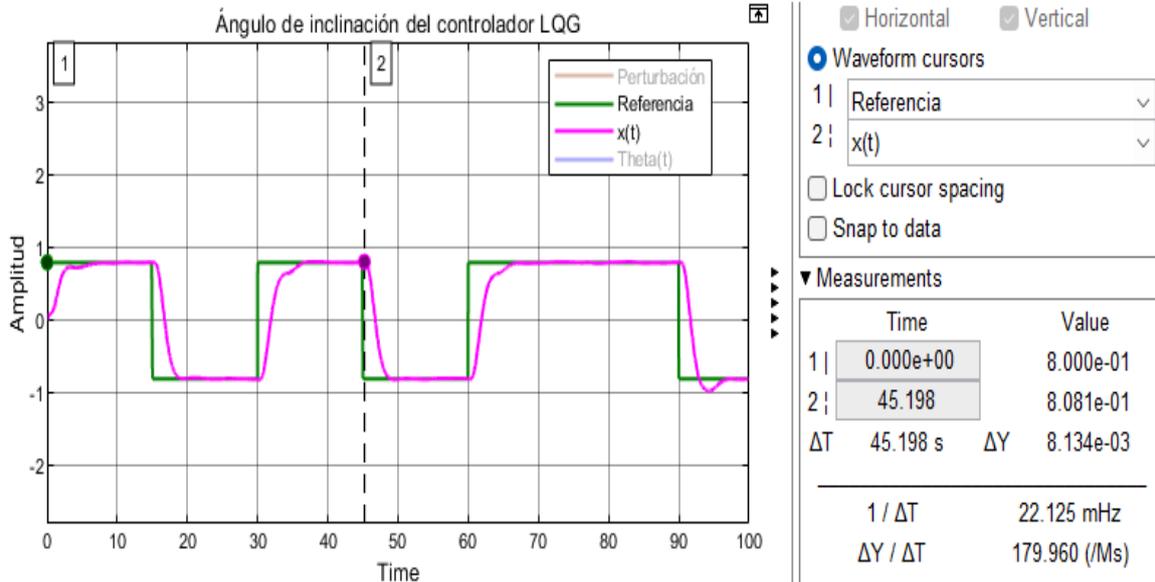


Figura 89 Error de estado estacionario (e_{ss}) en controlador LQG para Evaluación 1.

Realizado por: Yuri Torres, 2024.

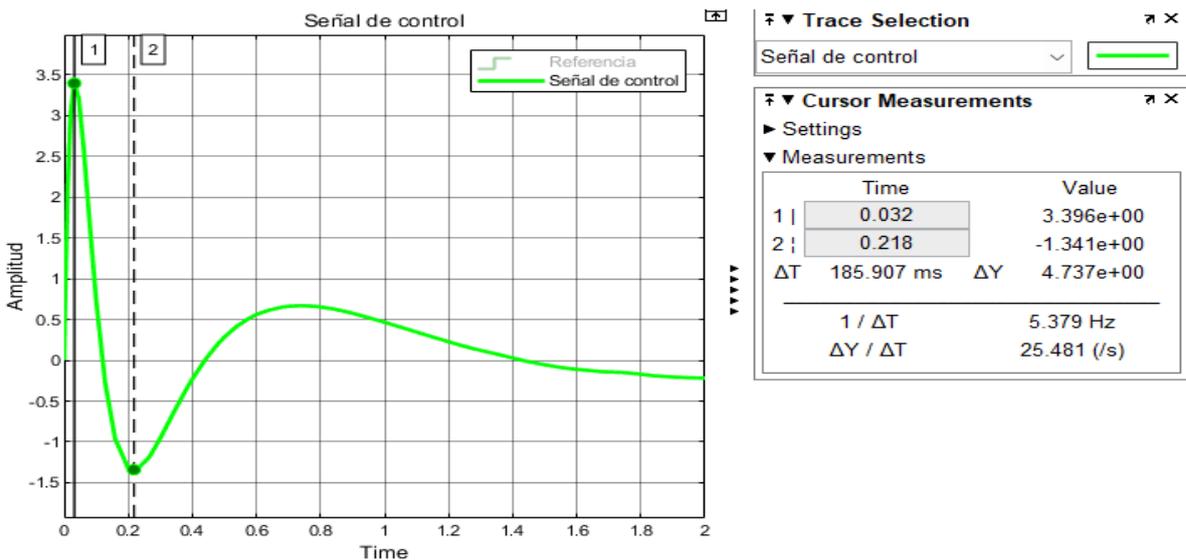


Figura 90 Señal de control en controlador LQG para Evaluación 1.

Realizado por: Yuri Torres, 2024.

4.1.2.2 Análisis de Evaluación 2: $x = 3$ y $\theta = 30^\circ$

En las gráficas (91)-(95) se justifican los datos de evaluación 2, en la tabla (7).

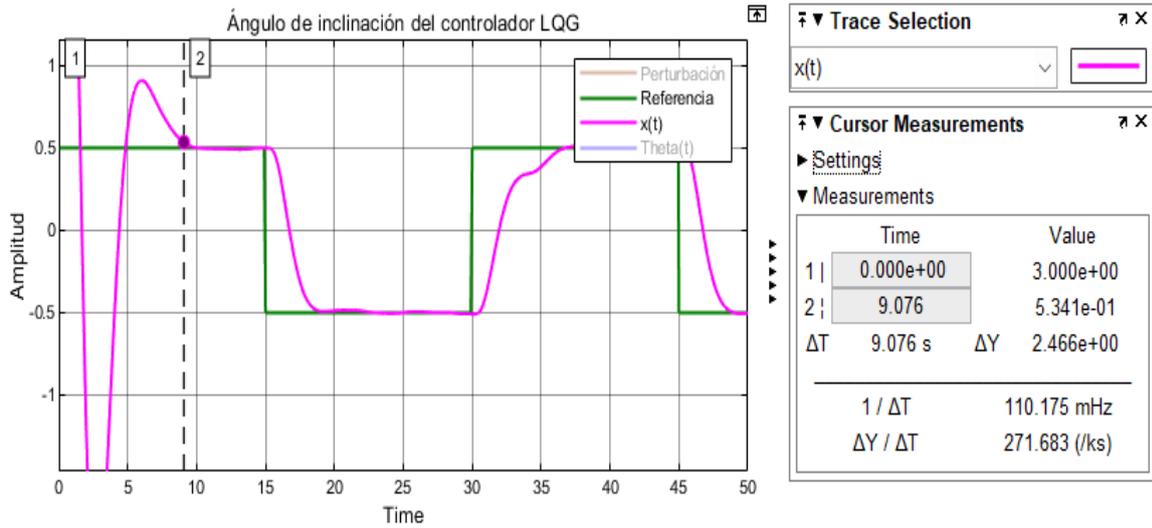


Figura 91 Tiempo de estabilización (t_{ss}) en controlador LQG para Evaluación 2.

Realizado por: Yuri Torres, 2024.

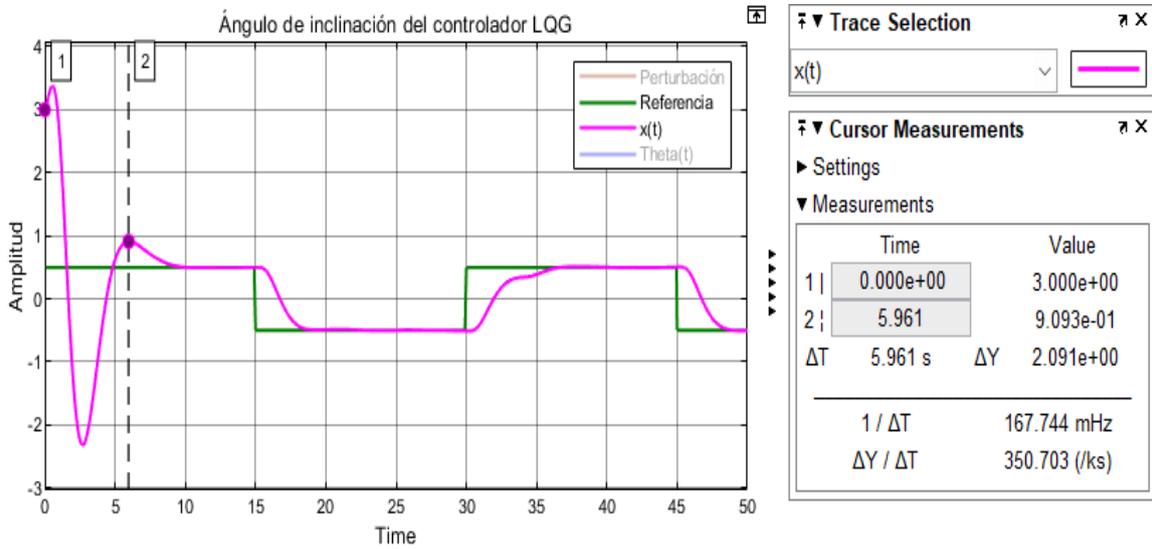


Figura 92 Sobre impulso máximo (M_p) en controlador LQG para Evaluación 2.

Realizado por: Yuri Torres, 2024.

$$M_p = \frac{0.8093 - 0.80}{0.80} = 0.011625 \quad (191)$$

$$M_p = 1.1625\% \quad (192)$$

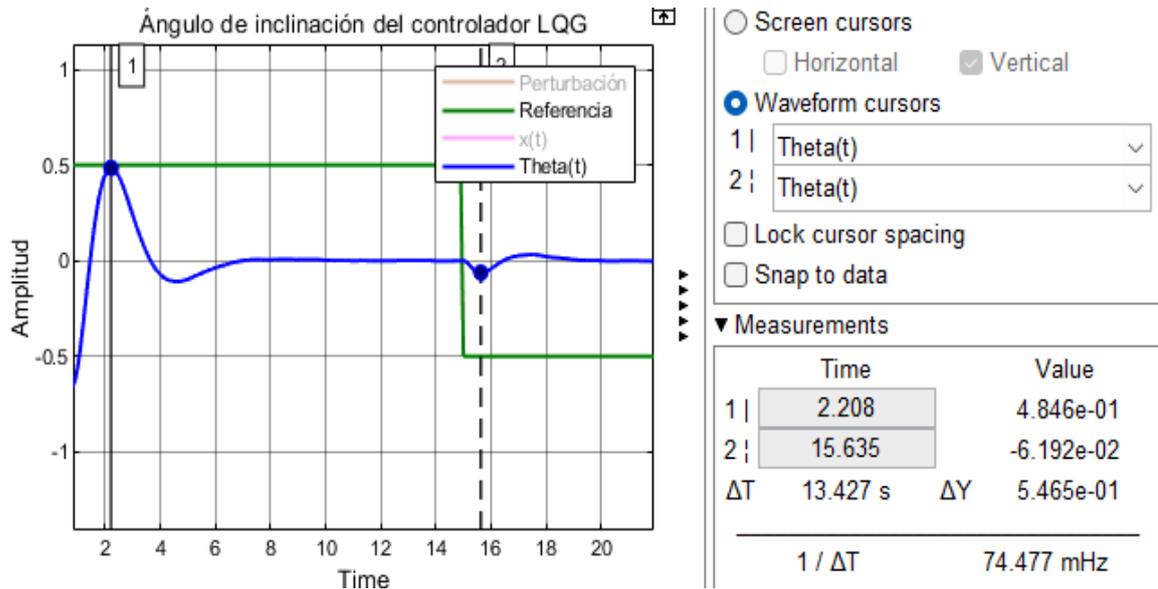


Figura 93 Ángulo de inclinación en el controlador LQG en la Evaluación 2.

Realizado por: Yuri Torres, 2024.

$$-0.05324 \text{ rad} \leq \theta \leq 0.4846 \text{ rad}$$

$$-3.0553^\circ \leq \theta \leq 27.746^\circ$$

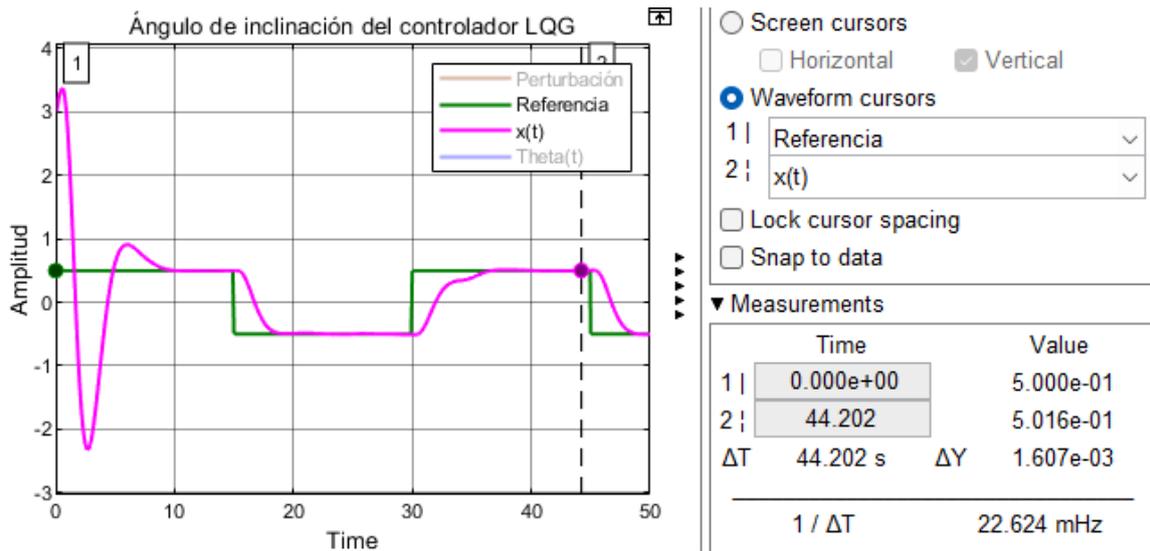


Figura 94 Error de estado estacionario (e_{ss}) en controlador LQG para Evaluación 2.

Realizado por: Yuri Torres, 2024.

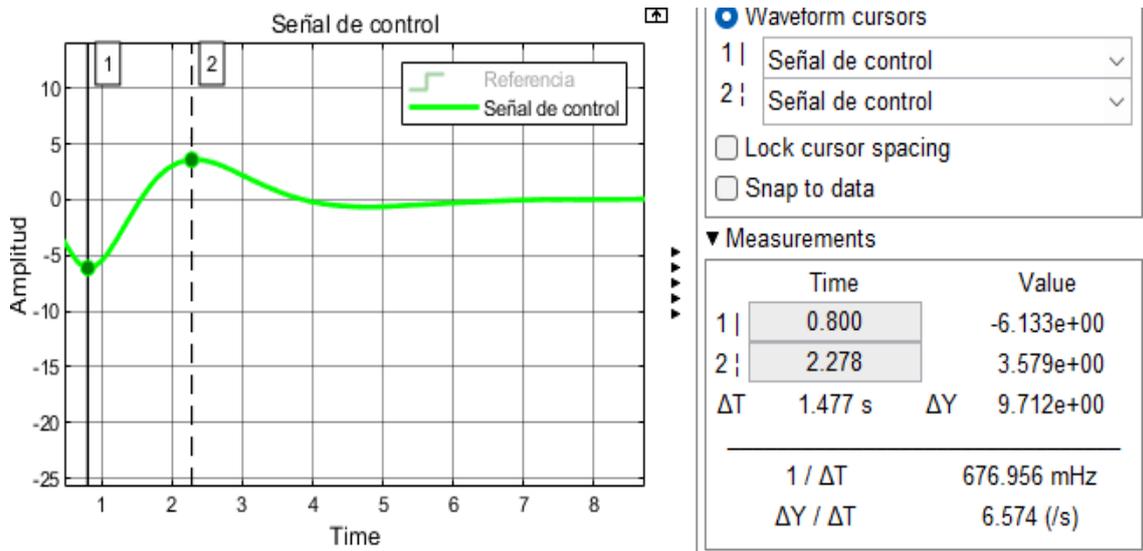


Figura 95 Señal de control en controlador LQG para Evaluación 2

Realizado por Yuri Torres, 2024.

4.1.2.3 Análisis de Evaluación 3: $x = 0$ y $\theta = 0^\circ$

En las gráficas (96)-(100) se justifican los datos de evaluación 3, en la tabla (7).

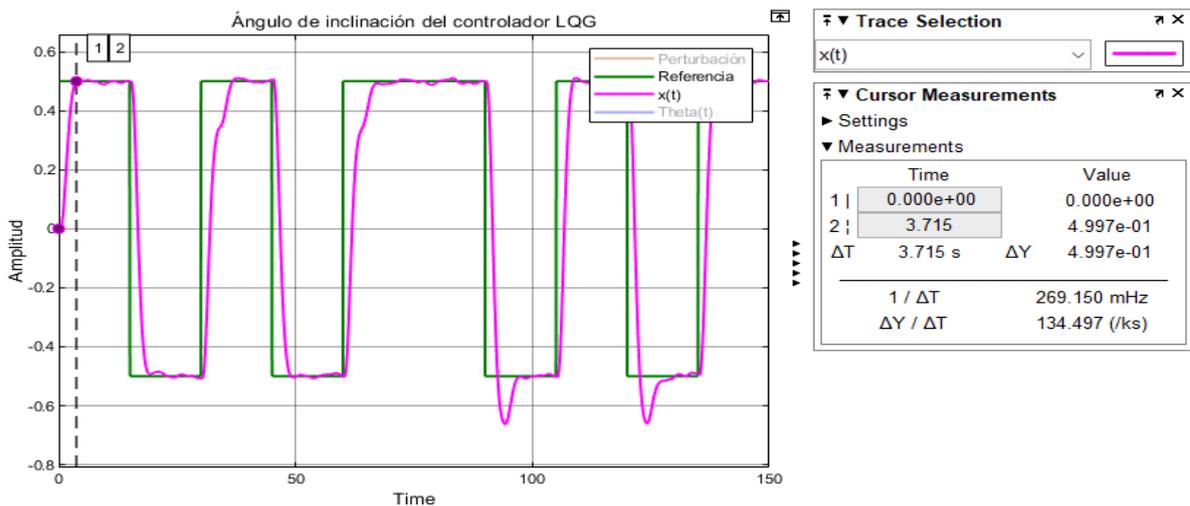


Figura 96 Tiempo de estabilización (t_{ss}) en controlador LQG para Evaluación 3.

Realizado por: Yuri Torres, 2024.

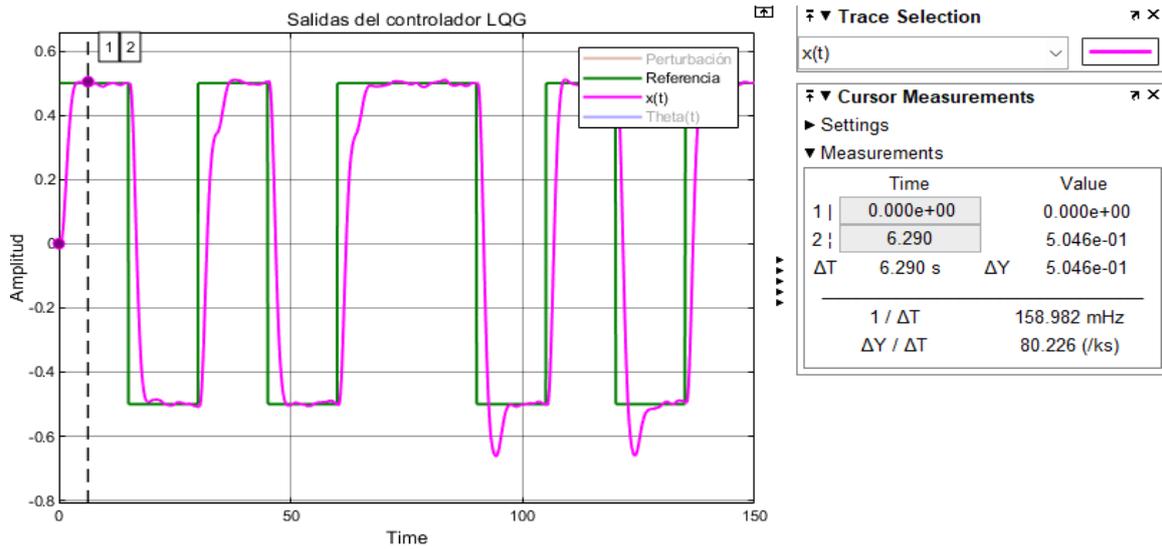


Figura 97 Sobre impulso máximo (M_p) en controlador LQG para Evaluación 3.

Realizado por: Yuri Torres, 2024.

$$M_p = \frac{0.5046 - 0.50}{0.50} = 0.0092 \quad (193)$$

$$M_p = \mathbf{0.92\%} \quad (194)$$

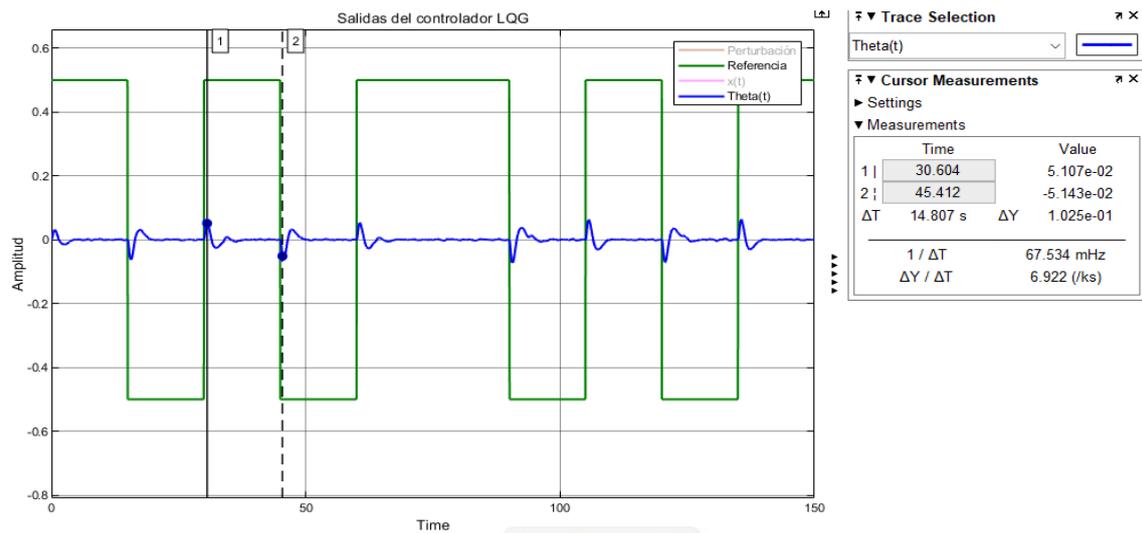


Figura 98 Ángulo de inclinación en controlador LQG para Evaluación 3.

Realizado por: Yuri Torres, 2024.

$$-0.05143^\circ \leq \theta \leq 0.5107^\circ$$

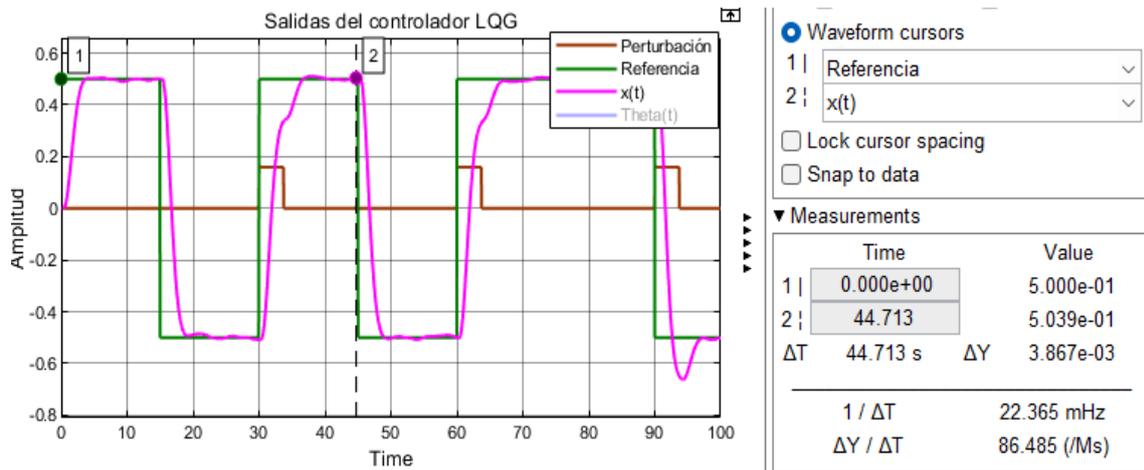


Figura 99 Error de estado estacionario (e_{ss}) en controlador LQG para Evaluación 3.

Realizado por: Yuri Torres, 2024.

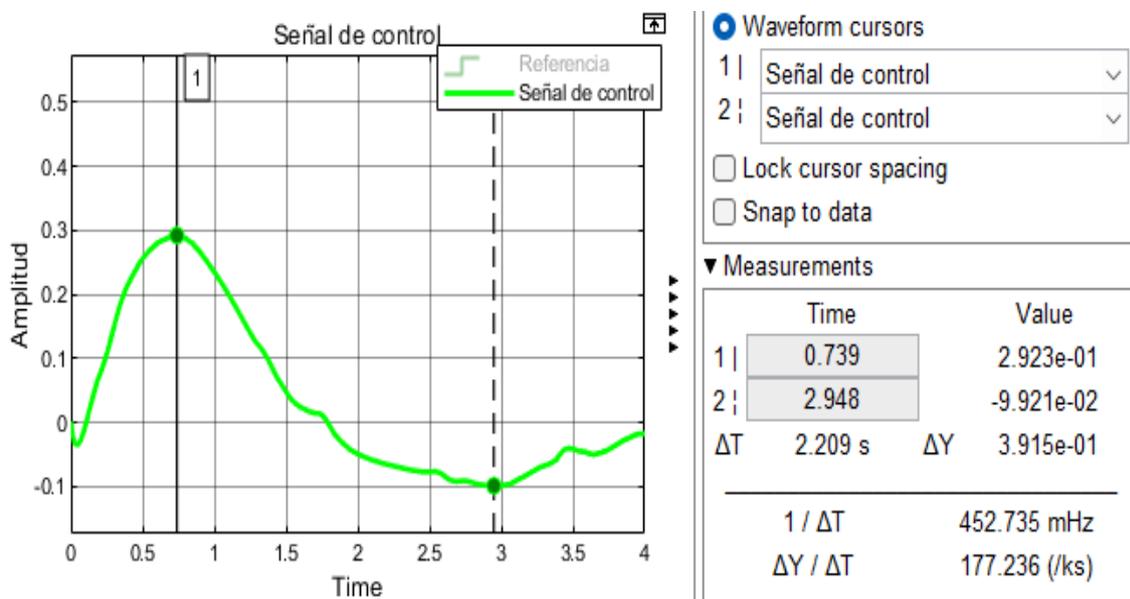


Figura 100 Señal de control para el controlador LQG para Evaluación 3.

Realizado por: Yuri Torres, 2024.

4.1.2.4 Análisis de Evaluación 4: $x = 10$ y $\theta = 15^\circ$

En las gráficas (101)-(105) se justifican los datos de evaluación 4, en la tabla (7).

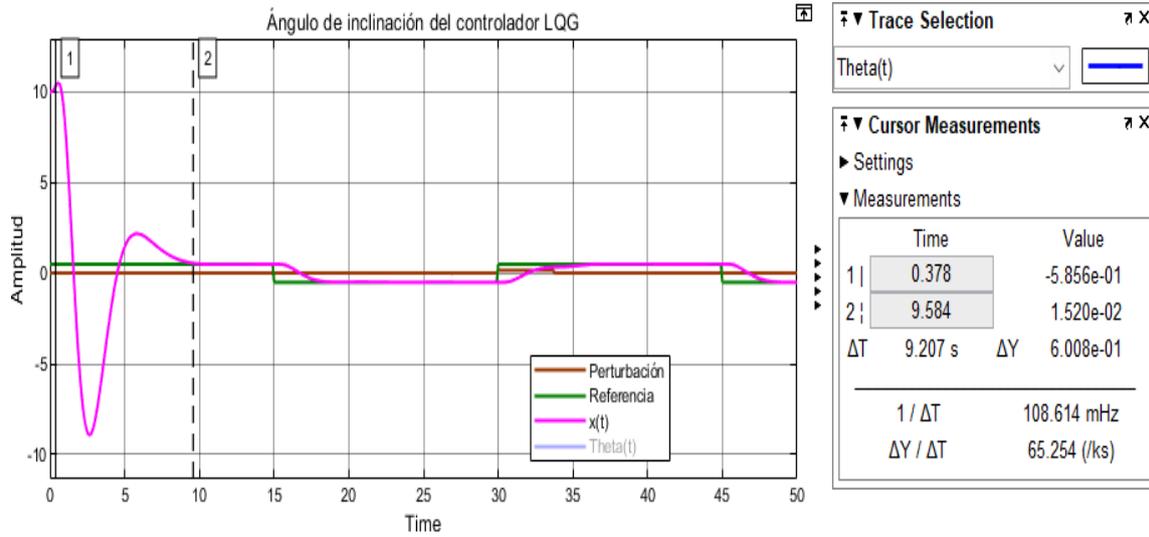


Figura 101 Tiempo de estabilización (t_{ss}) en controlador LQG para Evaluación 4.

Realizado por: Yuri Torres, 2024.

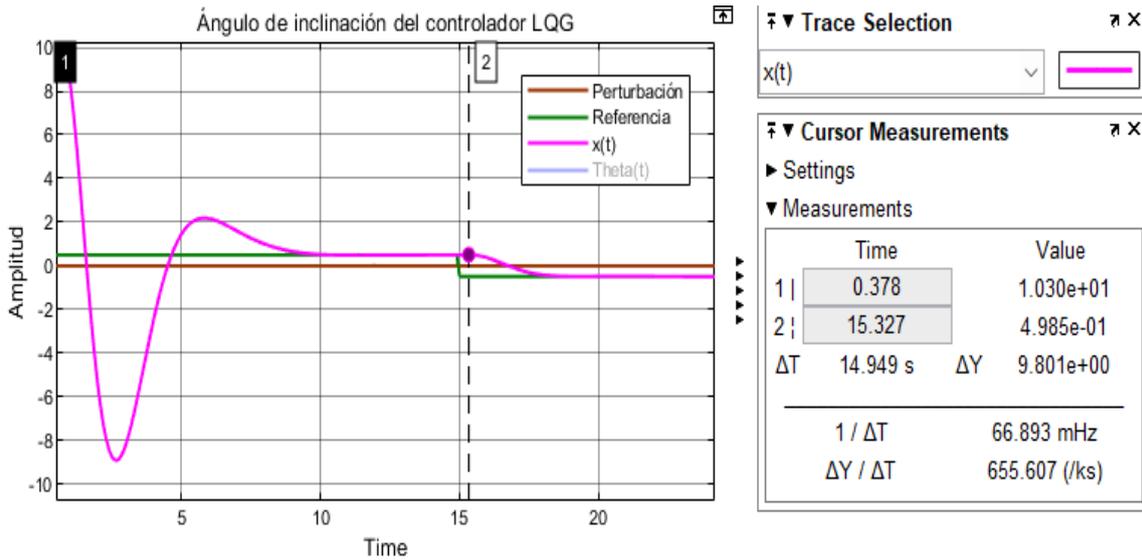


Figura 102 Sobre impulso máximo (M_p) en controlador LQG para Evaluación 4.

Realizado por: Yuri Torres, 2024.

$$M_p = \frac{0.4985 - 0.50}{0.50} = -0.003 \quad (195)$$

$$M_p = 0.3\% \quad (196)$$

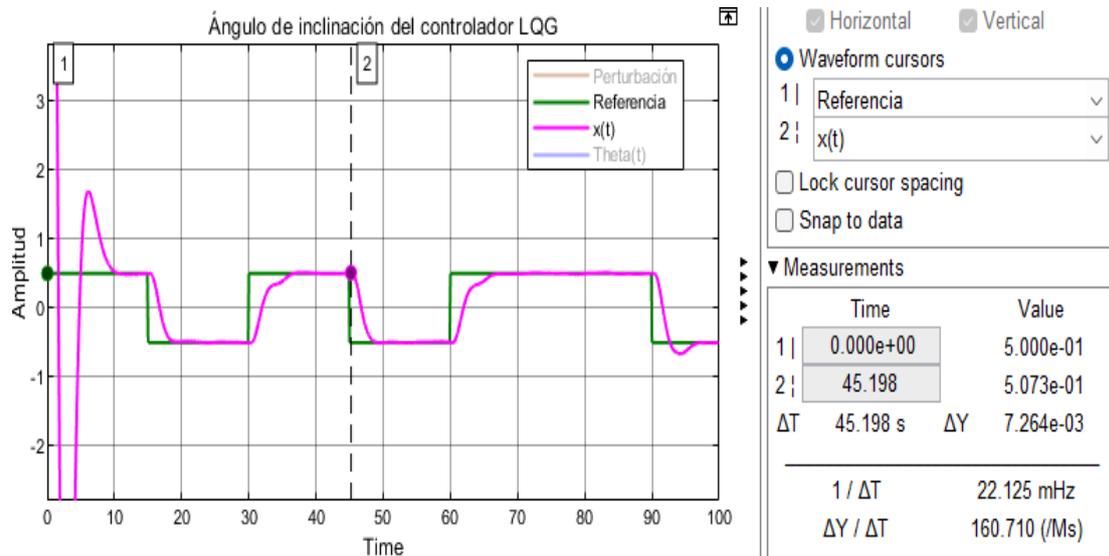


Figura 103 Error de estado estacionario (e_{ss}) en controlador LQG para Evaluación 4.

Realizado por: Yuri Torres, 2024.

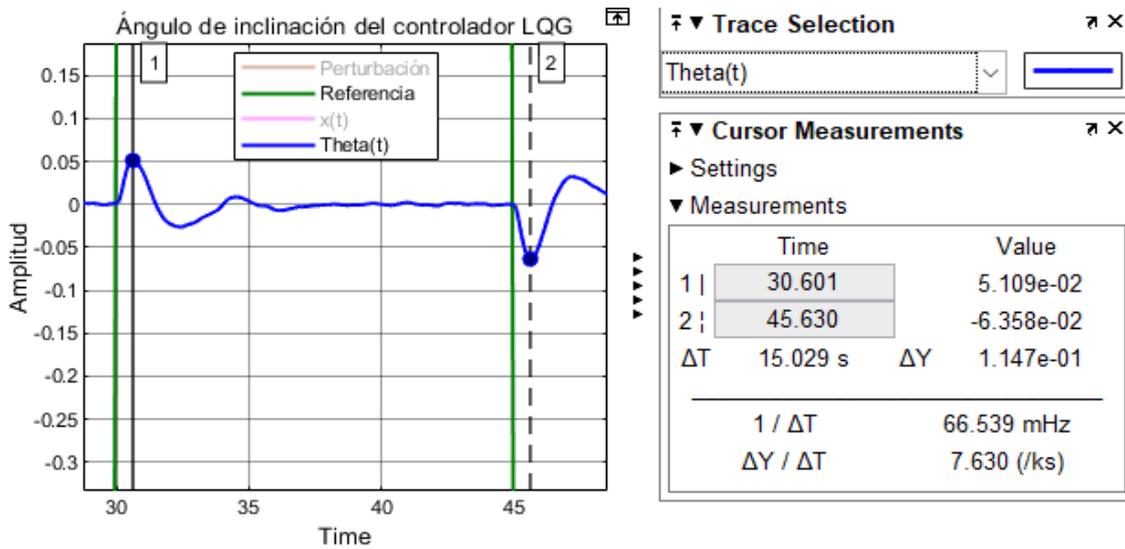


Figura 104 Ángulo de inclinación en el controlador LQG para Evaluación 4.

Realizado por: Yuri Torres, 2024.

$$-0.06358 \leq \theta \leq 0.05109$$

$$-3.643^\circ \leq \theta \leq 2.927^\circ$$

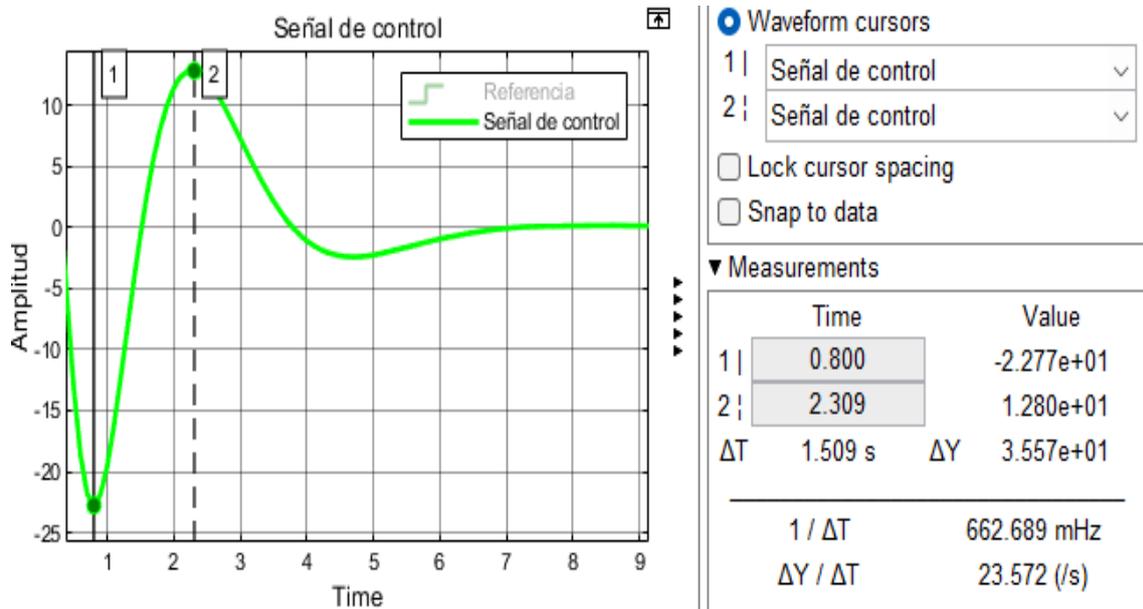


Figura 105 Señal de control en controlador LQG para Evaluación 4.

Realizado por: Yuri Torres, 2024.

Conclusiones

- La esencia del proceso radica en la integración del microcontrolador Arduino Uno para personalizar la implementación del control LQG en el robot Elegoo Tumbler. Este paso resulta fundamental, pues no solo optimiza su funcionamiento, sino también evidencia la eficiencia en diversos campos de control aplicativo, adaptándose eficazmente a las necesidades específicas del entorno de trabajo requerido.
- El diagrama de conexiones de los elementos electrónicos mediante el software de simulación Proteus, facilitó la identificación de posibles fallos o mejoras en el diseño, así como la optimización de la eficiencia del sistema.
- La parte analítica de la dinámica del robot se desarrolló mediante el modelo matemático del sistema no lineal aplicando el método Euler-Lagrange que describen las relación entre las entradas y salidas del sistema, del cual se utilizaron constantes reducidas que fueron precisas para una mejor identificación de las variables del sistema utilizadas para la linealización por medio de espacio de estados y técnicas del jacobiano.
- El uso de MATLAB© - Simulink permite la evaluación del control lineal cuadrático gaussiano LQG mediante el simulador numérico. Este enfoque permite estimar y regular perturbaciones en el sistema no lineal y el proceso ya linealizado.
- Se destaca la eficiencia del control LQG a través de las comparaciones con el control LQR, donde se comprueba que LQG alcanza un $t_{ss} = 3.71$ segundos, mientras el control LQR se estabiliza en $t_{ss} = 4.36$ segundos.
- El Sobre impulso máximo del control LQG es $M_p = 0.92\%$ bajo y significativo para el controlador LQR con $M_p = 13.22\%$. Este resultado indica que se logra una respuesta más suave y controlada, reduciendo las oscilaciones indeseadas gracias a LQG.
- En cuanto al ángulo de inclinación ($\theta_{max} = 0.05107^\circ$ y $\theta_{min} = -0.05143^\circ$) del control LQG son más cercanos al equilibrio que los obtenidos con el control LQR ($\theta_{max} = 0.1256^\circ$ y $\theta_{min} = -0.06146^\circ$). Esto indica una mayor precisión y estabilidad en el posicionamiento del sistema.

- Para señal de control en el control LQG ($u_{max} = 0.2925 V$ y $u_{min} = -0.0357 V$) muestran una mayor estabilidad en comparativa con el control LQR ($u_{max} = 0.2271 V$ y $u_{min} = -0.0818 V$) y el rango de valores ($-0.6042 V \leq u(t) \leq 0.4766 V$) es más estrecho, lo que sugiere una mayor robustez y precisión en la regulación.
- Los resultados numéricos respaldan de manera contundente la elección del control LQG para la estabilización efectiva del sistema electrónico robótico balanceado. La respuesta más rápida, menor sobre impulso, ángulos de inclinación más próximos al equilibrio y valores de señal de control más estables que destacan el desempeño superior del control LQG en comparativa con el control LQR.
- Después del análisis de pruebas, se realiza el algoritmo de control LQG en ARDUINO IDE y se embebe en la tarjeta del microcontrolador estableciendo una comunicación directa con el robot y manteniendo una exitosa estabilidad del péndulo cumpliendo con el objetivo específico del tema.
- Posteriormente se establece la comunicación Bluetooth entre el simulador numérico y la planta con el propósito de generar datos provenientes de los sensores lo que refleja el ángulo de inclinación y las posición de giro en X Y Z de aceleración angular en tiempo real robot.

Recomendaciones

- Se insta considerar el trabajo de titulación como un aporte investigativo en teorías de control y se sugiere referirlo en futuros aportes con validaciones experimentales en controladores y plataformas robóticas reales, esencial para corroborar su eficacia y robustez a las complejidades del entorno.
- Es importante indicar que para la interacción del robot ELEGOO Tumbler se recomienda usar comunicación por Wifi debido a su mayor velocidad en comparación con el Bluetooth.
- De acuerdo con el desarrollo del capítulo III, se constata la posibilidad de minimizar la latencia, asegurando así un control más reactivo y rápido. Incluso se observa que es factible lograr una mayor robustez mediante el reajuste de las ponderaciones Q y R.
- En futuras investigaciones, se recomienda explorar el impacto de las restricciones de salida y la saturación de entrada en el desempeño del sistema. Esto es crítico para aplicaciones donde los límites físicos del sistema pueden afectar significativamente la seguridad y eficacia del control, como en los sistemas de transporte automatizado o dispositivos médicos automatizados.

Bibliografías

- [1] J. S. Castro Beltrán, C. F. Vergara Ramírez, y J. S. Herrera Guayazan, «Comparación de un control lineal y no lineal en un levitador magnético», *Revista Ingenierías USBMed*, vol. 9, n.º 1, Art. n.º 1, 2018.
- [2] D. S. Torres, «t. TIEMPO CONTINUO 0 DISCRETO? UN PROBLEMA DE CONTROL OPTIMO».
- [3] C. D. C. Morales, «DISEÑO Y SIMULACIÓN DE UN CONTROLADOR POR MODOS DESLIZANTES DINÁMICOS PARA SISTEMAS INESTABLES EN LAZO ABIERTO CON RETARDO DE TIEMPO».
- [4] D. I. Zabala-Benavides y J. F. Salazar-Cáceres, «Design and implementation of LQG controller in Ball & Beam system», vol. 15, n.º 2, Art. n.º 2, 2021.
- [5] G. Oliver Peiró, «Diseño de un Prototipo del Robot Handle" de Boston Dynamics con Recurdyn y Mathematica."», 2018.
- [6] I. E. Erazo-Velasco, J. V. Bautista-Sánchez, R. I. Rodríguez-Jijón, y L. A. González-Quiñonez, «Diseño óptimo de un controlador LQG para un sistema mecánico de un péndulo invertido», *SIJIS*, vol. 3, n.º 7, Art. n.º 7, oct. 2022, doi: 10.51798/sijis.v3i7.536.
- [7] R. I. Carpio Frago, «Crédito bancario en el ciclo económico de México, 2003-2019: una aplicación del algoritmo Bry-Boschan y el filtro de Kalman», 2020, Accedido: 20 de enero de 2023. [En línea]. Disponible en: <http://repositorio-digital.cide.edu/handle/11651/4293>
- [8] L. E. G. Jaimes y D. P. Flórez, «Diseño de un controlador lineal cuadrático gaussiano y un controlador autosintonizado por asignación de polos para el control del voltaje de un generador CD», *Ingeniería*, vol. 26, n.º 1, Art. n.º 1, 2021, doi: 10.14483/23448393.16719.
- [9] A. G. Moisés, «DISEÑO Y CONSTRUCCIÓN DE UN ROBOT AUTO-BALANCEADO MEDIANTE ARDUINO».
- [10] «Projeto e implementação de controlador LQG/LTR em tempo real para um sistema de três tanques acoplados». Accedido: 26 de enero de 2023. [En línea]. Disponible en: <https://repositorio.unesp.br/handle/11449/202650>

- [11] M. Kamgarpour, «LUCA.FURIERI@EPFL.CH».
- [12] Y. Zheng, L. Furieri, M. Kamgarpour, y N. Li, «Sample complexity of linear quadratic gaussian (LQG) control for output feedback systems», en *Learning for dynamics and control*, PMLR, 2021, pp. 559-570.
- [13] L. G. T. Macías, «Modelado, simulación y control de un péndulo invertido».
- [14] E. G. Martínez Reyes, «Sistema cooperativo de robots industriales para tareas con interacción humana», Master's Thesis, Tesis (MC)–Centro de Investigación y de Estudios Avanzados del IPN ..., 2022.
- [15] «Estrategia de Gestión en la Educación Superior; pertinencia e impacto en la ... - JESÚS ESTUPIÑÁN RICARDO - Google Libros». Accedido: 9 de febrero de 2023. [En línea]. Disponible en: https://books.google.es/books?hl=es&lr=&id=Ww7UDwAAQBAJ&oi=fnd&pg=PA10&dq=Acuerdo+2018-040&ots=1jMF3oND9P&sig=3DShLzTTkx0a_Qv1JnB4f9ktpL8#v=onepage&q=Acuerdo%202018-040&f=false
- [16] K. L. Jácome Barahona, «Análisis de la ejecución presupuestaria de los proyectos de investigación científica y su impacto financiero en el cumplimiento del plan de desarrollo institucional de la Universidad Central del Ecuador en el periodo 2015–2020», Master's Thesis, Quito: UCE, 2022.
- [17] L. F. Sánchez-Gómez, «Estimados del fador dinámico bajo el principio de optimalidad de Bellman: el filtro Taylor-LQG-Fourier», n.º 63, Art. n.º 63, 2014.
- [18] I. E. V. Garcés, «NÉSTOR GABRIEL DE LA ROSA MEJILLÓN».
- [19] N. Kouakou, «Mobile robot odometric localization using decentralized kalman filter», Master's Thesis, Fen Bilimleri Enstitüsü, 2018.
- [20] T. Asfihani, D. Khusnul Arif, Subchan, F. Priyatno Putra, y Moch. Ardi Firmansyah, «Comparison of LQG and Adaptive PID Controller for USV Heading Control», *J. Phys.: Conf. Ser.*, vol. 1218, n.º 1, Art. n.º 1, may 2019, doi: 10.1088/1742-6596/1218/1/012058.
- [21] J. S. Aguilar Malavé, «Control predictivo basado en modelo para un sistema tipo SISO de nivel de líquido representado en variables de estados y su comparación con el

control PID y control por redes neuronales artificiales», bachelorThesis, La Libertad: Universidad Estatal Península de Santa Elena, 2023., 2023. Accedido: 15 de septiembre de 2023. [En línea]. Disponible en: <https://repositorio.upse.edu.ec/handle/46000/9608>

[22] C. Rauch y S. Wibbeling, «Serie: Digitalisierung in Beschaffung und Logistik (Teil 1/6): Transport: Neue Robotiklösungen für das Krankenhaus», *Klinik Einkauf*, vol. 5, n.º 01, pp. 32-34, 2023.

[23] J. miguel Garcia-Haro, E. Daniel Oña, S. Martinez, J. Hernandez-Vicen, y C. Balaguer, «Aplicación Robot Camarero: Control de Equilibrio para Transporte de Objetos», en *Conferencia internacional IEEE/RSJ de 2018 sobre robots y sistemas inteligentes (IROS)*, doi: 10.1109/IROS.2018.8593760.

[24] J. miguel Garcia-Haro, E. Daniel Oña, S. Martinez, J. Hernandez-Vicen, y C. Balaguer, «Waiter Robot Application: Balance Control for Transporting Objects», en *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, oct. 2018, pp. 1-9. doi: 10.1109/IROS.2018.8593760.

[25] J. Manyika *et al.*, «Un futuro que funciona: automatización, empleo y productividad», *Mckinsey Global Institute*, vol. 7, 2017.

[26] E. Universitat Politècnica De València, «Universitat Politècnica de València», *ing.agua*, vol. 18, n.º 1, p. ix, sep. 2014, doi: 10.4995/ia.2014.3293.

[27] «Home», Segway. Accedido: 9 de mayo de 2023. [En línea]. Disponible en: <https://www.segway.com/>

[28] J. D. Marchan Trinidad, «Diseño de un transportador personal eléctrico auto-equilibrante», abr. 2023, Accedido: 9 de mayo de 2023. [En línea]. Disponible en: <https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/24841>

[29] T. Lytvynenko, L. Hasenko, M. Elgandour, y I. Tkachenko, «Settlements Preparation to Future Transport Progress», en *Proceedings of the 4th International Conference on Building Innovations*, V. Onyshchenko, G. Mammadova, S. Sivitska, y A. Gasimov, Eds., en *Lecture Notes in Civil Engineering*. Cham: Springer International Publishing, 2023, pp. 433-440. doi: 10.1007/978-3-031-17385-1_34.

[30] «The All-New GT Series». Accedido: 9 de mayo de 2023. [En línea]. Disponible en: <https://store.segway.com/the-all-new-gt-series>

- [31] L. G. Triviño Macías y R. Vilanova i Arbós, *Modelado, simulación y control de un péndulo invertido*. 2020.
- [32] J. Aracil y F. Gordillo, «El Péndulo Invertido: un Desafío para el Control No Lineal», *Revista Iberoamericana de Automática e Informática industrial*, vol. 2, n.º 2, Art. n.º 2, 2005.
- [33] D. K. Solórzano Peñafiel, «Diseño e implementación de un controlador óptimo LQG, para un sistema de péndulo invertido aplicado en un equipo Lego Mindstorms.», B.S. thesis, 2018.
- [34] «Simax - Group SA Perfil de Compañía - Ecuador | Finanzas y ejecutivos clave | EMIS». Accedido: 9 de mayo de 2023. [En línea]. Disponible en: https://www.emis.com/php/company-profile/EC/Simax_-_Group_SA_es_9649813.html
- [35] «Tienda IO Mabe Ecuador | Homepage iomabeEC». Accedido: 9 de mayo de 2023. [En línea]. Disponible en: https://mabeglobal.com/es_EC/
- [36] J. R. Macías Palacios, «Análisis del proceso de lavado de botellas en una línea de producción de una compañía cervecera y propuesta de mejora.», PhD Thesis, Universidad de Guayaquil. Facultad de Ingeniería Industrial. Carrera de ..., 2023.
- [37] «Cervecería Nacional | La cervecera más grande», Cerveceria Nacional. Accedido: 9 de mayo de 2023. [En línea]. Disponible en: <https://www.cervecerianacional.ec/>
- [38] C. F. Caicedo Coro, «Importancia del uso del sistema scada para el desarrollo empresarial», 2019.
- [39] 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», *Computer Networks*, vol. 101, pp. 158-168, jun. 2016, doi: 10.1016/j.comnet.2015.12.017.
- [40] H. J. Rosas Ramos, «Diseño y análisis comparativo de un sistema de péndulo invertido aplicando técnicas de control lineal cuadrático Gaussiano», *Universidad Católica de Santa María*, sep. 2022, Accedido: 17 de octubre de 2023. [En línea]. Disponible en: <https://repositorio.ucsm.edu.pe/handle/20.500.12920/12050>
- [41] «2.6 Sistemas continuos y discretos, analógicos y digitales – Señales y Sistemas». Accedido: 17 de octubre de 2023. [En línea]. Disponible en: <http://blog.espol.edu.ec/telg1001/sistemas-continuos-y-discretos/>

- [42] A. A. R. Suárez y E. K. P. Segura, «ELABORACIÓN DE UN ROBOT TIPO PÉNDULO INVERTIDO E IMPLEMENTACIÓN DE CONTROLADOR DIFUSO EN BASE A CAMBIO DE POSICIÓN ANGULAR», 2015.
- [43] J. F. Gómez y J. J. S. Martín, «DISEÑO E IMPLEMENTACIÓN DE UN ROBOT AUTOBALANCEADO».
- [44] S. J. Pincin, «Diseño, construcción y control de un robot balancín.», PhD Thesis, Universidad Nacional de Cuyo, 2016.
- [45] «LQG Controller Design of the Mobile Inverted Pendulum - ProQuest». Accedido: 17 de octubre de 2023. [En línea]. Disponible en: <https://www.proquest.com/openview/ba000e64fc4dc8e9f3ccdaaf57cccdf6/1?pq-origsite=gscholar&cbl=18750>
- [46] S. J. Pincin, «Diseño, construcción y control de un robot balancín.», *piIngenieriaMecanica*, Universidad Nacional de Cuyo, 2016. Accedido: 19 de mayo de 2023. [En línea]. Disponible en: <https://ricabib.cab.cnea.gov.ar/557/>
- [47] P. M. Llosa Rodriguez, «Modelamiento clásico en el espacio de estado de los sistemas de control en tiempo discreto», jul. 2018, Accedido: 5 de mayo de 2023. [En línea]. Disponible en: <http://repositorio.une.edu.pe/handle/20.500.14039/6171>
- [48] 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.
- [49] W. B. Reyes Castillo, «Modelo Matemático de métodos Adaptativos para predecir el comportamiento del consumo de Energía Eléctrica en la ciudad de Esmeraldas 2020-2021», Master's Thesis, 2022.
- [50] «Modern Control Engineering 5th edition (9780136156734) - Textbooks.com». Accedido: 5 de octubre de 2023. [En línea]. Disponible en: <https://www.textbooks.com/Modern-Control-Engineering-5th-Edition/9780136156734/Katsuhiko-Ogata.php>
- [51] K. Ogata, *Ingeniería de control moderna*. Pearson Educación, 2003.
- [52] E. A. Braschi, «Enfoque práctico de control moderno».

- [53] «metodos de linealizacion de sistemas». Accedido: 9 de octubre de 2023. [En línea]. Disponible en: <https://studylib.es/doc/8864848/metodos-de-linealizacion-de-sistemas>
- [54] N.-M. A. Πατσο\acute\nuρας, «Μοντελοπο\acuteiotaση καιota\acute\varepsilonpsilonλεγχος ποδηλ\acute\alphalφατου ιοτασοροπ\acuteiotaας», B.S. thesis, 2022.
- [55] 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
- [56] W. Bolton, *Mecatrónica*. Alpha Editorial, 2017.
- [57] J. E. Pérez Valencia, «Construcción de un robot balancín controlado por una aplicación móvil», feb. 2021, Accedido: 5 de mayo de 2023. [En línea]. Disponible en: <http://dspace.istvidanueva.edu.ec/handle/123456789/116>
- [58] V. D. Caiza García y C. E. Verdezoto Cedeño, «Implementación de dos robots autónomos y uno controlado mediante tecnología bluetooth para las categorías seguidor de línea con obstáculos, laberinto y balancín.», sep. 2016, Accedido: 10 de octubre de 2023. [En línea]. Disponible en: <http://repositorio.ucsg.edu.ec/handle/3317/6607>
- [59] «Balance-Bot - ProQuest». Accedido: 15 de septiembre de 2023. [En línea]. Disponible en: <https://www.proquest.com/openview/080d5273c83b3058e802b3684705a356/1?pq-origsite=gscholar&cbl=2026366&diss=y>
- [60] «White Noise». Accedido: 26 de octubre de 2023. [En línea]. Disponible en: https://www.probabilitycourse.com/chapter10/10_2_4_white_noise.php
- [61] «Ruido gaussiano», *Wikipedia, la enciclopedia libre*. 1 de julio de 2020. Accedido: 26 de octubre de 2023. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Ruido_gaussiano&oldid=127387629
- [62] Δ. Φ. Σαμαρ\acute\alphaς, «Virtual sensing in structures by monitoring linear substructures», B.S. thesis, 2022.
- [63] C. Urrea y R. Agramonte, «Kalman Filter: Historical Overview and Review of Its Use in Robotics 60 Years after Its Creation», *Journal of Sensors*, vol. 2021, p. e9674015, sep. 2021, doi: 10.1155/2021/9674015.

- [64] 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.
- [65] C. Iwendi, M. A. Alqarni, J. H. Anajemba, A. S. Alfakeeh, Z. Zhang, y A. K. Bashir, «Robust Navigational Control of a Two-Wheeled Self-Balancing Robot in a Sensed Environment», *IEEE Access*, vol. 7, pp. 82337-82348, 2019, doi: 10.1109/ACCESS.2019.2923916.
- [66] «ELEGOO Tumbler Self-Balancing Robot Car V1.1/V1.0 Tutorial», ELEGOO Official. Accedido: 28 de febrero de 2024. [En línea]. Disponible en: <https://www.elegoo.com/blogs/arduino-projects/elegoo-tumbler-self-balancing-robot-car-tutorial>
- [67] E. W. Kamen y B. S. Heck, *Fundamentos de señales y sistemas usando la web y Matlab*. Pearson Educación, 2008.
- [68] «Form Linear-Quadratic-Gaussian (LQG) servo controller - MATLAB lqgtrack - MathWorks América Latina». Accedido: 31 de octubre de 2023. [En línea]. Disponible en: <https://la.mathworks.com/help/control/ref/ss.lqgtrack.html>

Anexos

Anexo 1. Desarrollo de las ecuaciones de movimiento

Para obtener las ecuaciones de energía cinética específicas del sistema, es necesario considerar la inercia, masa de la rueda y el péndulo. Por lo tanto, se procede a calcular la primera ecuación cinética de la rueda.

La ecuación (1), se reemplaza por el sistema de notación para expresar en términos dinámicos y obtener la primera ecuación cinética del sistema como se muestra en la expresión (84).

Ecuación de energía cinética de la masa de la rueda:

$$E_{C1} = \frac{1}{2} m_r \dot{x}^2 \quad (84)$$

Para la siguiente ecuación denominada energía cinética de la inercia de la rueda; se considera el movimiento angular (φ), dado por la velocidad tangencial de la rueda en movimiento circular (v) y el desplazamiento angular (r) que ha recorrido como se muestra en (5).

Cinemática del movimiento circular:

$$v = r \cdot \varphi \quad (5)$$

Para E_{C2} es necesario determinar los componentes de la velocidad con respecto a la posición del péndulo.

Componente en 'x'

$$x_p = x + l \text{sen}(\theta)$$

Velocidad lineal del péndulo en posición 'x'

$$\frac{\partial x_p}{\partial t} [x + l \text{sen}(\theta)] \frac{d(\theta)}{dt}$$

Lo que representa ($E_{C2.1}$): primer componente para la E_{C2}

$$x_{p(t)} = \dot{x} + l \cos(\theta) \dot{\theta} \quad (E_{C2.1})$$

Componente en "y"

$$y_p = l \cos(\theta)$$

Velocidad lineal del péndulo en posición 'y'.

$$\frac{\partial y_p}{\partial t} [-l \operatorname{sen}(\theta)] \frac{d(\theta)}{dt}$$

Entonces se obtiene lo siguiente ($E_{C2.2}$): segunda componente para la ecuación (E_{C2}).

$$y_{p(t)} = -l \operatorname{sen}(\theta) \dot{\theta}$$

A continuación, se suma las componentes ($E_{C2.1}$) y ($E_{C2.2}$) y se eleva al cuadrado para obtener la componente de E_{C2} :

$$v^2 = (\dot{x} + l \cos(\theta) \dot{\theta})^2 + (-l \operatorname{sen}(\theta) \dot{\theta})^2$$

$$v^2 = \dot{x}^2 + 2\dot{x} \dot{\theta} l \cos(\theta) + \dot{\theta}^2 l^2$$

Se reemplaza en la ecuación principal (1) y obtenemos (85).

$$E_{C2} = \frac{1}{2} m_p (\dot{x}^2 + 2\dot{x} \dot{\theta} l \cos(\theta) + \dot{\theta}^2 l^2)$$

$$E_{C2} = \frac{1}{2} m_p \dot{x}^2 + m_p \dot{x} \dot{\theta} l \cos(\theta) + \frac{1}{2} m_p \dot{\theta}^2 l^2 \quad (85)$$

Se obtiene la energía potencial de la forma general como indica (2), considerando el objeto en reposo y suponiendo que el péndulo está en equilibrio, lo cual implica que no hay un ángulo de inclinación en las ruedas, es decir, el ángulo de inclinación 0° .

$$E_p = m_p g l [\cos(\theta) - \cos(0^\circ)]$$

$$\text{Energía potencial} \quad E_p = m_p g l [\cos(\theta) - 1] \quad (86)$$

Anexo 2. Desarrollo de Lagrangiano

Con E_{C1} , E_{C2} y E_p , se aplica la definición del Lagrangiano en la ecuación (3) y se obtiene la ecuación (87).

$$L = E_{C1} + E_{C2} - E_p$$

$$L = \frac{1}{2} m_r \dot{x}^2 + \frac{1}{2} m_p \dot{x}^2 + m_p \dot{x} \dot{\theta} l \cos(\theta) + \frac{1}{2} m_p \dot{\theta}^2 l^2 - m_p g l \cos(\theta) \quad (87)$$

Anexo 3. Desarrollo de Euler Lagrange

La ecuación de Euler-Lagrange, mediante coordenadas generalizadas (qi) y velocidades generalizadas (\dot{qi}), permite obtener las ecuaciones de movimiento de un sistema al derivar parcialmente la variación de una función fundamental conocida como el Lagrangiano (3). Por tanto, son necesarias las fuerzas externas que actúan sobre el robot tal como indica la expresión (12).

$$\frac{d}{dt} \frac{dL}{d\dot{q}_i} - \frac{dL}{dq_i} = \sum Q_i \quad (12)$$

Las dos ecuaciones dinámicas del robot se obtienen mediante la expresión Lagrangiana (3).

Se une la derivada con respecto a 'x'; ecuación (102), (107) y (112) componentes de Euler Lagrange formando la ecuación (114).

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

Se reemplaza $\dot{\phi}$

$$\begin{aligned} & \frac{\Gamma_m}{r} - \frac{u_e}{r} (\dot{\phi} - \dot{\theta}) - u_s \dot{x} \\ & \frac{\Gamma_m}{r} - \frac{u_e}{r} \left(\frac{\dot{x}}{r} \right) + \frac{u_e}{r} \dot{\theta} - u_s \dot{x} \\ & \frac{\Gamma_m}{r} - \dot{x} \left(u_s + \frac{u_e}{r^2} \right) + \frac{u_e}{r} \dot{\theta} \end{aligned} \quad (114)$$

Se unen la derivada con respecto a 'θ'; ecuación (103) y (113), componentes de Euler Lagrange, formando la ecuación (115)

$$\begin{aligned} & \frac{\partial}{\partial t} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = \sum Q_\theta \\ & -\Gamma_m + u_e (\dot{\phi} - \dot{\theta}) \\ & -\Gamma_m + u_e \left(\frac{\dot{x}}{r} - \dot{\theta} \right) \end{aligned} \quad (115)$$

Se deriva parcialmente la expresión (3), respecto a las variables $x, \dot{x}, \theta, \dot{\theta}$:

La derivada con respecto a x es =0 como se indica en la ecuación (116).

Derivada con respecto a \dot{x}

$$\begin{aligned}\frac{\partial L}{\partial \dot{x}} &= \frac{\partial}{\partial \dot{x}} \left[\frac{1}{2} (m_r + m_p) \dot{x}^2 \right] + \frac{d}{d\dot{x}} [m_p \dot{x} \dot{\theta} l \cos(\theta)] + \frac{d}{d\dot{x}} \left[\frac{1}{2} m_p \dot{\theta}^2 l^2 \right] \\ &\quad - \frac{d}{d\dot{x}} [m_p g l \cos(\theta)] \\ \frac{\partial L}{d\theta} &= m_r \dot{x} + m_p \dot{x} + m_p \dot{\theta} l \cos(\theta) + 0 - 0 \\ \frac{\partial L}{\partial \dot{x}} &= (m_r + m_p) \dot{x} + m_p \dot{\theta} l \cos(\theta)\end{aligned}$$

Una vez derivado respecto a \dot{x} , volvemos a derivar respecto al tiempo por el contexto de un sistema dinámico y se tiene la ecuación (117).

$$\begin{aligned}&\frac{d}{dt} [(m_r + m_p) \dot{x} + m_p \dot{\theta} l \cos(\theta)] \\ (m_r + m_p) \ddot{x} + m_p \ddot{\theta} l \cos(\theta) - m_p l \operatorname{sen}(\theta) (\dot{\theta})^2 &\end{aligned} \quad (117)$$

Derivada con respecto a θ y formación de la ecuación (108).

$$\begin{aligned}\frac{\partial L}{\partial \theta} &= \frac{\partial}{\partial \theta} \left[\frac{1}{2} (m_r + m_p) \dot{x}^2 \right] + \frac{\partial}{\partial \theta} [m_p \dot{x} \dot{\theta} l \cos(\theta)] + \frac{\partial}{\partial \theta} \left[\frac{1}{2} m_p \dot{\theta}^2 l^2 \right] \\ &\quad - \frac{\partial}{\partial \theta} [m_p g l \cos(\theta)] \\ \frac{\partial L}{\partial \theta} &= 0 - (m_p l \dot{x} \dot{\theta} \operatorname{sen}(\theta)) + 0 - (m_p g l (-\operatorname{sen}(\theta))) + 0 \\ \frac{dL}{d\theta} &= m_p g l \operatorname{sen}(\theta) - m_p l \dot{\theta} \dot{x} \operatorname{sen}(\theta)\end{aligned} \quad (118)$$

Derivamos con respecto a $\dot{\theta}$

$$\begin{aligned}\frac{\partial L}{\partial \dot{\theta}} &= \frac{\partial}{\partial \dot{\theta}} \left[\frac{1}{2} (m_r + m_p) \dot{x}^2 \right] + \frac{\partial}{\partial \dot{\theta}} [m_p \dot{x} \dot{\theta} l \cos(\theta)] \\ &\quad + \frac{\partial}{\partial \dot{\theta}} \left[\frac{1}{2} m_p \dot{\theta}^2 l^2 \right] - \frac{\partial}{\partial \dot{\theta}} [m_p g l \cos(\theta)] \\ \frac{\partial L}{\partial \dot{\theta}} &= 0 + m_p \dot{x} l (\cos \theta) + m_p \dot{\theta} l^2 - 0 \\ \frac{\partial L}{\partial \dot{\theta}} &= m_p \dot{x} l (\cos \theta) + m_p \dot{\theta} l^2\end{aligned}$$

Una vez derivado respecto a $\dot{\theta}$, volvemos a derivar respecto al tiempo por el contexto de un sistema dinámico y se tiene la ecuación (119).

$$\frac{d}{dt} [m_p \dot{x} l \cos(\theta) + m_p \dot{\theta} l^2]$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) = m_p l \ddot{x} \cos(\theta) - m_p l \dot{x} \sin(\theta) \dot{\theta} + m_p l^2 (\ddot{\theta}) \quad (119)$$

Anexo 4. Cambio de ecuaciones dinámicas a ecuaciones en variables de estados y puntos de equilibrio

La ecuación (123), se expresa en términos de estados con aproximaciones del ángulo $\sin(\theta) \approx \theta$, $\cos(\theta) \approx 1$ y $\theta \approx 0$ y se muestra en la expresión (127).

$$\ddot{x} = -\frac{m_p l \cos(\theta)}{(m_r + m_p)} \ddot{\theta} - \frac{m_p l \sin(\theta)}{(m_r + m_p)} \dot{\theta} + \frac{1}{(m_r + m_p)} F - \frac{ue}{(m_r + m_p)} \dot{x} \quad (123)$$

$$\ddot{x} = -\frac{m_p l \cos(0)}{(m_r + m_p)} \ddot{\theta} - \frac{m_p l \sin(0)}{(m_r + m_p)} \dot{\theta} + \frac{1}{(m_r + m_p)} F - \frac{ue}{(m_r + m_p)} \dot{x}$$

$$\ddot{x} = -\frac{m_p l}{(m_r + m_p)} \ddot{\theta} + \frac{1}{(m_r + m_p)} F - \frac{ue}{(m_r + m_p)} \dot{x}$$

$$\dot{x}_2 = -\frac{m_p l}{(m_r + m_p)} x_4 + \frac{u}{(m_r + m_p)} - \frac{ue}{(m_r + m_p)} x_2 \quad (127)$$

La ecuación (125) también se reduce para llevar a variables de estados y obtener la ecuación (129).

$$\ddot{\theta} = \frac{ue}{m_r l} \dot{x} - \frac{(m_r + m_p)us}{m_r m_p l^2} \dot{\theta} + \frac{(m_r + m_p)g}{m_r l} \theta - \frac{F}{m_r l} \quad (125)$$

$$\dot{x}_4 = \frac{ue}{m_r l} x_2 - \frac{(m_r + m_p)us}{m_r m_p l^2} x_4 + \frac{(m_r + m_p)g}{m_r l} x_3 - \frac{u}{m_r l} \quad (129)$$

Anexo 5. Desarrollo de Jacobianos

En el desarrollo de la linealización de sistemas no lineales, los puntos de equilibrio se utilizan como referencia para aproximar el comportamiento local del sistema. Para hacerlo, se calculan las derivadas parciales lo que es Jacobianos del sistema; y con los puntos de equilibrio se aproximan a cero todas las variables del sistemas y se obtienen las matrices de estados linealizadas.

De las ecuaciones (127) y (129), se procede al desarrollo de Jacobianos considerando las demás expresiones como (126) y (128).

$$\dot{x}_1 = x_2 \quad (126)$$

$$\dot{x}_2 = -\frac{m_p l}{(m_r + m_p)} x_4 + \frac{u}{(m_r + m_p)} - \frac{ue}{(m_r + m_p)} x_2 \quad (127)$$

$$\dot{x}_3 = x_4 \quad (128)$$

$$\dot{x}_4 = \frac{ue}{m_r l} x_2 - \frac{(m_r + m_p)us}{m_r m_p l^2} x_4 + \frac{(m_r + m_p)g}{m_r l} x_3 - \frac{u}{m_r l} \quad (129)$$

Se lleva a cabo las derivadas parciales de las cuatro ecuaciones con el modelo linealizado para generar las matrices de estados siguientes.

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

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial y_1(x)}{\partial x_1} & \frac{\partial y_1(x)}{\partial x_2} & \frac{\partial y_1(x)}{\partial x_3} & \frac{\partial y_1(x)}{\partial x_4} \\ \frac{\partial y_2(x)}{\partial x_1} & \frac{\partial y_2(x)}{\partial x_2} & \frac{\partial y_2(x)}{\partial x_3} & \frac{\partial y_2(x)}{\partial x_4} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} u$$

MATRIZ A

Derivada parcial para la fila 1 donde se cumple la condición de $\dot{x}_1 = x_2$.

$$\text{FILA } f_1 = \dot{x}_1 = x_2$$

$$\frac{\partial f_1}{\partial x_1} = \frac{\partial x_2}{\partial x_1} = 0 \quad \frac{\partial f_1}{\partial x_2} = \frac{\partial x_2}{\partial x_2} = 1 \quad \frac{\partial f_1}{\partial x_3} = \frac{\partial x_2}{\partial x_3} = 0 \quad \frac{\partial f_1}{\partial x_4} = \frac{\partial x_2}{\partial x_4} = 0$$

$$A = \frac{\partial f_1}{\partial x} = [0 \quad 1 \quad 0 \quad 0]$$

Derivada parcial fila 2, respecto a x_1, x_2, x_3, x_4

$$\text{FILA } f_2 = \dot{x}_2$$

Para $f_2: x_1$

$$\frac{\partial f_2}{\partial x_1} \left[-\frac{m_p l}{(m_r + m_p)} x_4 + \frac{u}{(m_r + m_p)} - \frac{ue}{(m_r + m_p)} x_2 \right] = 0$$

Para $f_2: x_2$

$$\frac{\partial f_2}{\partial x_2} \left[-\frac{m_p l}{(m_r + m_p)} x_4 + \frac{u}{(m_r + m_p)} - \frac{ue}{(m_r + m_p)} x_2 \right] = -\frac{ue}{(m_r + m_p)}$$

Para $f_2: x_3$

$$\frac{\partial f_2}{\partial x_3} \left[-\frac{m_p l}{(m_r + m_p)} x_4 + \frac{u}{(m_r + m_p)} - \frac{ue}{(m_r + m_p)} x_2 \right] = 0$$

Para $f_2: x_4$.

$$\frac{\partial f_2}{\partial x_4} \left[-\frac{m_p l}{(m_r + m_p)} x_4 + \frac{u}{(m_r + m_p)} - \frac{ue}{(m_r + m_p)} x_2 \right] = -\frac{m_p l}{(m_r + m_p)}$$

FILA $f_3 = \dot{x}_3 = x_4$

$$\frac{\partial f_3}{\partial x_1} = \frac{\partial x_4}{\partial x_1} = 0 \quad \frac{\partial f_3}{\partial x_2} = \frac{\partial x_4}{\partial x_2} = 0 \quad \frac{\partial f_3}{\partial x_3} = \frac{\partial x_4}{\partial x_3} = 0 \quad \frac{\partial f_3}{\partial x_4} = \frac{\partial x_4}{\partial x_4} = 1$$

$$A = \frac{\partial f_3}{\partial x} = [0 \quad 0 \quad 0 \quad 1]$$

Derivada parcial para la fila 4, respecto a x_1, x_2, x_3, x_4

FILA f_4

Para $f_4: x_1$

$$\frac{\partial f_4}{\partial x_1} \left[\frac{ue}{m_r l} x_2 - \frac{(m_r + m_p)us}{m_r m_p l^2} x_4 + \frac{(m_r + m_p)g}{m_r l} x_3 - \frac{u}{m_r l} \right] = 0$$

Para $f_4: x_2$

$$\frac{\partial f_4}{\partial x_2} \left[\frac{ue}{m_r l} x_2 - \frac{(m_r + m_p)us}{m_r m_p l^2} x_4 + \frac{(m_r + m_p)g}{m_r l} x_3 - \frac{u}{m_r l} \right]$$

$$\frac{\partial f_4}{\partial x_2} = \frac{ue}{m_r l}$$

Para $f_4: x_3$:

$$\frac{\partial f_4}{\partial x_3} \left[\frac{ue}{m_r l} x_2 - \frac{(m_r + m_p)us}{m_r m_p l^2} x_4 + \frac{(m_r + m_p)g}{m_r l} x_3 - \frac{u}{m_r l} \right]$$

$$\frac{\partial f_4}{\partial x_3} = \frac{(m_r + m_p)g}{m_r l}$$

Para $f_4: x_4$:

$$\frac{\partial f_4}{\partial x_4} \left[\frac{ue}{m_r l} x_2 - \frac{(m_r + m_p)us}{m_r m_p l^2} x_4 + \frac{(m_r + m_p)g}{m_r l} x_3 - \frac{u}{m_r l} \right]$$

$$\frac{\partial f_4}{\partial x_4} = \frac{(m_r + m_p)us}{m_r m_p l^2}$$

MATRIZ B

Para $f_1: u$

$$B = \frac{\partial f_1}{\partial u} = 0$$

Para $f_2: u$

$$\frac{\partial f_2}{\partial u} \left[-\frac{m_p l}{(m_r + m_p)} x_4 + \frac{u}{(m_r + m_p)} - \frac{ue}{(m_r + m_p)} x_2 \right]$$

$$\frac{\partial f_2}{\partial u} = \frac{1}{(m_r + m_p)}$$

Para $f_3: u$

$$B = \frac{\partial f_3}{\partial u} = 0$$

Para $f_4: u$

$$\frac{\partial f_4}{\partial u} \left[\frac{ue}{m_r l} x_2 - \frac{(m_r + m_p)us}{m_r m_p l^2} x_4 + \frac{(m_r + m_p)g}{m_r l} x_3 - \frac{u}{m_r l} \right]$$

$$\frac{\partial f_4}{\partial u} = -\frac{1}{m_r l}$$

Con el desarrollo anterior se justifica la matriz B de la expresión (121).

Anexo 6. Código MATLAB© para la comunicación Bluetooth

```
/*Control LQG para robot autoequilibrio.*/

#include "Motor.h"
#include "LQG.h"
#include "LQE.h"

// #include "ControlLQG.h"
// #include <util/atomic.h>

// extern Mpu6050 Mpu6050;
```

```

Temporizador Temporizador;
extern Mpu6050 Mpu6050;
extern Motor Motor;
extern LQR LQR;
extern FiltroKalman FiltroKalman;

/*Declaración de las constantes de realimentación*/
LQR::LQR(){
/*  theta, theta_dot,  x_dot,      x      ,  omega  , omega_dot*/
  K[0] = 100, K[1] = 0.7, K[2] = 15, K[3] = 0.24, K[4] = 2.5, K[5] =
0.5;
}

/*-----*/
/*-----variables para la comunicación*/
float distancia, velocidad, anguloG, velanguloG, senalControl;
int16_t distanciapE, distanciapD, distancianE, distancianD,
        velocidadpE, velocidadpD, velocidadnE, velocidadnD,
        anguloGpE, anguloGpD, anguloGnE, anguloGnD,
        velanguloGpE, velanguloGpD, velanguloGnE, velanguloGnD,
        senalControlpE, senalControlpD, senalControlnE, senalControlnD;
byte buffer_rx[10], dato_recibido[4]; //Se ubica un espacio más porque
el último bit siempre contiene "basura".
byte buffer_tx[42];
/*-----*/

void setup(){
  Serial.begin(9600);
  // TimeC=millis();
  Motor.IniciarPines();
  Motor.IniciarEncoder();
  Temporizador.iniciar(TIMER);
  Mpu6050.iniciar();
  delay(100);
  pinMode(9,OUTPUT);
  digitalWrite(9,HIGH);
}

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]);
    }

/*=====ENVIO DE DATOS A SIMULINK=====*/
    buffer_tx[0]='R';
    buffer_tx[41]='\n';
    /*=====DISTANCIA X=====*/
    distancia=LQR.integral_velocidad_robot; //Estado X en Serial
    if (distancia >= 0) {
        distanciapE = int(distancia);
        distanciapD = int((distancia - int(distancia)) * 100);
        distancianE = 0, distancianD = 0;
    } else {
        distancianE = int((-1) * distancia);
        distancianD = int((-1) * distancia - int((-1) * distancia));
        distanciapE = 0, distanciapD = 0;
    }
    buffer_tx[1] = distanciapE >> 8; //corridos de 8 bits para sacar
parte alta
    buffer_tx[2] = distanciapE & 0x00FF; //mascara
    buffer_tx[3] = distanciapD >> 8; //corridos de 8 bits para sacar
parte alta
    buffer_tx[4] = distanciapD & 0x00FF; //mascara
    /*-----*
-----*/
    buffer_tx[5] = distancianE >> 8; //corridos de 8 bits para sacar
parte alta
    buffer_tx[6] = distancianE & 0x00FF; //mascara
    buffer_tx[7] = distancianD >> 8; //corridos de 8 bits para sacar
parte alta
    buffer_tx[8] = distancianD & 0x00FF; //mascara

/*=====VELOCIDAD X'=====*/
    velocidad=LQR.filtro_velocidad; //Estado X' en Serial
    if (velocidad >= 0) {
        velocidadpE = int(velocidad);
        velocidadpD = int((velocidad - int(velocidad)) * 100);
        velocidadnE = 0, velocidadnD = 0;
    } else {
        velocidadnE = int((-1) * velocidad);
        velocidadnD = int((-1) * velocidad - int((-1) * velocidad));
        velocidadpE = 0, velocidadpD = 0;
    }
}

```

```

    buffer_tx[9] = velocidadpE >> 8; //corridos de 8 bits para sacar
parte alta
    buffer_tx[10] = velocidadpE & 0x00FF; //mascara
    buffer_tx[11] = velocidadpD >> 8; //corridos de 8 bits para sacar
parte alta
    buffer_tx[12] = velocidadpD & 0x00FF; //mascara
/*-----*/
-----*/
    buffer_tx[13] = velocidadnE >> 8; //corridos de 8 bits para sacar
parte alta
    buffer_tx[14] = velocidadnE & 0x00FF; //mascara
    buffer_tx[15] = velocidadnD >> 8; //corridos de 8 bits para sacar
parte alta
    buffer_tx[16] = velocidadnD & 0x00FF; //mascara

/*=====ANGULO θ=====*/
anguloG=FiltroKalman.angulo; //Estado θ en Serial
if (anguloG >= 0) {
    anguloGpE = int(anguloG);
    anguloGpD = int((anguloG - int(anguloG)) * 100);
    anguloGnE = 0, anguloGnD = 0 ;
} else {
    anguloGnE = int((-1) * anguloG);
    anguloGnD = int((-1) * anguloG - int((-1) * anguloG));
    anguloGpE = 0, anguloGpD = 0;
}
    buffer_tx[17] = anguloGpE >> 8; //corridos de 8 bits para sacar
parte alta
    buffer_tx[18] = anguloGpE & 0x00FF; //mascara
    buffer_tx[19] = anguloGpD >> 8; //corridos de 8 bits para sacar
parte alta
    buffer_tx[20] = anguloGpD & 0x00FF; //mascara
/*-----*/
-----*/
    buffer_tx[21] = anguloGnE >> 8; //corridos de 8 bits para sacar
parte alta
    buffer_tx[22] = anguloGnE & 0x00FF; //mascara
    buffer_tx[23] = anguloGnD >> 8; //corridos de 8 bits para sacar
parte alta
    buffer_tx[24] = anguloGnD & 0x00FF; //mascara

/*=====VELOCIDAD ANGULAR θ'=====*/
velanguloG=FiltroKalman.Giro_x; //Estado θ' en Serial
if (velanguloG >= 0) {
    velanguloGpE = int(velanguloG);
    velanguloGpD = int((velanguloG - int(velanguloG)) * 100);
    velanguloGnE = 0, velanguloGnD = 0;
} else {
    velanguloGnE = int((-1) * velanguloG);

```

```

        velanguloGnD = int((-1) * velanguloG - int((-1) * velanguloG));
        velanguloGpE = 0, velanguloGpD = 0;
    }
    buffer_tx[25] = velanguloGpE >> 8; // corridos de 8 bits para
sacar parte alta
    buffer_tx[26] = velanguloGpE & 0x00FF; //mascara
    buffer_tx[27] = velanguloGpD >> 8; // corridos de 8 bits para
sacar parte alta
    buffer_tx[28] = velanguloGpD & 0x00FF; //mascara
    /*-----*/
-----*/
    buffer_tx[29] = velanguloGnE >> 8; // corridos de 8 bits para
sacar parte alta
    buffer_tx[30] = velanguloGnE & 0x00FF; //mascara
    buffer_tx[31] = velanguloGnD >> 8; // corridos de 8 bits para
sacar parte alta
    buffer_tx[32] = velanguloGnD & 0x00FF; //mascara

    /*=====SEÑAL DE CONTROL=====*/
    senalControl=LQR.pwm_izquierda; //Señal de control, PWM
    if (senalControl >= 0) {
        senalControlpE = int(senalControl);
        senalControlpD = int((senalControl - int(senalControl)) * 100);
        senalControlnE = 0, senalControl = 0 ;
    } else {
        senalControlnE = int((-1) * senalControl);
        senalControlnD = int((-1) * senalControl - int((-1) *
senalControl));
        senalControlpE = 0, senalControlpD = 0;
    }
    buffer_tx[33] = senalControlpE >> 8; //corridos de 8 bits para
sacar parte alta
    buffer_tx[34] = senalControlpE & 0x00FF; //mascara
    buffer_tx[35] = senalControlpD >> 8; //corridos de 8 bits para
sacar parte alta
    buffer_tx[36] = senalControlpD & 0x00FF; //mascara
    /*-----*/
-----*/
    buffer_tx[37] = senalControlnE >> 8; //corridos de 8 bits para
sacar parte alta
    buffer_tx[38] = senalControlnE & 0x00FF; //mascara
    buffer_tx[39] = senalControlnD >> 8; //corridos de 8 bits para
sacar parte alta
    buffer_tx[40] = senalControlnD & 0x00FF; //mascara

    /*=====*/
=====*/

```

```

    Serial.write(buffer_tx,42);
//    Serial.print(LQR.integral_velocidad_robot); //Estado X en
Serial
//    Serial.print(LQR.filtro_velocidad); //Estado X' en Serial
//    Serial.print(FiltroKalman.angulo); //Estado  $\theta$  en Serial
//    Serial.print(FiltroKalman.Giro_x); //Estado  $\theta'$  en Serial
//    Serial.print(LQR.pwm_izquierda); //Señal de control, PWM
//    Serial.print(LQR.r); //Referencia
//    Serial.println(LQR.r - FiltroKalman.angulo); //Error (Ref -  $\theta$ )
}

```

Anexo 7. Análisis para la estabilidad del sistema

A continuación, el código en el software MATLAB© para comprobar la inestabilidad del sistema; desarrollado por el autor:

```

%BALANCÍN
%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
g=9.81; %aceleración de la gravedad
%%
%Matrices linealizadas
A=[0 1 0 0;
    0 -ue / (m_r + m_p) 0 -(m_p * lp) / (m_r + m_p);
    0 0 0 1;
    0 ue / (m_r * lp) ( (m_r + m_p) * g)/(m_r * lp) ( (m_r + m_p) * us) /
(m_r * m_p * lp^2)];
B=[0;
    1 / (m_r + m_p);
    0;
    -1 / (m_r * lp)];
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 el espacio de estados ');
model_2=ss(A,B,C,D)
%Modelo discretizado
disp('Modelo discreto en el 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('El sistema es Estable.');
```

else

```

    disp('El sistema es Inestable.');
```

end

Respuesta de la simulación:

```

>> El sistema es Inestable
```

Polos del sistema:

```

    0
    11.9999
   -0.3019
  -11.4688
```

El sistema es inestable.

Anexo 8. Análisis para la Controlabilidad

```

%% Comprobación de controlabilidad
cont=[B A*B (A^2)*B (A^3)*B];
if rank(cont)==4
    disp('El Sistema es Controlable.')
```

else

```

    disp('El Sistema no es Controlable')
```

end

Respuesta de la simulación:

```

>>El Sistema es Controlable
```

El Sistema es Controlable.

Anexo 9. Análisis para la Observabilidad

```
%% 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
```

Respuesta de la simulación:

```
>>El Sistema es Observable
```

El Sistema es Observable.

Anexo 10. Diseño del código para el servo-controlador LQG en MATLAB©

```
%BALANCÍN
%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
g=9.81; %aceleración de la gravedad
%%
%Matrices linealizadas
A=[0 1 0 0;
    0 -ue / (m_r + m_p) 0 -(m_p * lp) / (m_r + m_p);
    0 0 0 1;
    0 ue / (m_r * lp) ( (m_r + m_p) * g)/(m_r * lp) ( (m_r + m_p) * us) /
(m_r * m_p * lp^2)];
B=[0;
    1 / (m_r + m_p);
    0;
    -1 / (m_r * lp)];
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 el espacio de estados ');
model_2=ss (A, B,C,D)
%Modelo discretizado
disp ('Modelo discreto en el 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('El sistema es Estable.');
```

```

else
    disp('El sistema es Inestable.');
```

```

end
%Análisis Bibo
%Tiempo continuo
[n1,d1]=ss2tf(A,B,C,D)
n1a=n1(1,:);
n1b=n1(2,:);
gx=tf(n1a,d1) %Función de transferencia X(s)/F(s)
zpk(gx)
gt=tf(n1b,d1) %Función de transferencia 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
%Análisis 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_amp=[A zeros(4,1);-C2 0]
```

```

B_amp=[B;0]
C_amp=[C2 0]
D_amp=[0]
sys_a=ss(A_amp,B_amp,C_amp,D_amp)
ctrb_a=[B_amp A_amp*B_amp (A_amp^2)*B_amp (A_amp^3)*B_amp]
rangoctrb=rank(ctrb_a)
%Matriz de Controlabilidad del sistema aumentado (P)
Pp=[A_amp B_amp;-C_amp 0]
rangP=rank(Pp)
%Rango de P n=5, el sistema aumentado es controlable
rangA=rank(A_amp); %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=[10 0 0 0 0;0 3 0 0 0;0 0 50 0 0;0 0 0 3 0;0 0 0 0 20]
R_lqr=1;
[k_lqi,S,e]=lqr(A_amp,B_amp,Q_lqr,R_lqr)
disp('***Vector de Control Optimo*** ')
k_lqr=k_lqi(1:end-1) %Vector óptimo de regulación de estados
k_i=k_lqi(end) %Vector de regulación del error(Ki) - SEGUIMIENTO

disp('Autovalores del sistema realimentado ')
eig(A_amp - B_amp*k_lqi)
%Sistema de control LQR con seguimiento
disp('Sistema de Control LQI Realimentado')
sys_r=ss(A_amp - B_amp*k_lqi,B_amp,C_amp,D_amp)
%% filtro de Kalman
%Para 1 f salida, x posición del carro
%Obs: Se diseña el filtro de Kalman para 1 salida, ya que el diseño para el
%servocontrolador LQG es para el control de 1 salida.
G=ones(4,1);
%H_2s=zeros(2,1); %h para 2 salidas
H_1s=0; %h para 1 salida ,
%Da es para 1 salida
disp('Sistema aumentado para el filtro de Kalman');
sys_k=ss(A,[B G],C2,[D_amp H_1s])
%Matrices de Covarianza de los Ruidos
Q_k=0.75; %1 ruido de proceso
Rak=1.5; %1 ruido de medida, para 1 salida ,"x" posición del carro
disp('*****Filtro de Kalman *****)
disp('Modelo en espacio de estados:')
[Kest,L,P]=kalman(sys_k,Q_k,R_k)
%% Servocontrolador LQG
disp('*****Servo-Controlador LQG*****)
disp('Modelo en espacio de estados :')
K_lqg=lqgtrack(Kest,k_lqi) %Servocontrolador LQG
%Sistema realimentado con Regulador LQG(SIN SERVO)
disp('*****Sistema realimentado con Regulador LQG*****)
aaa=A - B*k_lqr;
aab=B*k_lqr;
aba=zeros(4,4);
abb=A - L*C2;

```

```

Aaa=[aaa aab;aba abb]
Baa=[zeros(4,1) zeros(4,1);ones(4,1) -L]
Caa=[C zeros(2,4)]
Daa=[zeros(2,2)]

model_rlqg=ss(Aaa,Baa,Caa,Daa)
%Sistema realimentado con SERVOCONTROLADOR LQG (Seguimiento de Referencia)
disp('****Sistema realimentado con SERVOCONTROL LQG ****');
model_2Slqg=feedback(model_2, K_lqg,+1)

```

Anexo 11. Diseño del código para el controlador LQG en Arduino

Robot Auto equilibrio.ino

```

/*Control LQG para robot con autoequilibrio.*/
#include "Motor.h"
#include "LQG.h"
Temporizador Temporizador;
extern Mpu6050 Mpu6050;
extern Motor Motor;
extern LQR LQR;
LQR::LQR(){
/*  theta, theta_dot,  x_dot ,      x      ,  omega  , omega_dot*/
  K[0] = 67, K[1] = 0.7, 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();
  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[0]);
  }
}

```

LQE.cpp

```
#include "LQE.h"

////////////////////////////////LQE //////////////////////////////////
void FiltroKalman::Yiorderfilter(float angulo_med, float giro_med, float
dt, float K1) {

    angle6 = K1 * angulo_med + (1 - K1) * (angle6 + giro_med * dt);
    // return angle6;
}

////////////////////////////////kalman////////////////////////////////

void FiltroKalman::Filtro_Kalman(double angulo_med, double giro_med,
float dt, float Q_angulo, float Q_giro, float R_angulo, float C_0) {

    /*Actualizar nuestra estimación de ángulo
    angle += angle_dot * dt
        += (gyro - gyro_bias) * dt
        += q * dt*/
    angulo += (giro_med - q_sesgo) * dt;//Integración de la velocidad
angular dt: tiempo de muestreo

    angulo_err = angulo_med - angulo; //Retroalimentación del ángulo
obtenido y el ángulo real

    /*Calcule la derivada de la matriz de covarianza
    A continuación, se añade a los elementos diagonales de Q, que son
    Q_angulo y Q_giro.
    */

    Pdot[0] = Q_angulo - P[0][1] - P[1][0]; /* 0,0 */
    Pdot[1] = -P[1][1]; /* 0,1 */
    Pdot[2] = -P[1][1]; /* 1,0 */
    Pdot[3] = Q_giro; /* 1,1 */

    /*Matriz de covarianza.*/
    P[0][0] += Pdot[0] * dt;
    P[0][1] += Pdot[1] * dt;
    P[1][0] += Pdot[2] * dt;
    P[1][1] += Pdot[3] * dt;

    /*P<2,1> = P<2,2> * C'<2,1>, Lo utilizamos dos veces. Esto hace
posible pre calcular y almacenar los dos valores. Nótese que C[0,1] =
C_1 es cero, por lo que no calculamos ese término.*/
```

```

    PCt_0 = C_0 * P[0][0]; /* + C_1 * P[0][1] = 0 */ /*El C_1 muestra que
la medición de estado no se relaciona con la estimación del sesgo del
giróscopo.*/
    PCt_1 = C_0 * P[1][0]; /* + C_1 * P[1][1] = 0 */ /*El C_1 muestra que
la medición de estado no se relaciona con la estimación del sesgo del
giróscopo.*/

    /* Calcular la estimación del error. Del documento del filtro de
Kalman:
        E = C P C' + R
    E = R_angulo + C_0 * PCt_0 /* + C_1 * PCt_1 = 0 */;

    /* Calcular las ganancias del filtro de Kalman. Del documento de
Kalman:
    K_0 = PCt_0 / E;
    K_1 = PCt_1 / E;

    Esto nos ahorra una multiplicación en coma flotante.*/
    t_0 = PCt_0; /* C_0 * P[0][0] + C_1 * P[1][0] */
    t_1 = C_0 * P[0][1]; /* + C_1 * P[1][1] = 0 */
    P[0][0] -= K_0 * t_0;
    P[0][1] -= K_0 * t_1;
    P[1][0] -= K_1 * t_0;
    P[1][1] -= K_1 * t_1;

    /* Actualizar nuestra estimación de estado. Una vez más, desde el
documento de Kalman:
    angulo += K_0 * angulo_err; //ángulo óptimo
    q_sesgo += K_1 * angulo_err;// diferencia entre el valor medio y el
verdadero valor de la magnitud medida

    /* Elimina el sesgo de nuestro giroscopio */
    angulo_dot = giro_med - q_sesgo; //velocidad angular óptima
}

////////////////////Kalman////////////////////////////////////
//////////////////// Evaluación de ángulo////////////////////////////////////
void FiltroKalman::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 de:
    Angulo_RW = Angulo; //automóvil calculado por el filtro complementario
de primer orden

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

```

```

    Filtro_Kalman(Angulo, Giro_x, dt, Q_angulo, Q_giro, R_angulo,
C_0); //Llamado al filtro de Kalman

//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
Yiorderfilter(angleAx, Giro_y, dt, K1); //filtro de primer orden
}

```

Lqe.h

```

#ifndef LQE_h
#define LQE_h

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

class FiltroKalman{
public:
    /* angulo_med en rad, determinado desde aceleraciones */

    /* R representa el ruido de covarianza de la medición. En este caso,
es una matriz 1x1 que dice que esperamos 0.5 rad jitter del
acelerómetro.*/

    /* Q es una matriz 2x2 que representa el ruido de covarianza del
proceso.
En este caso, indica cuánto confiamos en el acelerómetro en
relación con los giroscopios.*/

    /* El C_0 muestra que la medida de estado se relaciona directamente
con la estimación de estado. El C_1 muestra que la medición de estado no
se relaciona con la estimación del sesgo del giróscopo.
En realidad, no usamos esto.*/

```

```

    void Yiordenfilter(float angulo_med, float giro_med, float dt, float
K1);
    void Filtro_Kalman(double angulo_med, double giro_med, float dt,
float Q_angulo, float Q_giro, float R_angulo, float C_0); /*float C_1*/
    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;
    float angle6;
    float angulo_dot; /*Derivada del ángulo*/

    float Angulo_RW;
private:
    /*Nuestros dos estados, el ángulo y el sesgo giroscópico.
    Como subproducto del cálculo del ángulo.*/
    float angulo_err, q_sesgo;
    float Pdot[4] = { 0, 0, 0, 0 }; /*Almacena la derivada de la matriz
de covarianza*/

    /*Nuestra matriz de covarianza.
    Esto se actualiza en cada paso de tiempo para determinar qué tan
bien los sensores están rastreando el estado real.*/
    float P[2][2] = {{ 1, 0 }, { 0, 1 }};

    float Pct_0, Pct_1, E, K_0, K_1, t_0, t_1;

    //float angulo_dot; /*Derivada del
ángulo*/
};
#endif

```

LQG.cpp

```

#include "LQG.h"
#include "Wire.h"
#include "Motor.h"
#include "MPU6050.h"
#include "LQE.h"

MPU6050 MPU6050;
Mpu6050 Mpu6050;
LQR LQR;
FiltroKalman FiltroKalman;
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::conta
dor_encoder_izquierda_a;

    encoder_derecha_pulso_num_velocidad += pwm_derecha < 0 ? (-
Motor::contador_encoder_derecha_a) :
Motor::contad
or_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] * (FiltroKalman.angulo - 0) + K[1] *
(FiltroKalman.Giro_x - 0);
}
void LQR::Estado_omega_dot()
{
    rotation_control_output = ajuste_velocidad_de_giro + K[5] *
FiltroKalman.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
    FiltroKalman.Angletest(ax, ay, az, gx, gy, gz, dt, Q_angulo, Q_giro,
R_angulo, C_0, K1); // Obtaining Angle by Kalman Filter
}

```

LQG.h

```

#ifndef _LQG_h
#define _LQG_h

#include "MsTimer2.h"
#include "LQE.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 LQG 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
};

```

Motor.cpp

```

#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);
}

```

Motor.h

```

#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

```

Velocidad de Medcion.cpp

```

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

static void ContadorEncoderDerechaA();
static void ContadorEncoderIzquierdaA();
//Se encargan de contar los pulsos de los encoder de
//la rueda de derecha a izquierda
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 12. Datos Técnicos del Motor DC GA37-520

Tabla 8 Datos Técnicos del Motor DC GA37-520 [66].

Información Técnica	
Tensión Nominal	12 V DC
Rango de voltaje de operación	1.0V ~ 16.0V DC
Relación de Reducción	1:30
Peso	0,150 kg
Corrientes sin carga	≤200 mA
Velocidad sin carga	360rpm ± 10%
Ruido eléctrico, Motor equipado con Varistor	Si
Encoder, Doble interruptor Hall	26 Pulse de Salida
Temperatura de Almacenamiento.	-20°C ~ + 85°C

Anexo 13. Datos técnicos del Arduino Uno

Tabla 9 Datos técnicos del Arduino Uno [9]

INFORMACIÓN TÉCNICA	
Característica	Valor
Microcontrolador	ATmega328P
Voltaje de Operación	5V
Voltaje de Entrada (Recomendado)	7-12V
Voltaje de Entrada (Límite)	6-20V
Pines Digitales I/O	14 (6 proporcionan PWM)
Pines Analógicos I/O	6
Corriente Máxima por Pin	40 mA
Corriente Máxima en 3,3V Pin	50 mA
Memoria Flash	32 KB (0.5 KB usado por bootloader)
SRAM	2 KB
EEPROM	1 KB
Velocidad del Reloj	16 MHz

Dimensiones	68,6 × 53,4 mm
-------------	----------------

Anexo 14. Datos Técnicos del Sensor MPU-6050

Tabla 10 Datos Técnicos del Sensor MPU-6050 [66].

INFORMACION TÉCNICO					
Parámetro	Condiciones	Mín.	Típ.	Máx.	Unidad
Sensor de Temperatura	Rango	-40		+85	°C
Sensibilidad	Sin ajuste		340		LSB/°c
Desplazamiento de Temperatura	35°C		-521		LSB
Linealidad	Ajuste lineal mejor (de -40°C a +85°C)			±1	°C
VDD Suministro de energía	Voltajes de Operación	2.375	3.46		V
Voltaje de Referencia VLOGIC	Solo MPU-6050	1,71	1,8	1,89	V
Corriente de Operación Normal	Giroscopio + Acelerómetro		3,9		mA
	Giroscopio + Acelerómetro (DMP deshabilitado)		3,8		mA
	Solo Giroscopio (DMP y Acelerómetro deshabilitados)		3,6		mA
	Solo Acelerómetro (DMP y Giroscopio deshabilitados)		500		µA
Corriente de Modo Bajo Consumo	Tasa de actualización de 1,25 Hz		10		µA
	Tasa de actualización de 5 Hz		20		µA
	Tasa de actualización de 20 Hz		70		µA

	Tasa de actualización de 40 Hz		140		μA
Voltaje de Referencia VLOGIC	Solo MPU-6050, este valor de ser ≤ VDD en todo momento	-	-	-	-
Interfaz Serie	Frecuencia de operación SPI, Todos los registros Lectura/Escritura	-	100	±10%	MHz

Anexo 15. Datos Técnicos de Modulo Bluetooth HC-05

Tabla 11 Datos Técnicos de Modulo Bluetooth HC-05 [58].

INFORMACIÓN TÉCNICA	
Características	Valor
Protocolo Bluetooth	Bluetooth v2.0 + EDR
Frecuencia de Operación.	2,4 GHz
Alcance máximo	Aproximadamente 10 metros
Modos de Operación	Maestro y Esclavo
Perfil Soportado	SPP (Serial Port Profile)
Voltaje de Operación	3,3V – 5V
Consumo de Corriente	30 mA (Conexión Activa)
	1 mA (Modo de Espera)
Interfaz de Conmutación	UART (TTL)
Tasa de Transmisión UART	Ajustable (Hasta 1382400 bps)
Nivel Lógico UART	3,3 V
	5 V (Tolerante)
Dimensiones	Aproximadamente 28 × 15 mm
Antena	Antena incorporada

Anexo 16. Datos técnicos del Puente H TB6612FNG

Tabla 12 Puente H del Robot Balancín [66].

Características	Símbolo	Condición de Prueba	Mínimo	Típico	Máximo	Unidad
	STBY = VCC = 3 V VM = 5 V			1.1	1.8	
Corriente de Suministro ICC(3V)	STBY = VCC = 5.5 V VM = 5 V	-	-	1.5	2.2	mA
Corriente de Suministro en Standby ICC(STB)	-	-	-	-	1	mA
Corriente de Control IM(STB)	STBY = 0 V		-	-	1	uA
Voltaje de Entrada de Control Alto VIH	-	$VCC \times 0.7$	-	-	$VCC + 0.2$	uA
Voltaje de Entrada de Control Bajo VIL	-	-0.2	-	-	$VCC \times 0.3$	V
Corriente de Entrada de Control Alto I _{IH}	VIN = 3 V	-	5	15	25	uA
Corriente de Entrada de Control Bajo I _{IL}	VIN = 0 V	-	-	-	1	uA
Voltaje de Entrada en Standby Alto VIH(STB)	-	$VCC \times 0.7$	-	-	$VCC + 0.2$	V
Voltaje de Entrada en Standby Bajo VIL(STB)	-	-0.2	-	-	$VCC \times 0.3$	V
Corriente de Entrada en Standby Alto I _{IH} (STB)	VIN = 3 V	-	5	15	25	uA
Corriente de Entrada en	VIN = 0 V	-	-	-	1	Ua

Standby Bajo IIL(STB)						
Voltaje de Saturación de Salida (U+L)1	IO = 1 A VCC = VM = 5 V	-	-	0.5	0.7	V
Voltaje de Saturación de Salida (U+L)2	IO = 0.3 A, VCC = VM = 5 V	-	0.15	-	0.21	V
Corriente de Fuga de Salida Superior (IL(U))	VM = VOU = 15 V	-	-	-	1	uA
Tensión de Diodo Regenerativo Superior (VF(U))	IF = 1 A	-	1	-	1.1	V
Tensión de Diodo Regenerativo Superior (VF(U))	IF = 1 A	-	1	-	1.1	V
Voltaje de Detección de Baja Tensión UVLD	-	-	-	1.9	-	V
Voltaje de Recuperación de Baja Tensión UVLC (Valor Diseñado)	-	-	2.2	-	-	V
Velocidad de Respuesta tr (Valor Diseñado)	-	-	24	-	-	ns
Tiempo Muerto de Apagado tf (Valor Diseñado)	-	-	41	-	-	ns
Velocidad de Respuesta de Alto a Bajo Hacia L	-	-	50	-	-	ns
Tiempo de Protección de Penetración (Valor Diseñado)	-	-	230	-	-	ns
Temperatura de Operación del	-	-	-	-	175	°C

Circuito de Apagado TSD						
Histéresis del Apagado Térmico Δ TSD (Valor Diseñado)	-	-	-	20	-	$^{\circ}\text{C}$

Anexo 17. Diagrama del Circuito Integrado Puente H

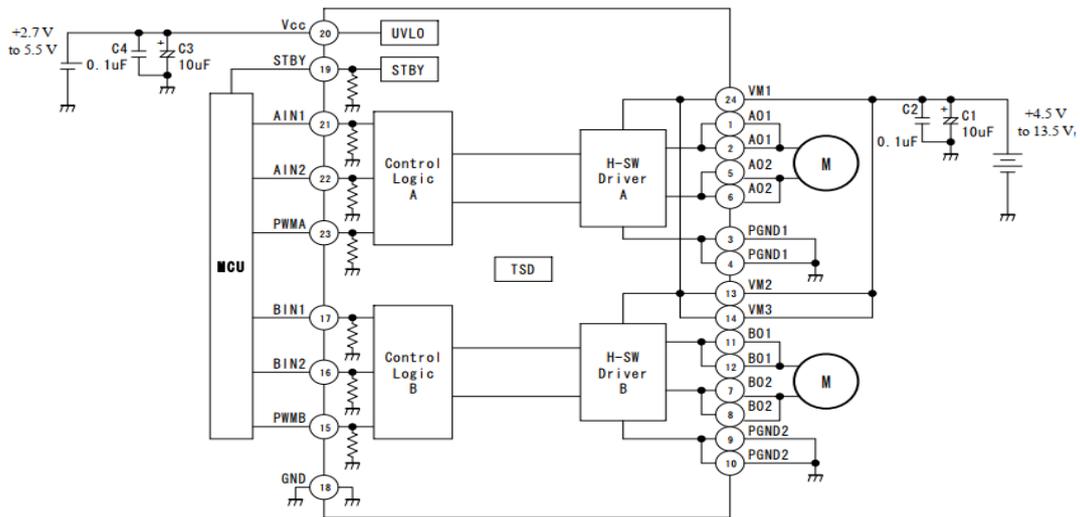


Figura 106 Diagrama del puente H [66].

Anexo 18. Herramienta System Identification.

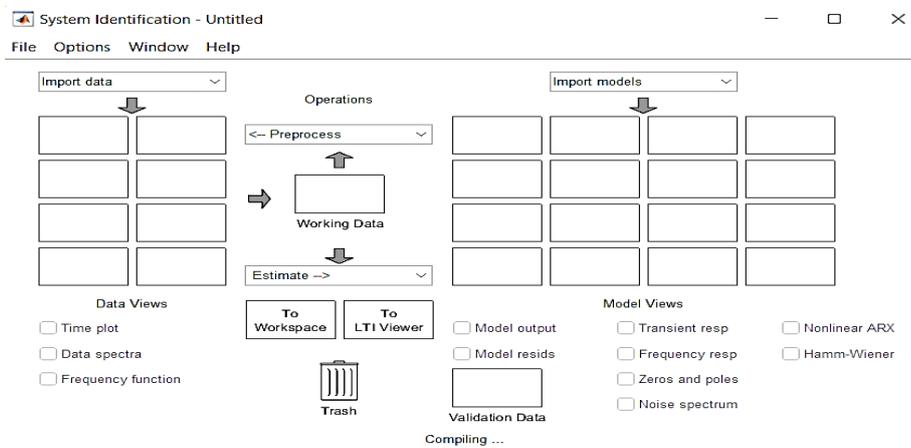


Figura 107 System Identification [67].

Anexo 19. Acción controlada del Robot Balancín

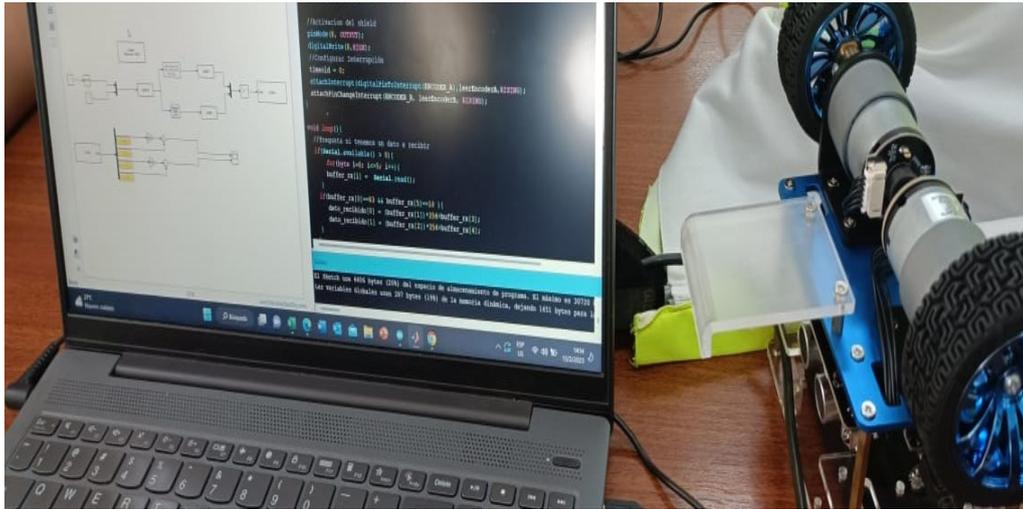


Figura 108 Identificación de la planta.

Realizado por: Yuri Torres, 2024.



Figura 109 Robot balancín en su acción equilibrista.

Realizado por: Yuri Torres, 2024.