

UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE EDUCACIÓN CIENCIA Y TECNOLOGÍA
CARRERA DE MECATRÓNICA



TEMA:

**“SISTEMA BARRA-ESFERA PARA PRÁCTICAS DE SISTEMAS DE
CONTROL”**

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN MECATRÓNICA**

AUTOR:

MARTÍNEZ RODRÍGUEZ ANDRÉS JAVIER

DIRECTORA:

ING. TOBAR SUBÍA CONTENTO LUZ MARÍA

Ibarra, 2023



UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1716704679		
APELLIDOS Y NOMBRES:	ANDRÉS JAVIER MARTÍNEZ RODRÍGUEZ		
DIRECCIÓN:	CAYAMBE		
EMAIL:	andresjavier_12@hotmail.com		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0991563158
DATOS DE LA OBRA			
TÍTULO:	SISTEMA BARRA-ESFERA PARA PRÁCTICAS DE SISTEMAS DE CONTROL		
AUTOR:	ANDRÉS JAVIER MARTÍNEZ RODRÍGUEZ		


FECHA:	12/10/2023
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	INGENIERO EN MECATRÓNICA
DIRECTOR:	MSc. TOBAR SUBÍA CONTENTO LUZ MARÍA

CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 12 días del mes de octubre de 2023

EL AUTOR:


.....

Andrés Javier Martínez Rodríguez



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE EDUCACIÓN CIENCIA Y TECNOLOGÍA

CERTIFICACIÓN

Ibarra, 10 de octubre de 2023

ING. TOBAR SUBÍA CONTENTO LUZ MARÍA

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de titulación, el mismo que se ajusta a las normas vigentes de la Unidad Académica de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

A handwritten signature in blue ink, reading "Luz María Tobar Subía Contento", is written over a horizontal dotted line.

MSc. Tobar Subía Contento Luz María

C.C.: 1002444204



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE EDUCACIÓN CIENCIA Y TECNOLOGÍA

APROBACIÓN DEL COMITÉ CALIFICADOR

El Tribunal Examinador del trabajo de titulación "SISTEMA BARRA-ESFERA PARA PRÁCTICAS DE SISTEMAS DE CONTROL" elaborado por Andrés Javier Martínez Rodríguez, previo a la obtención del título de ingeniero mecatrónico, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:

A handwritten signature in blue ink, appearing to read "Luz María Tobar Subía Contento", written over a dotted line.

MSc. Tobar Subía Contento Luz María
C.C.: 1002444204

A handwritten signature in blue ink, appearing to read "Carlos Xavier Rosero Chandi", written over a dotted line.

PHd. Rosero Chandi Carlos Xavier
C.C.: 1002515821



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE EDUCACIÓN CIENCIA Y TECNOLOGÍA

AGRADECIMIENTO

Quisiera expresar mi más sincero agradecimiento a todas aquellas personas que contribuyeron de manera significativa a la realización de esta tesis. En especial, agradezco a mi directora de tesis, MSc. Luz María Tobar, por su invaluable orientación, paciencia y apoyo constante. A mi familia y amigos, por su amor, motivación y aliento inquebrantable en cada etapa de este proceso. A la comunidad académica de la Universidad Técnica del Norte, por brindarme las herramientas y conocimientos necesarios para el desarrollo de este trabajo.

También, quiero reconocer a todos los participantes y colaboradores que compartieron su tiempo y experiencia, haciendo posible la culminación de esta investigación. Y a alguien muy especial en mi vida, que ha sido mi sostenimiento a lo largo de este camino, tu amor y comprensión han sido un motor constante en cada desafío. Tu apoyo incondicional hizo que este logro fuera posible, y por eso, Melida Morillo, quiero expresarte mi profundo agradecimiento. Tu presencia ha sido fundamental en mi vida y en este proyecto.



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE EDUCACIÓN CIENCIA Y TECNOLOGÍA

DEDICATORIA

A mi querida madre, Carmen Martínez, cuyo amor y apoyo han sido mi pilar inquebrantable. A Melida Morillo, por su paciencia y comprensión en cada paso de este camino. Dedico este trabajo a ustedes, con todo mi amor y gratitud.

También quisiera dedicar este logro a todos mis maestros y mentores, que, con su sabiduría y guía, han sido fuente de inspiración y crecimiento. A mis amigos, por su amistad eterna y momentos de alegría que aligeraron el viaje.

Además, quiero dedicar esta tesis a mi hermano, Cristian Urbano, quien, con sus virtudes y defectos, ha sido una parte fundamental de mi vida. A través de nuestra relación, he aprendido la importancia del amor incondicional y la aceptación. A pesar de nuestras diferencias, quiero que sepas que te amo profundamente y te valoro. Esta tesis es un testimonio de que las cosas se pueden lograr con amor y perseverancia.

Finalmente, esta tesis está dedicada a todos aquellos que buscan el conocimiento y la verdad, con la esperanza de que pueda contribuir, aunque sea de manera modesta, a nuestro entendimiento colectivo.

RESUMEN

Este trabajo de grado se centró en el diseño, construcción y validación de un sistema de control para manipular la posición de una esfera en una barra inclinada, utilizando un servomotor y un sensor Sharp. Los parámetros esenciales para la construcción del sistema fueron determinados meticulosamente, y se desarrollaron modelos matemáticos teóricos y experimentales utilizando herramientas como MATLAB para describir la dinámica del sistema.

La planta real construida se ajustó y validó en comparación con las simulaciones y predicciones, identificando discrepancias y realizando ajustes en los parámetros del controlador PID para mejorar la estabilidad del sistema. A pesar de los desafíos, el proyecto resultó en un sistema operativo y propuso prácticas de laboratorio utilizando la planta, abriendo caminos para futuras investigaciones y mejoras en el campo de control.

ABSTRACT

This degree work focused on the design, construction, and validation of a control system to manipulate the position of a ball on an inclined bar, using a servomotor and a Sharp sensor. The essential parameters for the construction of the system were meticulously determined, and theoretical and experimental mathematical models were developed using tools such as MATLAB to describe the dynamics of the system.

The actual plant constructed was adjusted and validated in comparison with simulations and predictions, identifying discrepancies and making adjustments to the PID controller parameters to enhance the system's stability. Despite the challenges, the project resulted in an operational system and proposed laboratory practices using the plant, paving the way for future research and improvements in the field of control.

ÍNDICE DE CONTENIDOS

IDENTIFICACIÓN DE LA OBRA	ii
CONSTANCIAS	iii
CERTIFICACIÓN	iv
APROBACIÓN DEL COMITÉ CALIFICADOR	v
AGRADECIMIENTO.....	vi
DEDICATORIA.....	vii
RESUMEN.....	viii
ABSTRACT	ix
ÍNDICE DE CONTENIDOS	x
ÍNDICE DE TABLAS	xvi
ÍNDICE DE FIGURAS.....	xvii
INTRODUCCIÓN	1
Problema.....	1
Objetivos	1
General	1
Específicos	1
Justificación.....	2
Alcance.....	2
1. CAPÍTULO I: MARCO TEÓRICO.....	3

1.1.	Sistema de procesos	3
1.2.	Sistemas de control	3
1.2.1.	Tipos de sistemas de control.....	5
1.2.1.1.	Sistema de control de velocidad	5
1.2.1.2.	Sistema de control de temperatura.....	7
1.3.	Sistema de lazo abierto y cerrado	7
1.3.1.	Sistema de lazo cerrado	8
1.4.	Tipos de diseños del sistema barra -esfera.....	9
1.4.1.	Aparato de esfera y viga modelo CE 106	9
1.4.2.	Modelo esfera viga ACROME	10
1.4.3.	Modelo esfera viga Quanser	12
1.4.4.	Modelado del experimento de esfera y viga.....	13
1.5.	Controlador PID.....	16
1.5.1.	Acción proporcional	17
1.5.2.	Acción Integral	18
1.5.3.	Acción Derivativa.....	18
1.6.	Arduino	20
1.7.	Bibliotecas Arduino	21
1.8.	Labview.....	22
2.	CAPÍTULO II: MATERIALES Y MÉTODOS.....	24

Modelo de investigación	24
2.1.1. Investigación documental	24
2.2. Diseño de investigación	26
2.3. Tipo de investigación	27
2.4. Técnicas e instrumentos	28
2.5. Preguntas de investigación	28
2.5.1. Proceso de construcción	34
2.6. Proceso de programación	36
2.6.1. Proceso de control	37
2.6.2. Envío de parámetros PID	42
2.6.3. Configuración Arduino	44
2.6.4. Función PID:	44
2.6.5. Función distancia_sharp	46
2.6.6. Filtro de media exponencial	47
3. CAPÍTULO III: RESULTADOS Y DISCUSIÓN	51
3.1 Metodología de Validación	51
3.1.1. Herramientas y software utilizados	51
3.1.1. Procedimiento de simulación	52
Modelado en Simulink:	52
Configuración de la Simulación:	52

Selección de Entradas:	53
Simulación y Registro de Datos:	53
Implementación en Arduino:.....	53
Pruebas en Tiempo Real:.....	53
Iteraciones:	53
3.1.2. Parámetros y condiciones iniciales utilizadas.	54
Parámetros del Sistema:	54
Condiciones Iniciales:	54
Condiciones Externas:.....	55
3.1.3. Descripción de las entradas utilizadas para probar el sistema.....	55
Entrada en Escalón:.....	55
Entrada Sinusoidal:	55
Entrada de Impulso:	55
Entrada Rampa:	55
Entradas Combinadas:.....	56
Entradas de Perturbación:.....	56
3.2. Resultados de Simulación	56
3.2.1. Datos Obtenidos	56
Modelo matemático.....	56
Simulación.....	60

3.2.2.	Descripción Resultados del controlador	64
	Cálculos referenciales al PID	64
3.3.	Análisis de Resultados.	68
3.3.1.	Contraste de los resultados de la simulación con las predicciones o especificaciones teóricas	68
	Observaciones Generales:	69
	Análisis Cuantitativo:	69
3.3.2.	Gráficos Comparativos	70
3.3.3.	Cálculo y Análisis de Errores	71
	Análisis de errores:	71
	Error a los 10 cm:	71
3.3.4.	Discusión de los Resultados	73
	Concordancia entre los Modelos Teórico y Experimental	73
	Estabilidad Conseguida Mediante la Ajuste de Parámetros	73
	Valiosa Información Obtenida del Método de Ziegler-Nichols	73
3.3.5.	Propuesta para prácticas de laboratorio	74
	Práctica de Laboratorio 1: Calibración de Sensores y Actuadores	75
	Práctica de Laboratorio 2: Diseño y Ajuste de un Controlador PID	76
	Práctica de Laboratorio 3: Validación Experimental	77
4.	CAPITULO IV: CONCLUSIONES Y RECOMENDACIONES.....	78

4.1. Conclusiones	78
4.2. Recomendaciones	79
BIBLIOGRAFÍA.....	81
ANEXOS.....	84
Anexo 1: Evidencia fotográfica.....	84
Anexo 2: Algoritmos	86
Anexo 3: Planos de la planta	94
Anexo 4: Datasheet	95
<i>Práctica de Laboratorio 1: Calibración de Sensores y Actuadores</i>	101
<i>Práctica de Laboratorio 2: Diseño y Ajuste de un Controlador PID</i>	102
<i>Práctica de Laboratorio 3: Validación Experimental</i>	102

ÍNDICE DE TABLAS

Tabla 1. Comparativa de medidas de distancia con el sensor sharp.....	69
Tabla 2. Comparativa de errores en la medición.....	71

ÍNDICE DE FIGURAS

Figura 1 Componentes básicos de un sistema de control.....	4
Figura 2 Esquema general de un sistema de control	5
Figura 3 Sistema de control de velocidad	6
Figura 4 Sistema de control de temperatura.....	7
Figura 5 Sistema de control en lazo abierto	8
Figura 6 Sistema de control en lazo abierto	9
Figura 7 Aparato esfera viga modelo CE106.....	10
Figura 8 Modelo esfera viga ACROME	11
Figura 9 Modelo esfera viga Quanser	12
Figura 10 Ejemplo de modelo experimental	14
Figura 11 Ejemplo sistema de control de lazo cerrado con controlador PID.....	17
Figura 12 Comportamiento del actuador de un controlador proporcional	17
Figura 13 Esquema del lazo cerrado con controlador PID	20
Figura 14 Esquema placa Arduino uno modelo estándar.....	21
Figura 15 Logotipo LabVIEW	23
Figura 16 Proceso para el desarrollo del proyecto	26
Figura 17 Viga inclinada y esfera rodando	29
Figura 18 Elementos principales en el modelado del sistema barra esfera.....	32
Figura 19 Modelado 3D del sistema barra esfera.....	32
Figura 20 Ensamble del sistema barra esfera	34
Figura 21 Instalación del servo motor.....	35

Figura 22 Ubicación de la esfera en los rieles.....	35
Figura 23 Ubicación de la placa de Arduino.....	36
Figura 24 Configuración de puerto serial.....	38
Figura 25 Configuración del tiempo en espera	38
Figura 26 Vaciado del puerto serial Configuración del tiempo en espera	39
Figura 27 Ciclo principal de control	40
Figura 28 Conteo de datos en el puerto serial	40
Figura 29 <i>Recepción de la posición de la esfera</i>	41
Figura 30 <i>Conteo de la posición del servomotor</i>	41
Figura 31 <i>Configuración de parámetros PID</i>	42
Figura 32 <i>Configuración del setpoint</i>	43
Figura 33 <i>Configuración de parámetros Alpha</i>	44

INTRODUCCIÓN

Problema

En la actualidad, el estudio de los sistemas de control es un factor esencial en el área de la automatización de sistemas industriales y domésticos que brindan mayor seguridad y confiabilidad a los procesos.

Los laboratorios de la carrera de ingeniería mecatrónica de la Universidad Técnica Del Norte tienen algunos módulos y máquinas que permiten a los estudiantes e investigadores desarrollar prácticas y generar conocimientos.

Actualmente los módulos existentes se limitan al uso exclusivo de software de propietario. Las desventajas que conlleva el uso de estos módulos son la incompatibilidad con otro software diferente y la falta de independencia tecnológica, además nos son reproducibles ni adaptables, esto quiere decir que la innovación y la mejora de los diseños de forma colaborativa está restringida.

Por tales motivos nace la idea de aportar con un robot basado en un sistema barra-esfera que además de ser compatible con software propietario también podrá ser usado con software libre y hará más sencilla la manera de aprender involucrando a los estudiantes directamente con los temas de control a través de la práctica.

Objetivos

General

Construir un sistema Barra-Esfera orientado al aprendizaje de conceptos de control.

Específicos

- Determinar los parámetros necesarios para la construcción del sistema.
- Realizar el modelamiento matemático del sistema.
- Construir el mecanismo en base al modelo.

- Validar el sistema diseñado contra simulación.

Justificación

En el presente proyecto se planea construir un sistema barra-esfera y proponer varias prácticas de laboratorio orientadas al aprendizaje de conceptos sobre sistemas de control. También se podrá usar para enseñar otros temas que no estén incluidos en la guía práctica.

El prototipo permitirá al estudiante concentrarse en el desarrollo de las prácticas librándolo de los detalles de implementación del sistema que no tienen nada que ver con el área de control.

Se ofrece una alternativa que no limita a los estudiantes e investigadores a trabajar únicamente con software y hardware propietario, sino que cuenta con la versatilidad de ser adaptado a diferentes entornos y lenguajes de desarrollo lo cual que es una característica muy valorada cuando se habla de investigación.

El prototipo de este sistema estará presto a su mejora, innovación y reproducción con el fin de mejorar la educación e investigación de futuros estudiantes y docentes de la carrera de Ingeniería Mecatrónica de la Universidad Técnica Del Norte.

Alcance

Se realizará el diseño, la construcción y el control de un sistema Barra-Esfera con hardware libre, permitiendo la compatibilidad con software libre y de propietario.

Se desarrollará un manual de uso del sistema.

Se usará una estrategia de control para validar el modelo.

Se implementará el sistema de control en la placa Arduino.

El sistema funcionará a través de la recepción de parámetros que permitan modificar el comportamiento del sistema cuya interfaz se conectará a través de comunicación serial.

El sistema barra-esfera gozará de una estructura de fácil reproducción.

CAPÍTULO I: MARCO TEÓRICO

1.1. Sistema de procesos

Un sistema de control de procesos consta de un grupo de unidades en diferentes secuencias. Pueden ser eléctricos, neumáticos, hidráulicos, mecánicos, etc. El tipo de equipo depende en gran medida de los objetivos a alcanzar (Autycom, 2023). Pero solo con estos dispositivos no se crea un sistema de control, sino que se sigue la lógica de al menos 3 elementos básicos:

- La variable para controlar.
- Mecanismo (estructura).
- Punto de referencia o punto de consigna

En el proceso de gestión de lotes, es esencial que el sistema utilice tres componentes clave. Se considera el producto almacenado en grandes contenedores como variable a controlar. El "setpoint" o punto de referencia define cuánto se llena, y el actuador, que en este caso es una bomba, ya sea mecánica o eléctrica, lleva a cabo la tarea de llenado.

1.2. Sistemas de control

La automatización en el control ha sido esencial en el progreso tecnológico y científico. Ha influido en áreas como la aeroespacial, misiles dirigidos y la robótica. Los progresos en la teoría y aplicación del control automático establecen el fundamento para el rendimiento ideal de sistemas dinámicos, perfeccionando y optimizando procesos para lograr excelentes resultados. Además, simplifican tareas manuales repetitivas y otras operaciones, permitiendo entender y dominar las fuerzas naturales en pro de la sociedad (Contreras, 2015).

La tecnología de control se basa en los fundamentos de la teoría de retroalimentación y el análisis de sistemas lineales, integrando la teoría de redes y comunicaciones; Por tanto, la teoría de control no se limita a un campo específico de la ingeniería, sino que es aplicable a la ingeniería

aeronáutica, civil, química, mecánica y eléctrica, analizando así la dinámica de todo tipo de sistemas y mejorando su control (Carrillo, 2011).

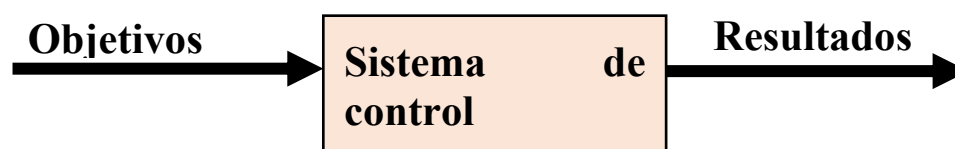
Los componentes principales de un sistema de control pueden ser describe:

- a) Objetivos de control
- b) Componentes del sistema de control
- c) Resultados o salida

Estos objetivos suelen identificarse como entradas o señales de referencia. Por otro lado, los resultados se interpretan como salidas o variables reguladas. En esencia, la finalidad de un sistema de control es gestionar la salida de forma precisa, orientando los componentes del sistema hacia la señal de entrada. (Carrillo, 2011).

Figura 1

Componentes básicos de un sistema de control



Los elementos principales de un sistema de control permiten la manipulación de los siguientes elementos:

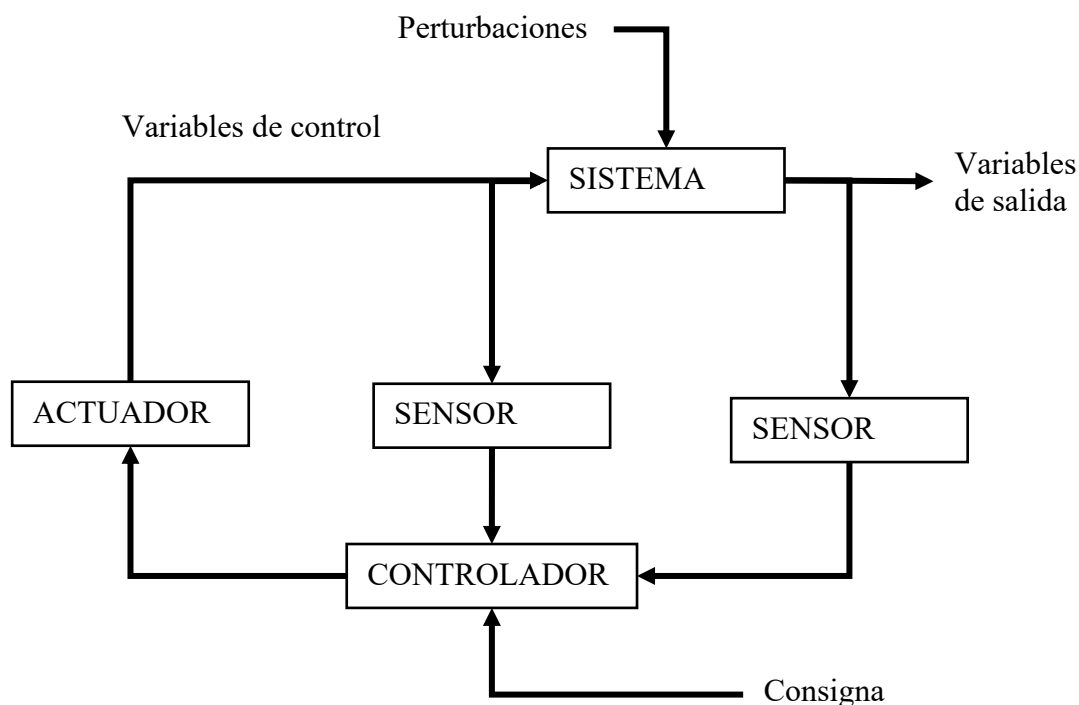
- Sensor: permite conocer valores de variables del sistema a evaluar
- Controlador: interpreta la señal que le llega, su origen ya veces su intensidad; La compara con una determinada señal de referencia y genera otra señal que hace funcionar otros elementos de la máquina. Por ejemplo, en una lavadora, esta función la realiza un programador. Los últimos cuentan con microprocesadores que calculan la cantidad de ropa, el grado de suciedad, el tipo de tejido y el método de lavado más adecuado en cada momento.

- Actuador: mecanismo de ejecución calculado por un controlador que modifica las variables de control. En la figura 2, se ilustra el sistema de control genérico.

En el caso de sistemas de control mecánico, esto significa que las señales en dicho sistema son generadas por los mecanismos encargados de mover la secuencia o transmitir potencia.

Figura 2

Esquema general de un sistema de control



Fuente: (Sira et al., 2004)

1.2.1. Tipos de sistemas de control

Existen varios tipos de sistemas de control, que se clasifican según su estructura y características. A continuación, se presentan algunos de los tipos más comunes:

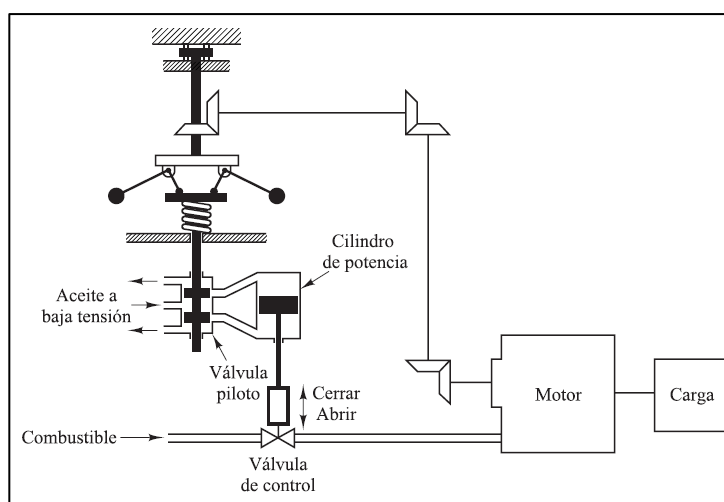
1.2.1.1. Sistema de control de velocidad

El principio básico de un controlador de velocidad de vatios de una máquina se muestra en el diagrama de la figura. La cantidad de combustible entregado a la máquina se regula en función de la diferencia entre la velocidad deseada y la real.

El proceso se puede describir de la siguiente manera: El controlador de velocidad se configura de modo que no fluya aceite a presión hacia ninguno de los cilindros de potencia a la velocidad deseada. Si la velocidad real cae por debajo de la velocidad deseada debido a la perturbación, la fuerza centrífuga más baja del gobernador hace que la válvula de control se mueva hacia abajo, agregando combustible y llevando la velocidad del motor al valor deseado. Por otro lado, cuando la velocidad del motor aumenta por encima de la deseada, la mayor fuerza centrífuga del gobernador hace que la válvula de control se mueva hacia arriba. Esto reduce el suministro de combustible y reduce la velocidad del motor al valor deseado (Ogata, 2010).

Figura 3

Sistema de control de velocidad



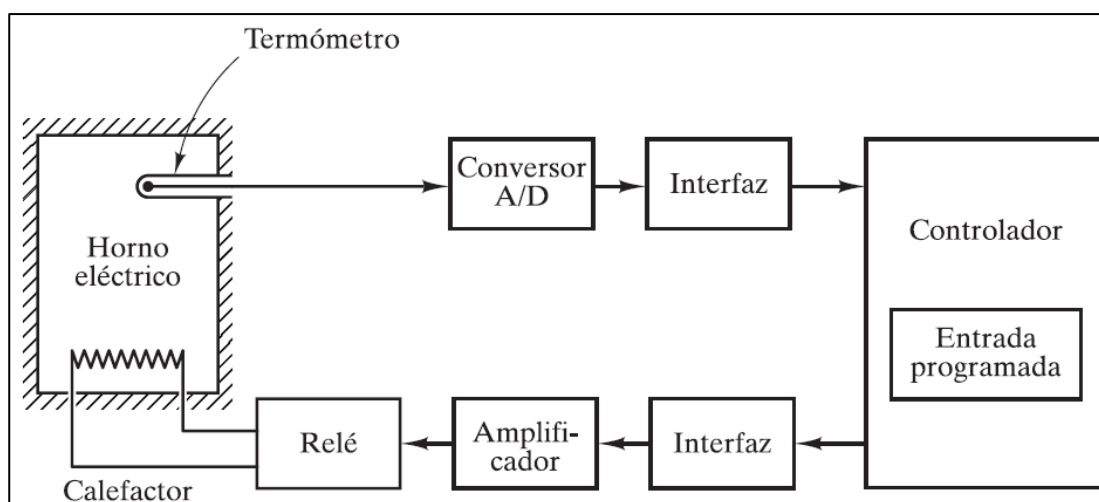
En este sistema de control de velocidad, la máquina es la planta o sistema controlado, mientras que la velocidad de la máquina es la variable controlada. La diferencia entre la velocidad deseada y la real se considera una señal de error. La señal de control correspondiente a la cantidad de combustible alimentada a la máquina se denomina señal de activación. Una perturbación se refiere a una entrada externa que cambia la variable controlada, como un cambio inesperado en la carga (Ogata, 2010).

1.2.1.2. Sistema de control de temperatura

La Figura 4 muestra el esquema de control de temperatura del horno eléctrico. La temperatura del horno eléctrico se mide con un termómetro, que es un dispositivo analógico. La temperatura analógica se convierte en temperatura digital utilizando un convertidor A/D. La temperatura digital se transfiere al controlador a través de la interfaz. Esta temperatura digital se compara con la temperatura de entrada programada, y en caso de desviación (error), el controlador envía una señal al calentador a través de la interfaz, amplificador y relé para obtener el valor deseado de la temperatura del horno (Ogata, 2010).

Figura 4

Sistema de control de temperatura



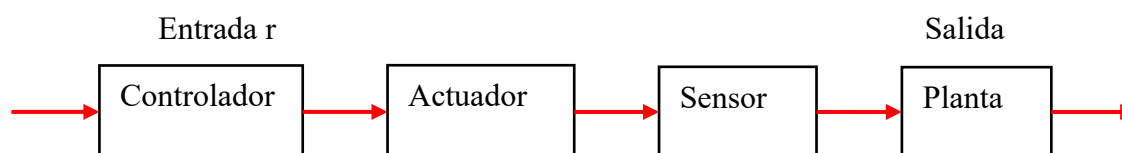
1.3. Sistema de lazo abierto y cerrado

Una técnica muy utilizada en el análisis de sistemas es la representación de los elementos del sistema mediante diagramas de bloques. Esta presentación describe visualmente la secuencia lógica de operaciones que permite el modelado del sistema y la identificación de posibles errores de diseño. El siguiente diagrama de bloques de la figura 5, al que se denomina "circuito de control", muestra la interconexión de los bloques que contienen los elementos de control. En este caso, el

lazo de control describe un "lazo de control abierto", también conocidos como sistemas de control no retroalimentados, no utilizan información de retroalimentación para ajustar o corregir el proceso de control. En este tipo de sistema, la salida no tiene influencia directa sobre la entrada o el controlador. En este tipo de configuración, la entrada del sistema afecta directamente al controlador, que a su vez ejerce el efecto deseado sobre la salida y a través del actuador (Morales & Ramírez, 2013).

Figura 5

Sistema de control en lazo abierto



1.3.1. Sistema de lazo cerrado

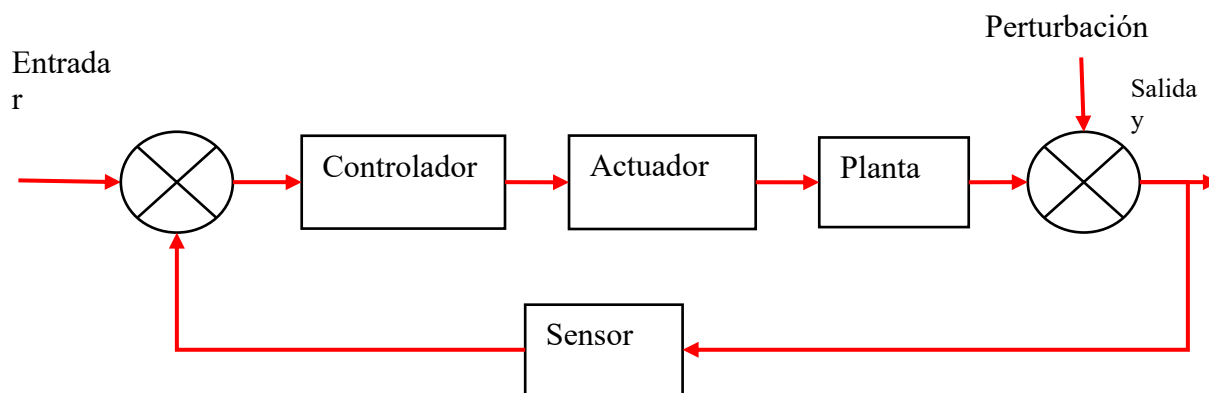
La dinámica de otro lazo de control, representado en la Figura 6, que se conoce como "lazo cerrado de control". Los sistemas de circuito cerrado se utilizan ampliamente en una variedad de aplicaciones, desde el control de procesos industriales hasta la automoción, aeronaves, sistemas de navegación y muchos otros sistemas de control. En este tipo de configuración, se toma decisiones de control basadas en la información proporcionada por los sensores para mover el actuador y alcanzar la posición deseada. La retroalimentación negativa desempeña un papel importante en el lazo cerrado, ya que garantiza la estabilidad del sistema (Morales & Ramírez, 2013).

En la práctica, las mediciones de los sensores se filtran para eliminar el ruido, las perturbaciones externas y los errores de calibración. Además, la posición no suele ser conocida directamente, sino que se estima utilizando herramientas matemáticas como los observadores de estado y el filtro de Kalman (Kalman, 1960 y Stengel, 1994). Estos conceptos se abordan en cursos avanzados o de posgrado, y es importante tenerlos en cuenta para evitar confusiones. En este libro

se focaliza en considerar las variables físicas en relación con la referencia establecida (Morales & Ramírez, 2013).

Figura 6

Sistema de control en lazo abierto



1.4. Tipos de diseños del sistema barra -esfera

Existen diferentes tipos de diseños para el sistema de barra-esfera en el ámbito de los sistemas de control. A continuación, se presentan algunos de los tipos de diseños más comunes:

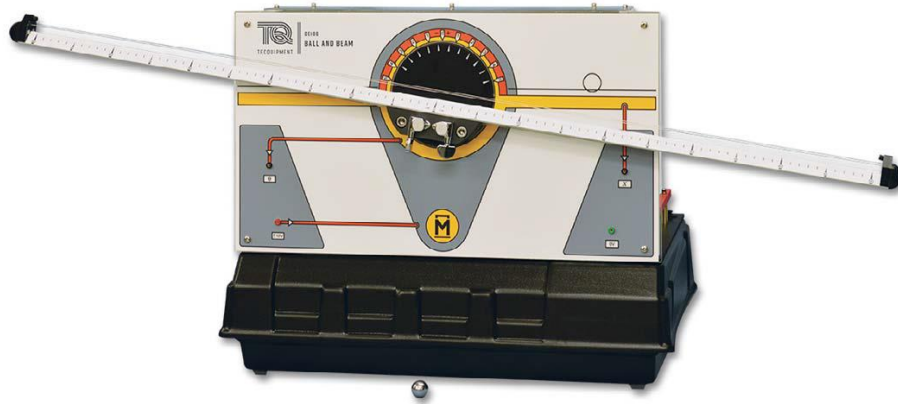
1.4.1. Aparato de esfera y viga modelo CE 106

El aparato de esfera y viga muestra los problemas de control de los sistemas inestables, por ejemplo, un cohete o un misil durante el lanzamiento, que necesita un control activo para evitar que el misil se vuelva inestable y vuelque. El aparato tiene una esfera de acero que rueda libremente sobre dos cables paralelos tensados situados en una viga que pivota en su centro. Un servomotor controla el ángulo de la viga y unos sensores miden el ángulo de la viga y la posición de la esfera. El problema básico de control consiste en variar el ángulo de la viga para controlar la posición de la esfera. El sistema es un integrador doble, por lo que es inestable por naturaleza. Necesita un control de realimentación activo mediante métodos de avance de fase (Tecquipment, 2021)

El diseño flexible del equipo permite al usuario desarrollar muchos otros ejercicios de análisis y control a sus necesidades. Es adecuado para experimentos de control ampliados o avanzados, y es ideal para proyectos de trabajo de estudiantes

Figura 7

Aparato esfera viga modelo CE106



Nota. Aparatos autónomos de sobremesa para demostrar los principios básicos y avanzados del control en sistemas naturalmente inestables. Tomado de (Tecquipment, 2021)

1.4.2. Modelo esfera viga ACROME

El sistema de esfera y viga consiste en una viga larga junto con una esfera que rueda hacia delante y hacia atrás en la parte superior de la viga. La viga larga se inclina mediante un servo. Nuestro sistema de esfera y viga permite practicar la teoría de control automático en un experimento de bucle cerrado. El control de la posición de una esfera en una viga es uno de los problemas clásicos de la teoría de control (Acrome, 2015)..

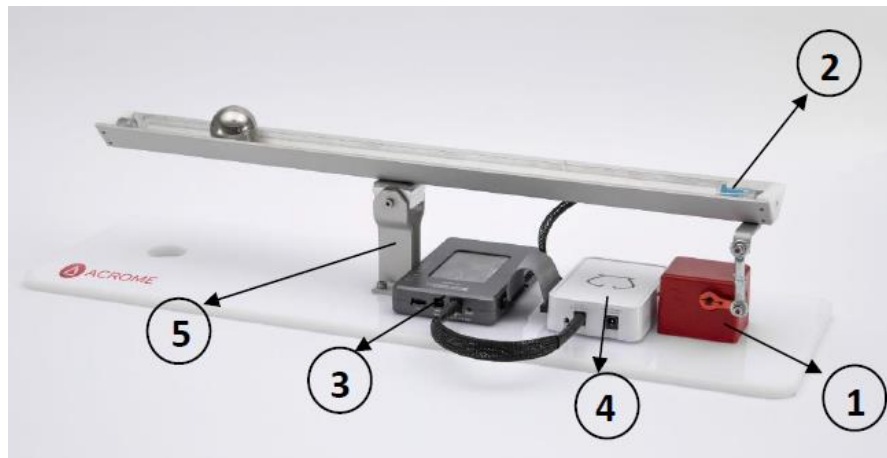
El control de la posición de una esfera sobre una viga es uno de los problemas clásicos de la teoría de control. Ball and Beam System se puede utilizar para una amplia gama de implementaciones de diseño de sistemas de control, desde controladores lineales básicos hasta

métodos no lineales avanzados. Los estudiantes pueden comprender los enfoques de diseño de sistemas con el método de aprendizaje práctico de Ball and Beam (Acrome, 2015).

El sistema de esfera y viga puede utilizarse para una amplia gama de implementaciones de diseño de sistemas de control, desde controladores lineales básicos hasta métodos no lineales avanzados. Los estudiantes son capaces de comprender los enfoques de diseño de sistemas con el método "aprender haciendo" de Ball and Beam. Con la ayuda del software del curso, pueden distinguir los efectos de las linealizaciones, las suposiciones y los errores de modelado debido a las diferencias entre las simulaciones y los experimentos del mundo real (Acrome, 2015).

Figura 8

Modelo esfera viga ACROME



Nota. Los componentes: 1. Servo RC 2. Sensor potenciómetro 3. MYRIO 4. Caja de alimentación 5. Mecánica de la viga. Tomado de (Acrome, 2015)

1.4.3. *Modelo esfera viga Quanser*

Es una herramienta ideal para introducir varios conceptos de control en sistemas inestables de lazo cerrado. Puede utilizarse para demostrar desafíos de control del mundo real, como el control de balanceo de aeronaves. El módulo Ball and Beam se conecta a la unidad base de un servo rotatorio. Consiste en una barra de acero colocada en paralelo con una resistencia de alambre bobinado de níquel-cromo, que forma una pista sobre la cual una esfera de metal puede rodar libremente. Esta pista funciona como un potenciómetro, emitiendo un voltaje proporcional a la posición de la esfera (Quanser, 2023).

Al acoplarlo a la unidad base del servo rotatorio, se puede controlar el ángulo de inclinación del haz mediante el ajuste del ángulo del engranaje del servo. El módulo Ball and Beam se puede operar de manera independiente, permitiendo controlar la posición de la esfera a través de una interfaz de usuario (Quanser, 2023).

Figura 9
Modelo esfera viga Quanser



Nota. La plataforma de movimiento rotatorio puede introducir conceptos de control de sistemas de bucle cerrado inestable. Tomado de (Quanser, 2023)

1.4.4. Modelado del experimento de esfera y viga

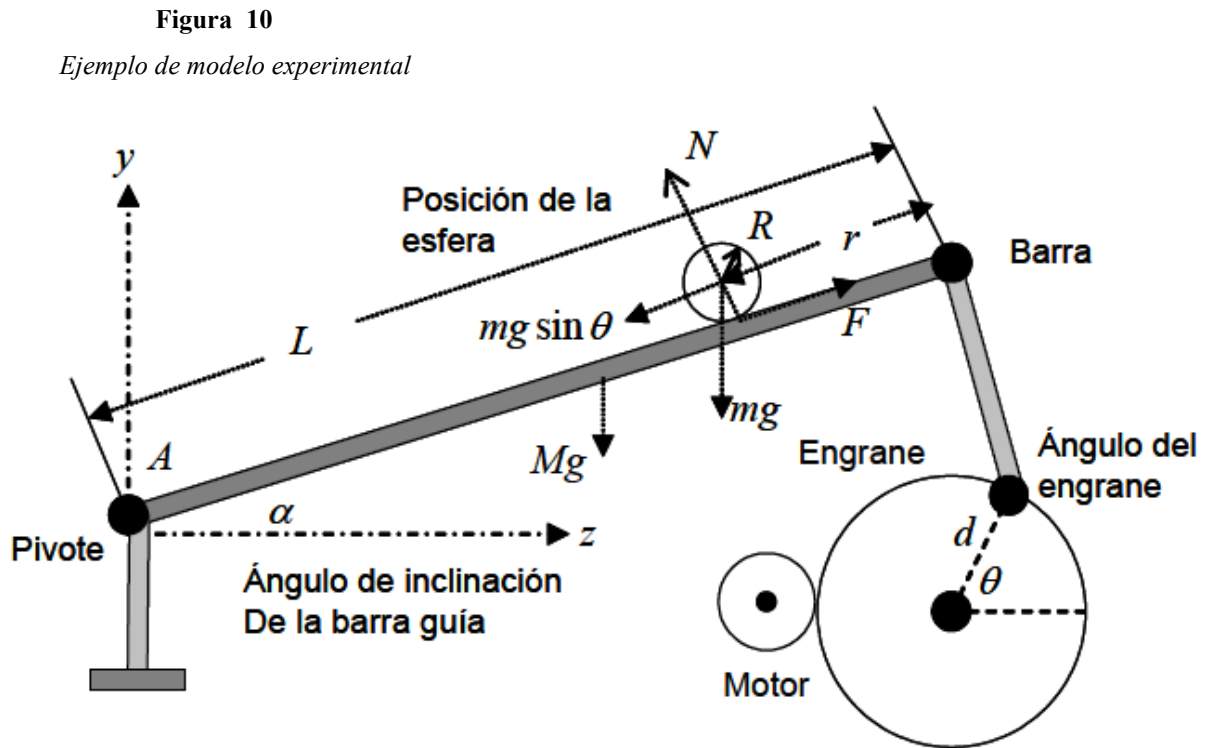
Se coloca una esfera en la viga, también conocido como el sistema de barra-esfera, se basa en la dinámica de una esfera que se desplaza sobre una viga horizontal. Este modelo se utiliza frecuentemente en el ámbito de los sistemas de control para ilustrar conceptos teóricos y realizar experimentos prácticos, ver figura 4, donde se le permite rodar a lo largo de la viga con 1 grado de libertad. Una palanca está unida a la viga en un extremo y una transmisión de potencia en el otro extremo. A medida que el engranaje del servo gira en un ángulo de θ , la palanca cambia el ángulo del haz en α . Al cambiar el ángulo de la vertical, la gravedad hace que la esfera ruede por la barra. El controlador de este sistema está diseñado para manipular la posición de la esfera (Ortiz, 2004, p. 35).

El modelado de prueba de esfera y viga puede realizar utilizando los principios de la mecánica y la dinámica. Los siguientes son los principales elementos que intervienen en el modelado:

Esfera: Una esfera representa una masa que se mueve en un radio. Su ubicación y velocidad son variables importantes en el sistema. Las fuerzas que actúan sobre la esfera son la gravedad, la fricción y la fuerza de dirección aplicada (Buza et al., 2020).

- **Viga:** La viga es la estructura horizontal a lo largo de la cual viaja la esfera. El radio y la masa son consideraciones importantes en el modelado. Además, el radio puede tener limitaciones físicas, como limitaciones en el movimiento o limitaciones en la dirección del movimiento de la esfera.
- **Actuador:** El actuador es el elemento encargado de aplicar la fuerza de control al sistema. Puede ser un motor, un electroimán o algún otro dispositivo que pueda afectar a la esfera.

- Sensor: El sensor es el componente encargado de medir la posición y velocidad de la esfera. Puede ser un sensor de posición, un sensor de inclinación o cualquier otro dispositivo que pueda registrar variables relevantes del sistema (Buza et al., 2020).



Nota. el sistema barra-esfera es utilizado en aplicaciones académicas para probar y diferenciar los algoritmos de control. Tomado de (Ortiz, 2004)

Para este modelo se supone que la esfera se desliza con fricción despreciable entre la viga y la esfera, las constantes y variables para este ejemplo se definen de la siguiente manera:

masa de la esfera: 0,11 kg

R radio de la esfera: 0,015 metros

d desplazamiento del brazo de palanca: 0,03 metros

aceleración gravitacional g : 9,8 m/s²

L longitud de la viga: 1,0 metro

J momento de inercia de la esfera: $9.99e-6 \text{ kgm}^2$

r coordenada de la posición de la esfera

alfa: coordenada del ángulo del haz

theta: ángulo de engranaje servo

se considera la ecuación lagrangiana de movimiento de la esfera:

$$\theta = \left(\frac{J}{R^2} + m \right) \ddot{r} + mg \sin \alpha - m r \dot{\alpha}^2$$

La linealización para esta ecuación viene dada por el ángulo de haz, $\alpha = 0$ se obtiene la aproximación lineal del sistema:

$$\left(\frac{J}{R^2} + m \right) \ddot{r} = -mg\alpha$$

La ecuación que relaciona el ángulo de haz con el del engrane se aproxima como lineal mediante la ecuación:

$$\alpha = \frac{d}{L} \theta$$

Luego se sustituye la ecuación:

$$\left(\frac{J}{R^2} + m \right) \ddot{r} = -mg \frac{d}{L} \theta$$

Obteniendo:

$$\left(\frac{J}{R^2} + m \right) R(s) s^2 = -mg \frac{d}{L} \theta(s)$$

Ordenando términos, finalmente se obtiene:

$$P(s) = \frac{R(s)}{\theta(s)} = -\frac{mgd}{L \left(\frac{J}{R^2} + m \right)} \frac{1}{s^2} \left[\frac{\text{m}}{\text{rad}} \right]$$

1.5. Controlador PID

El controlador PID tiene como objetivo reducir el error en estado estacionario de un sistema mediante el ajuste de tres parámetros: proporcional, integral y derivativo. Estos parámetros, que son ganancias, se ajustan en función del análisis de la respuesta en frecuencia y la ubicación de las raíces del sistema a controlar (Mogrovejo, 2017).

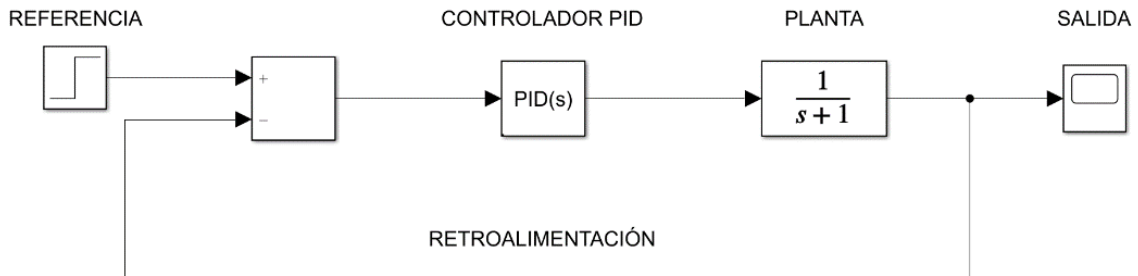
Un controlador PID se basa en la sintonización de tres constantes: una ganancia proporcional (KP), una ganancia integral (KI) o tiempo integral (Ti) y una ganancia derivativa (KD) o tiempo derivativo (Td). Estas constantes trabajan juntas para reducir el error en estado estacionario de un proceso específico, tomando en consideración la información del error en el tiempo presente, así como los errores previos y anticipándose a los errores futuros (Mogrovejo, 2017, p. 8)

En la figura 11, se muestra la retroalimentación de la salida del sistema, la cual se compara con una referencia para determinar el error en estado estacionario. Este error es corregido mediante los parámetros del controlador PID. Es importante destacar que tanto la referencia como la señal de retroalimentación es de la misma naturaleza (Guilcamaigua & Villacis, 2022).

El control PID es ampliamente utilizado en una variedad de aplicaciones debido a su capacidad para proporcionar un control preciso y estable. Además, es importante tener en cuenta que la sintonización adecuada de los parámetros PID es crucial para lograr un rendimiento óptimo del controlador. Existen diferentes métodos y técnicas para la sintonización de PID, como el método de Ziegler-Nichols, el método de ajuste por respuesta al escalón, entre otros, que permiten adaptar el controlador a las características específicas del sistema a controlar (Guilcamaigua & Villacis, 2022).

Figura 11

Ejemplo sistema de control de lazo cerrado con controlador PID



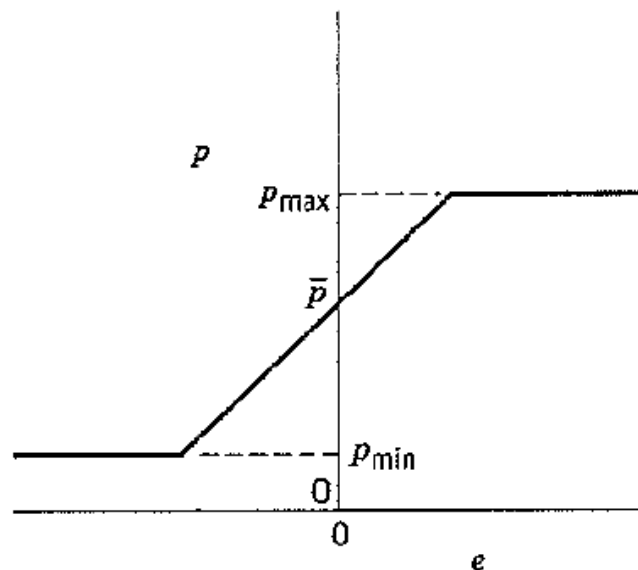
Nota. Los parámetros del PID, definen el error en base a los antecedentes y pronósticos del mismo error. Tomado de (Guilcamaigua & Villacis, 2022)

1.5.1. Acción proporcional

Un elemento proporcional, también conocido como "ganancia proporcional", representa la relación entre el cambio en la señal de salida y el cambio en la señal de entrada. La ganancia relativa (K_p) se calcula a partir del error, donde el error es la diferencia entre la referencia y la variable de salida. Este parámetro de sintonización (K_p) no tiene unidades (Carrasco, 2020).

Figura 12

Comportamiento del actuador de un controlador proporcional



Nota. tomado de (Dale et al., 2004)

La ganancia proporcional se ajusta de acuerdo con los requisitos de la salida del sistema, incluido su signo. Esto significa que el error puede aumentar o disminuir dependiendo de lo que el sistema intente lograr. Es importante tener en cuenta que la ganancia proporcional no garantiza un error cero, es decir, la diferencia entre la señal de referencia y la variable de salida, a menos que el proceso tenga integrado un integrador puro. Algunos procesos utilizan un grupo proporcional en lugar del control de ganancia proporcional, que es el inverso de la ganancia proporcional (Carrasco, 2020). El ancho de banda relativo se define de la siguiente manera:

$$PB = \frac{100 \%}{Kp}$$

1.5.2. Acción Integral

Un componente integrado del sistema de control asegura que el error de balance sea cero, lo que significa que la variable de salida es igual al valor de referencia establecido. Un controlador integral se basa en la integración de la señal de error y se utiliza junto con un controlador de ganancia proporcional. Su parámetro de sintonización se denota T_i y generalmente se expresa:

$$u_{(t)} = k * e_{(t)} + k \int_0^t \frac{e_{(t)}}{T_i} dt$$

Para asegurar la ausencia de errores, es necesario realizar una evaluación estable donde la señal $u(t)$ se mantenga constante. Esto significa que el sistema está en equilibrio y el controlador integrado asegura el reinicio automático si la señal $u(t)$ se mantiene constante en el estado estacionario.

1.5.3. Acción Derivativa

El componente derivativo en un sistema de control depende de la derivada de la señal de error y se utiliza en combinación con un controlador de ganancia proporcional. Esta relación se expresa en la siguiente ecuación

$$u_{(t)} = k * e_{(t)} + K * T_d * \frac{de_{(t)}}{dt}$$

El parámetro de sincronización del componente derivado del sistema de control se denota como T_d y generalmente se expresa en segundos. La componente derivada tiene la propiedad de amplificar el ruido con respecto a la frecuencia. Esto puede tener un efecto directo sobre la variable de control, ya que el ruido de alta frecuencia puede aumentar el valor de la variable de control. Para neutralizar este efecto, es necesario agregar un filtro de paso bajo adicional para suprimir la alta ganancia del componente derivado. Esto ayuda a reducir el efecto del ruido no deseado en el sistema de control. Tres componentes, proporcional, integral y derivativo, juntos forman un controlador PID. La parte proporcional se ajusta de acuerdo con los requisitos de salida del sistema para ajustar el nivel de error deseado. El componente integrado garantiza un error cero en el estado estable. El componente derivado asegura la estabilidad del sistema en estado estacionario.

Un controlador PID sigue una ecuación específica (continuación de la ecuación original) para calcular la señal de control $u(t)$.

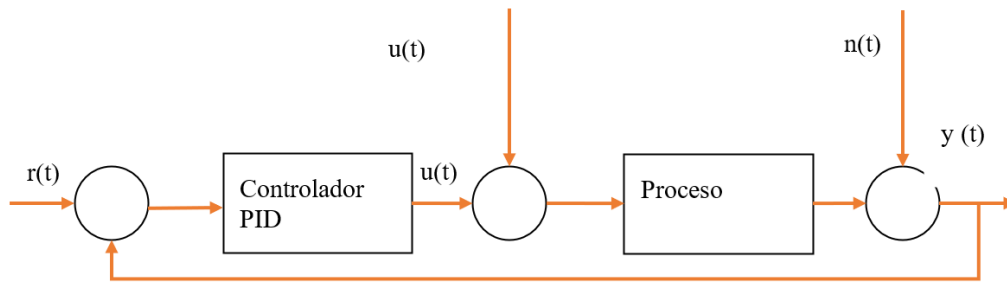
$$G_{PID} = u_{(t)} = K * e_{(t)} + K \int_0^t \frac{e_{(t)}}{T_i} dt + K * T_d \frac{de_{(t)}}{dt}$$

$$u_{(t)} = K \left[e_{(t)} + \int_0^t \frac{e_{(t)}}{T_i} dt + T_d \frac{de_{(t)}}{dt} \right]$$

Por lo tanto, se tiene la siguiente representación del control PID

Figura 13

Esquema del lazo cerrado con controlador PID



Nota. Adaptado de diseño e implementación de un control PID IMC. tomado de (Carrasco, 2020)

1.6. Arduino

Arduino es una plataforma de creación de prototipos electrónicos de código abierto basada en hardware y software flexible y fácil de usar. Está destinado a artistas, diseñadores, aficionados y cualquier persona interesada en crear objetos o entornos interactivos (Sosa, 2018).

Además, Arduino puede capturar señales físicas del entorno al recibir entradas de varios sensores, como temperatura, presión, intensidad de luz, nivel, etc. También puede controlar su entorno con actuadores como luces, motores, relés y más. Para ello, Arduino dispone de entradas y salidas analógicas y digitales que permiten la conexión a sensores y actuadores. El microcontrolador de la placa está programado en el lenguaje de programación Arduino (basado en Wiring) y en el entorno de desarrollo Arduino (basado en Processing). Los proyectos de Arduino pueden funcionar de forma independiente o interactuar con el software que se ejecuta en una computadora, como Flash, Processing, MaxMSP y otros (Sosa, 2018).

Las placas Arduino se pueden ensamblar a mano o comprar preensambladas, y el software se puede descargar de forma gratuita. Los diseños de referencia de hardware, incluidos los archivos CAD, están disponibles con una licencia de código abierto, lo que significa que puede personalizarlos para satisfacer sus necesidades. Hay una amplia gama de placas que incluyen

Arduino UNO, Arduino MEGA, Arduino DUE, Arduino COOKBOOK, Arduino DUEMILANOVE, Arduino ETHERNET, Arduino FIO. Este tipo de placa tiene: botón de encendido, fuente de alimentación, conexión USB, tres tierras, botón de reinicio (Muzo, 2018, p. 19).

Figura 14

Esquema placa Arduino uno modelo estándar



Nota. Tomado de (Xataka, 2022)

1.7. Bibliotecas Arduino

El entorno Arduino se amplía mediante bibliotecas, como la mayoría de las plataformas de programación. Estas bibliotecas brindan funcionalidad adicional que se puede usar en bocetos, como trabajar con hardware específico o procesar datos de manera más eficiente. Si desea utilizar la biblioteca en un boceto, seleccione "Importar biblioteca" en el menú "Bosquejo".

Varias bibliotecas están instaladas de forma predeterminada en el IDE de Arduino, pero es posible descargar otras bibliotecas o incluso crear las suyas propias. Para obtener más información sobre cómo instalar bibliotecas, siga las instrucciones. También hay un tutorial que explica cómo escribir bibliotecas personalizadas. Es necesario consultar la Guía de estilo de API para aprender a crear una API de estilo Arduino clara y coherente para la biblioteca que está desarrollando (Ruiz, 2016).

Las bibliotecas oficiales de Arduino:

- Robótica
- Comunicación
- Conectividad
- Biblioteca de la familia nano
- Memoria
- Mostrar
- Biblioteca familiar MKR
- Audio
- USB
- Otro
- Bibliotecas retiradas(Ruiz, 2016).

1.8. Labview

LabVIEW es un software de diseño desarrollado por National Instruments (NI) que facilita la programación de instrumentos virtuales utilizados en aplicaciones que requieren monitoreo, adquisición de datos y control de procesos. Proporciona acceso rápido y fácil a la información sobre los procesos que desea monitorear. LabVIEW fue creado en 1976 y lanzado en 1986, inicialmente solo disponible para computadoras MAC. Actualmente está disponible en plataformas como Windows, UNIX, MAC y Linux (Romero, 2021).

LabVIEW utiliza un método de programación gráfica que le permite visualizar todos los aspectos de una aplicación, desde la configuración del hardware hasta los datos de medición y la depuración. Esta visualización simplifica la integración de dispositivos de medición de diferentes

fabricantes, representa lógica compleja como diagramas, desarrolla algoritmos de análisis de datos y permite el diseño de interfaces de usuario adaptadas para la planificación (Romero, 2021, p. 22).

La principal ventaja de LabVIEW lograda por National Instruments es una reducción significativa en el tiempo de desarrollo del programa. El software gestiona de forma eficaz los recursos informáticos a través de un entorno de trabajo rápido y sencillo diseñado específicamente para aplicaciones que implican adquisición de datos del entorno físico, análisis, control y ejecución de operaciones (Romero, 2021, p. 22).

Figura 15

Logotipo LabVIEW



Nota. El sitio web de National Instruments tiene una extensa página de ayuda de LabVIEW.

Tomado de (AALBORG, 2021)

2. CAPÍTULO II: MATERIALES Y MÉTODOS

En este acápite se desarrolló la parte práctica del proyecto de investigación con base en los referentes teóricos del capítulo anterior, se procede al diseño, modelado y aplicación mediante pruebas de funcionamiento necesarios para el normal funcionamiento del estudio barra esfera.

Modelo de investigación

El objetivo de esta investigación es realizar el sistema de control para la barra-esfera que va a ser utilizado en el ámbito educativo, en el laboratorio de control o investigación de la facultad para validar algoritmos de control propuestos o realizados por los estudiantes de la carrera, también se utiliza para enseñar conceptos y teorías relacionados con los fundamentos de sistemas de control, retroalimentación, estabilidad y la respuesta en frecuencia. Finalmente es una plataforma práctica y dual intuitiva para experimentar los sistemas de control facilitando la comprensión y aprendizaje de los principios y conceptos dentro de este campo de estudio.

2.1.1. Investigación documental

La investigación documental es el proceso de búsqueda, recuperación, análisis, crítica e interpretación de información secundaria, es decir, datos previamente obtenidos y registrados por otros investigadores en fuentes documentales, ya sean impresas, audiovisuales o electrónicas. Como con cualquier investigación, el objetivo de este método es contribuir al desarrollo de nuevos conocimientos (Fidias, 2016). Por lo tanto, la investigación documental llevo un proceso sistemático mediante la revisión de libros, tesis, artículos científicos, catálogos y fichas de proveedores que contribuyen con las bases teóricas del proyecto.

Para el normal desarrollo se utiliza la metodología de cascada, también conocida como el modelo de ciclo de vida en cascada, es un enfoque lineal y secuencial para el desarrollo de software. Esta metodología divide el proceso de desarrollo en fases bien definidas, donde cada fase se inicia

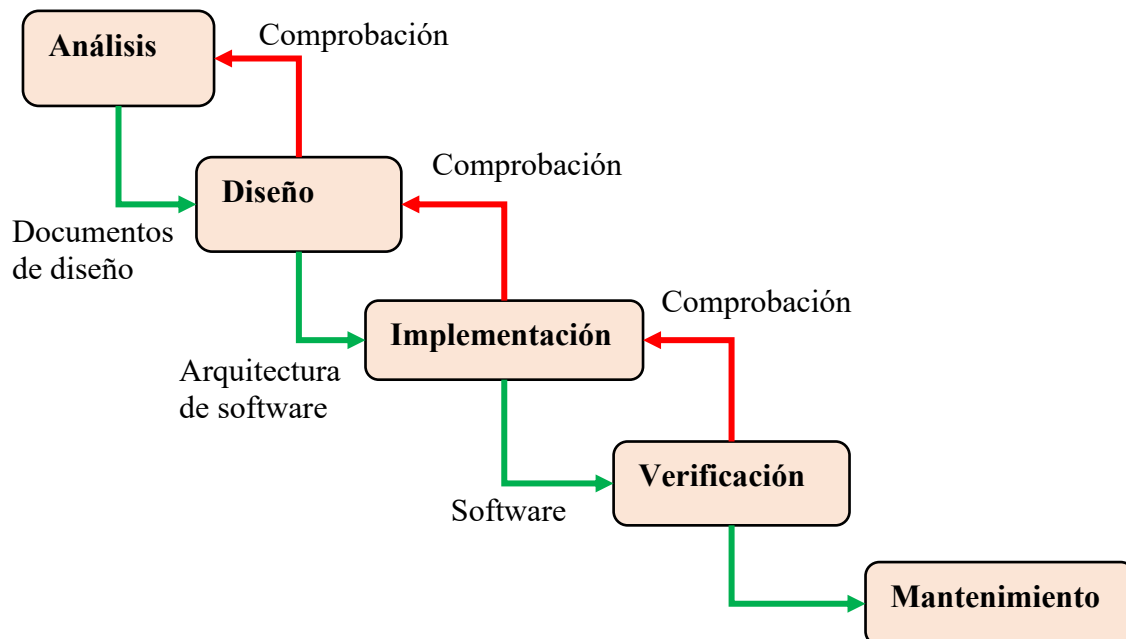
solo después de que la anterior se haya completado (Ionos, 2019). A continuación, se describe las fases principales de la metodología de cascada

Fases principales de la metodología de cascada:

- **Análisis de requisitos:** esta fase identifica y documenta los requisitos para que se desarrolle el software, incluidas las características, funciones y limitaciones.
- **Diseño:** cree un diseño de software detallado, incluida la arquitectura, las estructuras de datos, la interfaz de usuario y otros componentes necesarios.
- **Implementación:** En esta fase, el software se codifica de acuerdo con el diseño creado. Desarrollar y probar módulos del sistema.
- **Pruebas:** se realizan pruebas exhaustivas para verificar que el software cumple con los requisitos especificados. Identificar y corregir los errores o errores encontrados.
- **Mantenimiento:** Una vez que el software se ha implementado y probado satisfactoriamente, se distribuye y se realizan actividades de mantenimiento, que pueden incluir arreglos, actualizaciones o mejoras según lo soliciten los clientes o usuarios finales (Ionos, 2019).

Figura 16

Proceso para el desarrollo del proyecto



Nota. El modelo en cascada de cinco niveles, basado en la propuesta de Winston W. Royce, divide el proceso de desarrollo en las siguientes etapas del proyecto: análisis, diseño, implementación, verificación y mantenimiento (Ionos, 2019).

2.2. Diseño de investigación

En la base al análisis anterior se establece la estructura en base a los objetivos planteados con sus respectivas actividades para cada proceso, por lo tanto, como primer proceso se determinó los parámetros necesarios para la construcción del sistema barra esfera. A continuación, se mencionan los principales:

- Barra rígida que soporta el peso de la esfera para permitir su movimiento, los parámetros a considerar son, la longitud de la barra, material adecuado para reducir la fricción en el movimiento de la esfera.

- Esfera es el elemento que se coloca en un extremo de la barra y que se desplazada de forma controlada por el sistema de control, la esfera puede ser solida o hueca, además se puede escoger el material más adecuado como acero, plástico o madera.
- Sistema de actuación es el encargo de aplicar la fuerza o movimientos a la esfera para el control del movimiento, puede ser un motor
- Sensores son necesarios para medir y detectar la posición, velocidad y otros parámetros necesarios para el control del sistema.
- Controlador es un software de control que procesa las señales de los sensores y genera señales de control para el sistema de actuación.
- Sistema de comunicación es necesario para la comunicación entre el controlador y otros dispositivos, como son la interfaz de usuario, sistema de monitoreo mediante una conexión o cableado.

2.3. Tipo de investigación

Investigación descriptiva: Este tipo de investigación busca describir las características y propiedades del sistema barra-esfera. Se incluye la recopilación de datos sobre las dimensiones, materiales, comportamiento mecánico, restricciones y otras variables relevantes del sistema.

Investigación experimental: Se lleva a cabo experimentos para analizar el comportamiento y las interacciones del sistema barra-esfera en diferentes condiciones. Esto implica la manipulación de variables, la medición de resultados y la comparación de diferentes configuraciones para comprender mejor el sistema.

Investigación teórica o modelado: En este enfoque, se desarrollan modelos matemáticos o físicos para describir y predecir el comportamiento del sistema barra-esfera. Se implica la formulación de ecuaciones y la resolución de problemas mediante métodos analíticos o numéricos.

Investigación comparativa: Se realiza comparaciones entre diferentes diseños, materiales o configuraciones del sistema barra-esfera para evaluar su rendimiento relativo. Esto implica la recolección de datos y la realización de análisis comparativos para identificar las ventajas y desventajas de cada opción.

2.4. Técnicas e instrumentos

Técnicas de medición: Para obtener datos cuantitativos sobre el sistema barra-esfera, se utiliza técnicas de medición como la medición de longitudes, ángulos, fuerzas y desplazamientos. Los instrumentos comunes para estas mediciones pueden incluir reglas, micrómetros, calibradores.

Instrumentos de control y adquisición de datos: Para realizar experimentos y recopilar datos, se utiliza instrumentos de control y adquisición de datos, como osciloscopios, generadores de señales, amplificadores de instrumentación y sistemas de adquisición de datos.

Software de análisis de datos: Para procesar y analizar los datos recopilados durante la investigación, se utiliza herramientas de software especializadas. Esto incluye software estadístico, software de análisis numérico o software de visualización de datos.

2.5. Preguntas de investigación

¿Cuáles son las propiedades mecánicas y físicas del sistema barra-esfera y cómo influyen en su comportamiento?

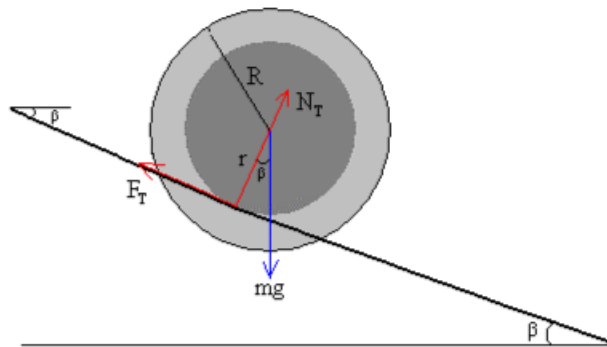
¿Cuál es la relación entre las dimensiones y materiales de la barra y la esfera?

¿Cuál es la relación entre las fuerzas aplicadas al sistema y los desplazamientos resultantes de la esfera en la barra?

- **Modelamiento matemático del sistema**

Para el modelado del sistema esfera viga se deba aplicar la segunda ley de Newton a la esfera. Además, se tiene en cuenta las fuerzas y momentos de reacción entre la barra y la esfera, en la figura se presenta el diagrama del sistema donde se observa todas las fuerzas que actúan en la esfera, como es el peso debido a la fuerza de la gravedad, la normal y la fuerza de rozamiento entre la esfera y la viga, donde se toma como referencia el centro de la viga y, con el eje x paralelo.

Figura 17
Viga inclinada y esfera rodando



Por lo tanto, se obtiene la siguiente ecuación:

$$m * g * \sin \beta - F_r = m * a ; \text{ en el eje x}$$

$$m * g * \cos \beta = n ; \text{ en el eje y}$$

Se supone que la esfera rueda sin deslizar por la viga, y se obtiene la condición de rodadura que se relaciona con el ángulo de la viga con la posición de la esfera.

$$\text{Condición de rodadura: } x = R_g * \beta$$

La fuerza de rozamiento que causa que la esfera ruede sobre la viga es decir el momento que la esfera entra en contacto por la fuerza de rozamiento se aplica la segunda ley de Newton de rotación.

$$\text{Rotación: } F_r * R_g = J_{esfera} * \frac{d^2\beta(t)}{dt^2}$$

La esfera es perfecta por lo tanto el momento de inercia será el mismo:

$$J_e = \frac{2}{5} m * R^2$$

Y la aceleración angular de la esfera y la velocidad lineal es la misma relacionando la condición de rodadura se tiene la ecuación:

$$\frac{d^2\beta(t)}{dt^2} = \frac{1}{R_g} * \frac{d^2x(t)}{dt^2}$$

Con esta ecuación se sustituye la ecuación de rotación y se obtiene la relación de la fuerza de rozamiento que es proporcional a la aceleración:

$$F_r = \frac{2}{5} * m \left(\frac{R}{R_g} \right)^2 * \frac{d^2x(t)}{dt^2}$$

Reemplazando la primera ecuación se obtiene:

$$m * g \sin \beta - \frac{2}{5} * m \left(\frac{R}{R_g} \right)^2 * \frac{d^2x(t)}{dt^2} = m * a$$

Operando se obtiene:

$$\frac{d^2x(t)}{dt^2} = \frac{g}{\left[1 + \frac{2}{5} * \left(\frac{R}{R_g} \right)^2 \right]} * \sin \beta$$

Aproximando se obtiene:

$$\frac{d^2 x(t)}{dt^2} = \frac{g}{\left[1 + \frac{2}{5} * \left(\frac{R}{R_g}\right)^2\right]} * \beta(t)$$

Linealizando se puede aplicar la transformada de Laplace:

$$\mathcal{L}\left[\frac{d^2 x(t)}{dt^2}\right] = \mathcal{L}\left[\frac{g}{\left[1 + \frac{2}{5} * \left(\frac{R}{R_g}\right)^2\right]} * \beta(t)\right]$$

$$[S^2 * X(s)] - s * x(0) - x'(0) = \frac{g}{\left[1 + \frac{2}{5} * \left(\frac{R}{R_g}\right)^2\right]} * \beta(s)$$

Para el estudio se toma las condiciones iniciales igual a cero para obtener la ecuación de la dinámica de la esfera y la viga la cual relación la salida $[X(s)]$ y la posición de la esfera con la entrada $[\beta(s)]$ para obtener la posición angular de la viga:

$$\frac{x(s)}{\beta(s)} = \frac{g}{\left[1 + \frac{2}{5} * \left(\frac{R}{R_g}\right)^2\right] * S^2}$$

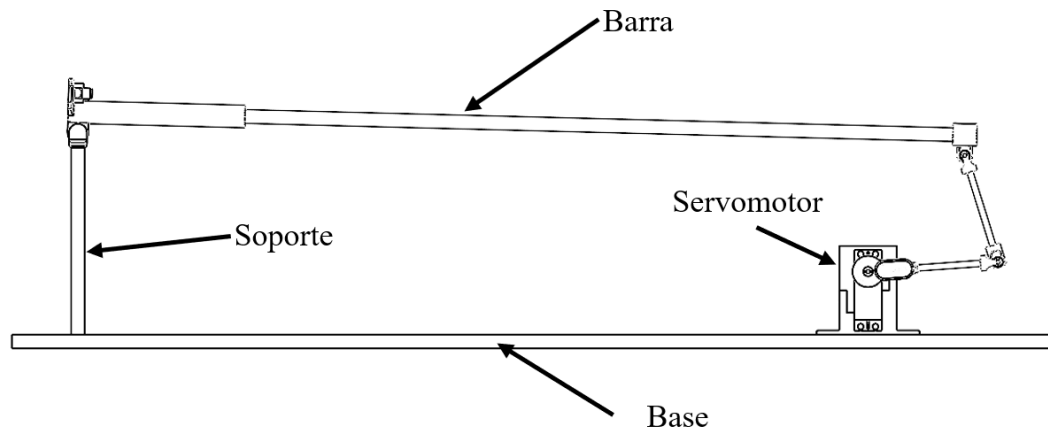
Por lo tanto, la función de transferencia depende del radio de la esfera y del radio de giro de esta. Para calcular el radio de giro es necesario conocer la distancia de los rieles por donde va a rodar la esfera.

- **Construir el mecanismo en base al modelo**

Para el diseño del sistema barra esfera se ha evaluado los existentes en el campo de la investigación por tal motivo este modelo de laboratorio será un modelo fácil de entender y que puede ser utilizado por los estudiantes para comprender los modelos de diseño en ingeniería de control en la figura 18, se muestra un esquema de los elementos del sistema.

Figura 18

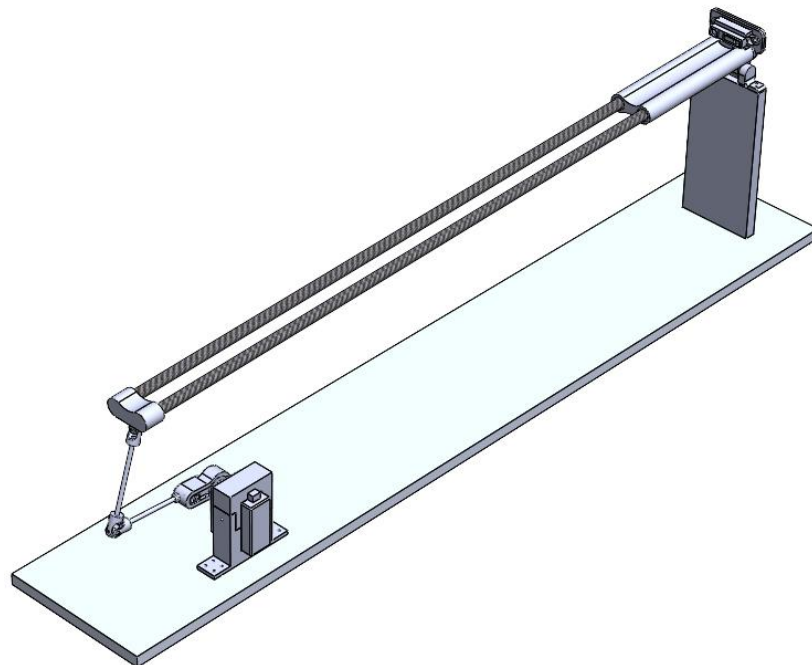
Elementos principales en el modelado del sistema barra esfera



El diseño de la barra esfera se lleva a cabo utilizando un programa de diseño asistido por computadora (CAD) altamente especializado. Este software permite crear modelos tridimensionales detallados y precisos de la barra esfera.

Figura 19

Modelado 3D del sistema barra esfera



Barra:

Para los rieles donde gira la esfera se escoge tubería en fibra de carbono con diámetro exterior, 8 mm, diámetro interior, 6 mm, longitud: 500 mm (19,68 pulgadas), espesor, 1 mm, el paquete incluye: 2 tubos de fibra de carbono lisos y brillantes. Entre las propiedades principales de este material es la alta calidad, ligero y alta resistencia: el tubo redondo de fibra de carbono ARRIS cuenta con una resistencia increíble, superando la de los tubos de aleación de aluminio y acero, mientras mantiene un peso significativamente menor. Con una impresionante resistencia a la corrosión, el agua, el calor y la fatiga, este tubo de carbono también es una alternativa confiable a los tubos de metal.

Acoples universales:

Los acoples universales permiten el movimiento de la barra al ser ensamblado por un mecanismo tipo balancín lo que se adquieren conectores de dirección de acoplamiento de eje universal para RC, el material de este componente es acero inoxidable la junta puede girar 360 grados en la figura se evidencia el modelo de la junta seleccionada para el ensamble del sistema de control barra esfera.

Rodamiento:

Para el normal movimiento del sistema es necesario instalar rodamientos por tal motivo se ha seleccionado un rodamiento de esferas 623zz 3mm x10mm x 4mm con doble blindado en la ranura profunda de acero al carbono con rendimiento estable y rentable en la figura se evidencia el modelo del rodamiento seleccionado.

Servomotor:

Se utiliza para controlar el movimiento del brazo que está controlada por el sistema de control PID para mantener el equilibrio del sistema esfera.

Varilla redonda:

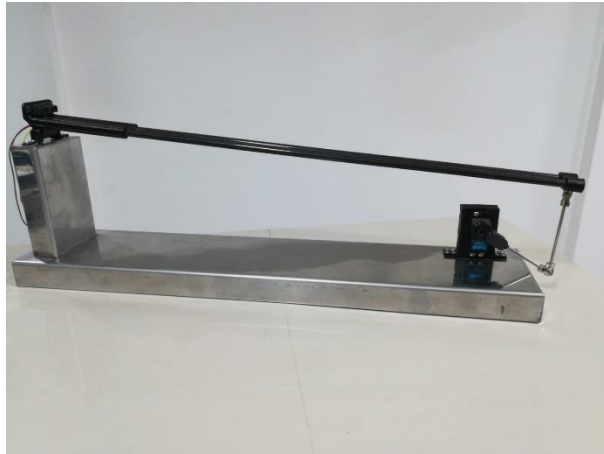
También, es necesario adquirir varillas de acero inoxidable 304 de alta dureza resistente al oxido y la corrosión, de diámetro 3 mm y 150 de longitud es altamente utilizado en el eje de motor como soporte, etc.

2.5.1. *Proceso de construcción*

Para construir el sistema barra-esfera basado en un modelado CAD, se adquieren los materiales necesarios y se preparan los componentes según las especificaciones de diseño. Luego, se ensamblan los componentes sobre una base de aluminio rígida, verificando la articulación para permitir el movimiento en un solo plano. Finalmente, se realiza una inspección y ajustes si es necesario, asegurando la estabilidad y el correcto funcionamiento del sistema en la figura 20, se muestra el ensamble completo del sistema.

Figura 20

Ensamble del sistema barra esfera



Además de los pasos mencionados anteriormente, también se coloca un servo motor en el sistema barra-esfera para ayudar en el movimiento y proporcionar estabilidad. El servo motor se conecta de manera adecuada al mecanismo, permitiendo controlar y regular el movimiento de la barra y la esfera según las señales de control enviadas al motor. Esto permite implementar

estrategias de control más precisas y mejorar el rendimiento del sistema barra-esfera en el aprendizaje de conceptos de control.

Figura 21

Instalación del servo motor



Se coloca la esfera en la parte superior de los tubos que actúan como rieles para el movimiento o desplazamiento de la esfera. Estos tubos proporcionan una guía lineal para el movimiento de la esfera a lo largo de la barra. Asegúrate de que los tubos estén colocados y fijados de manera adecuada para garantizar un desplazamiento suave y controlado de la esfera a lo largo de la barra.

Figura 22

Ubicación de la esfera en los rieles

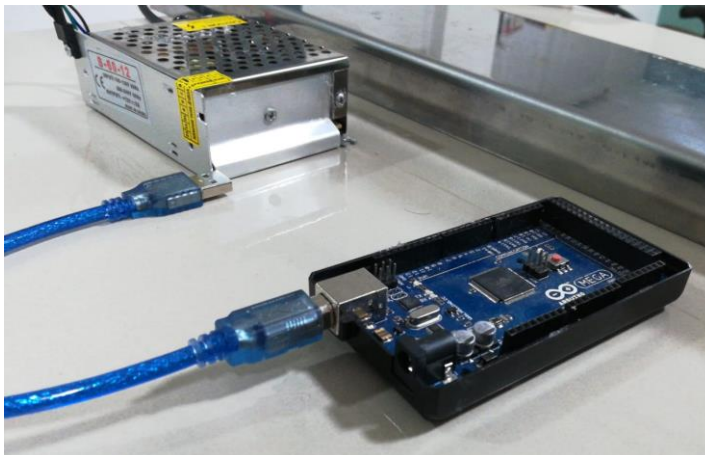


Luego se instala una fuente de alimentación de 12 VDC (12 voltios de corriente continua) en el sistema barra-esfera. Esta fuente de alimentación proporcionará la energía necesaria para alimentar los componentes eléctricos del sistema, como el servo motor u otros dispositivos electrónicos utilizados en el control. Asegúrate de conectar la fuente de alimentación correctamente, respetando las polaridades adecuadas y siguiendo las instrucciones del fabricante para garantizar un suministro de energía seguro y estable.

Después de instalar la fuente de alimentación, se coloca una placa de Arduino para el control del sistema barra-esfera. La placa de Arduino actuará como el cerebro del sistema, permitiendo la programación y ejecución de algoritmos de control para regular el movimiento de la barra y la esfera esto va conectado con el servo motor para el control del sistema. A través de la programación en Arduino.

Figura 23

Ubicación de la placa de Arduino



2.6. Proceso de programación

Para la implementación del controlador PID del sistema barra-esfera utilizando un Arduino. se lleva a cabo en este programa:

- Se incluye la biblioteca "Servo.h" para controlar el servo motor y se definen las variables y constantes necesarias.

- En la función `setup()`, se configura la comunicación serial y se inicializa el servo motor.
- En la función `loop()`, se lleva a cabo el ciclo principal del programa.
- La función `distancia_sharp()` calcula la distancia medida por un sensor Sharp y aplica un filtro de media exponencial para suavizar los valores.
- La función `PID ()` realiza el cálculo del controlador PID, donde se calcula el error, se acumula el término integral, se diferencia el error para el término derivativo y se calcula la salida del controlador PID.
- Se ajusta el ángulo del servo motor con base en la salida del controlador PID y se envía esa información al servo motor.
- La función `imprimir ()` muestra los valores relevantes a través de la comunicación serial.
- La función `serialEvent()` permite recibir parámetros de configuración a través de la comunicación serial y asignarlos a las variables correspondientes.

2.6.1. Proceso de control

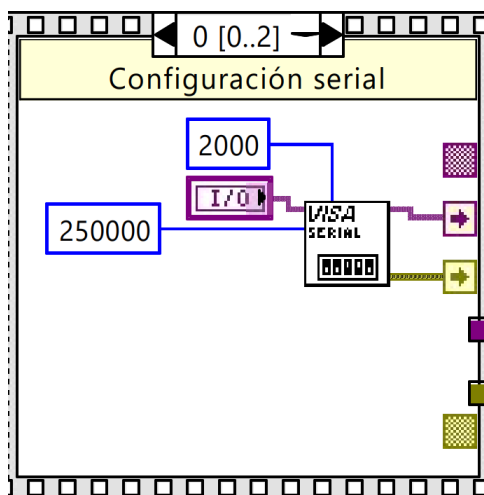
Se utiliza un sistema de visualización para controlar un sistema de esfera-barra. La interfaz principal incluye elementos que permiten controlar y monitorear el sistema mediante la visualización de datos. Esta plataforma de programación gráfica ofrece una interfaz intuitiva y poderosa para la creación de sistemas de control. Los componentes presentes en la interfaz principal ofrecen funcionalidades que permiten supervisar y regular el sistema de esfera-barra de manera eficiente.:

- Visualizar la posición de la esfera con respecto al inicio de la barra
- Modificar variables del sistema de control: P, I, D
- Ajustar las variables que definen la calidad del filtro de la señal adquirida

La primera etapa de la programación en LabVIEW se encarga de inicializar el sistema y las variables para garantizar un funcionamiento correcto como se observa en la figura 24.

Figura 24

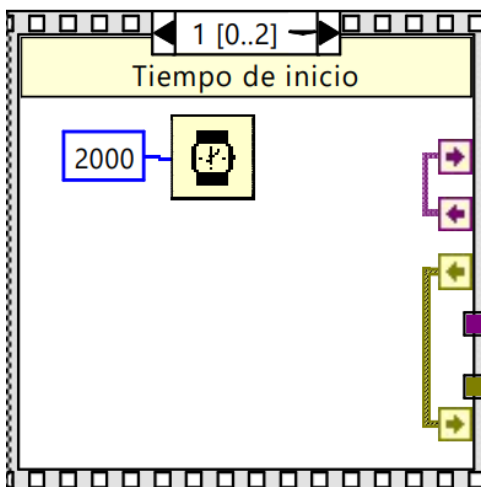
Configuración de puerto serial



En esta etapa se configura el puerto serial con una velocidad de transmisión de datos de 250,000 baudios, lo cual permite una comunicación fluida entre el sistema de software y hardware, como se muestra en la Figura 25 Además, se establece un tiempo de espera por error de comunicación de 2000 ms para garantizar un proceso eficiente y confiable.

Figura 25

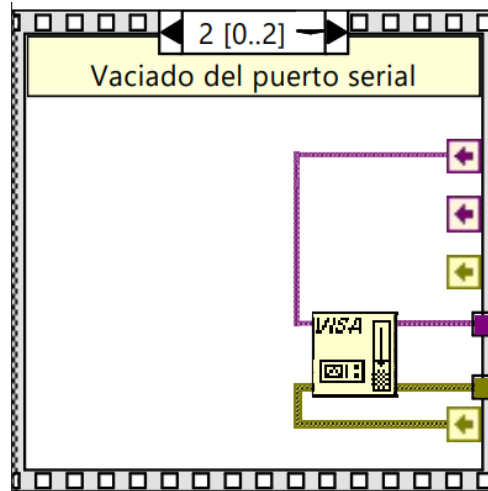
Configuración del tiempo en espera



Después de configurar la comunicación serial, se establece un tiempo de espera de 2000 ms para permitir que el sistema complete sus procesos antes de proceder con el ciclo de control, tal como se ilustra en la Figura 26.

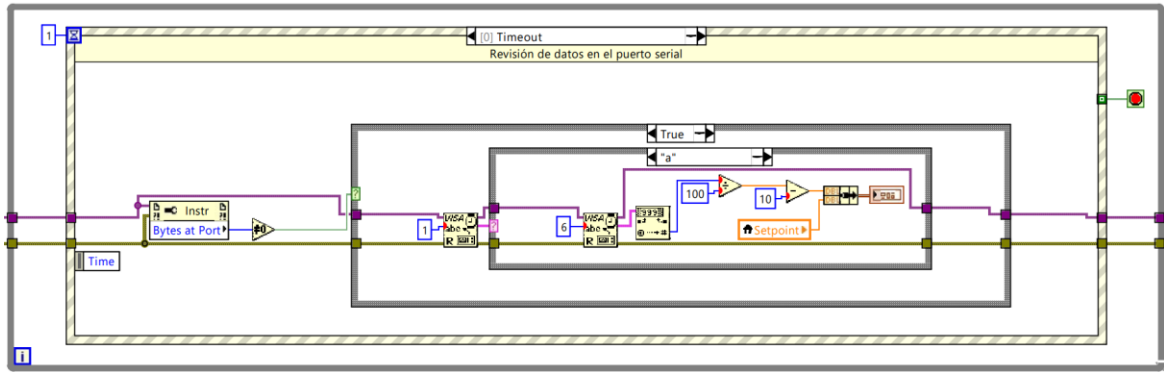
Figura 26

Vaciado del puerto serial Configuración del tiempo en espera

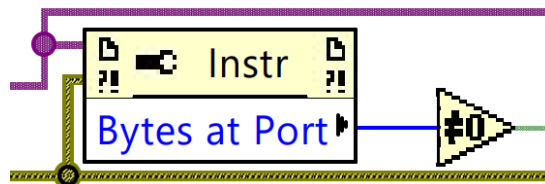


Como parte final del proceso de inicio, se lleva a cabo una operación de limpieza en el puerto serial, tal como se representa en la Figura 37. Este paso tiene como objetivo eliminar cualquier residuo de datos que pueda existir en el puerto, evitando posibles errores de comunicación al iniciar el sistema.

En la etapa siguiente del proceso, se da inicio al ciclo repetitivo que constituye la parte principal del programa. Cada proceso se ejecuta mediante un evento que se activa cuando el sistema lo requiere. Durante esta fase, se procesan las variables necesarias para asegurar su correcto funcionamiento. En este ciclo, se realizan las acciones y cálculos necesarios para controlar y monitorear el sistema de esfera-barra, utilizando las variables previamente configuradas. Esta dinámica se ilustra en la Figura 27.

Figura 27*Ciclo principal de control*

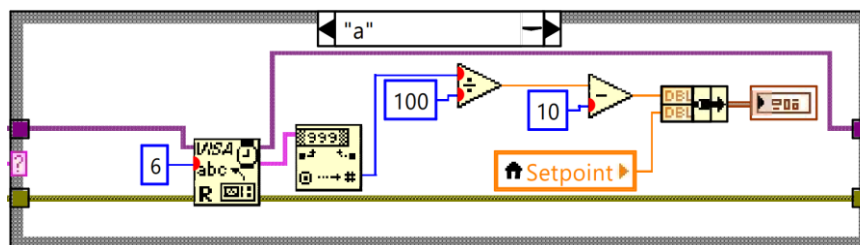
En la Figura 28. Si se detecta que no hay datos disponibles en el puerto serial, se repite el ciclo de verificación hasta que se reciban datos para su procesamiento. Esta repetición asegura que el sistema esté constantemente revisando y esperando la llegada de datos antes de continuar con el siguiente paso.

Figura 28*Conteo de datos en el puerto serial*

Una vez se detecta la presencia del primer dato en el puerto serial, se procede a realizar su lectura. Este dato puede ser 'a', indicando la recepción de la posición actual de la esfera, o 'b', indicando la recepción del ángulo actual del servomotor. Cada tipo de dato se representa en un gráfico distinto dentro de la interfaz del programa. La Figura 29 muestra la representación gráfica de la recepción de la posición de la esfera, permitiendo observar el comportamiento y los cambios de posición a lo largo del tiempo. Este gráfico proporciona una visualización clara y efectiva de la evolución de la posición de la esfera en el sistema de esfera-barra.

Figura 29

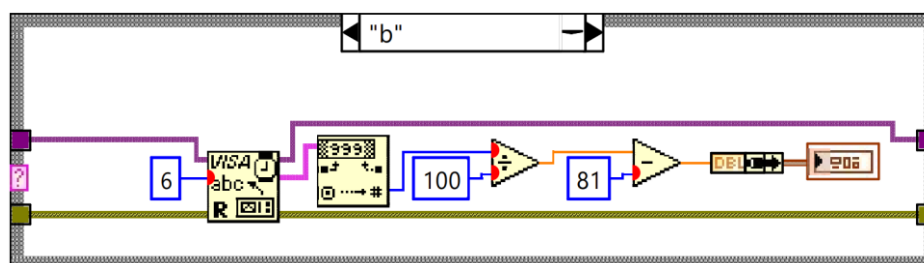
Recepción de la posición de la esfera



En contraste, la Figura 30, exhibe la recepción de la posición del servomotor, representando el ángulo actual del mismo en relación con el tiempo. Esta representación gráfica permite visualizar de forma clara y precisa la variación del ángulo del servomotor a lo largo del tiempo.

Figura 30

Conteo de la posición del servomotor



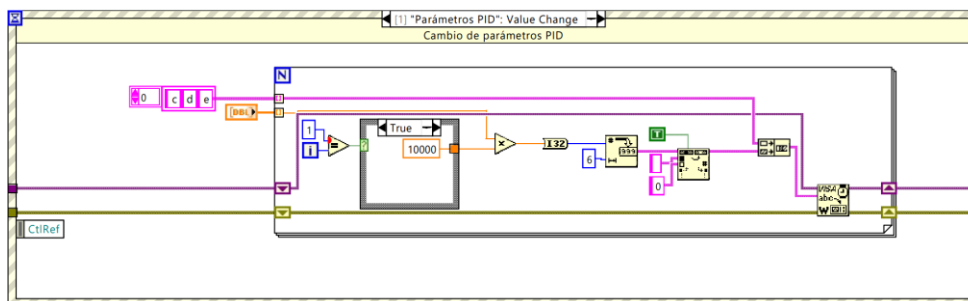
En cada ciclo de recepción, se realiza una transformación numérica específica para cada parámetro con el objetivo de lograr una escala visualmente adecuada para la percepción del usuario. Estas transformaciones son fundamentales para asegurar que los datos se presenten en una escala apropiada y comprensible, lo que facilita la interpretación y comprensión de la información por parte del usuario. La siguiente etapa del proceso permite la modificación de los diferentes parámetros del sistema. En esta fase, se brinda al usuario la capacidad de ajustar y configurar los parámetros según sea necesario, lo que permite adaptar el sistema de esfera-barra a las necesidades y requisitos específicos de control y monitoreo.

2.6.2. Envío de parámetros PID

En esta fase, se lleva a cabo la configuración de los parámetros PID, tal como se ilustra en la Figura 31. En la interfaz principal, se incluyen tres controles deslizables para ajustar cada uno de los parámetros: proporcional, integral y derivativo. Estos controles proporcionan una forma intuitiva y visualmente accesible para que el usuario modifique y ajuste los parámetros PID según sea necesario. Con esta configuración, se logra una mayor precisión y estabilidad en el control del sistema de esfera-barra.

Figura 31

Configuración de parámetros PID



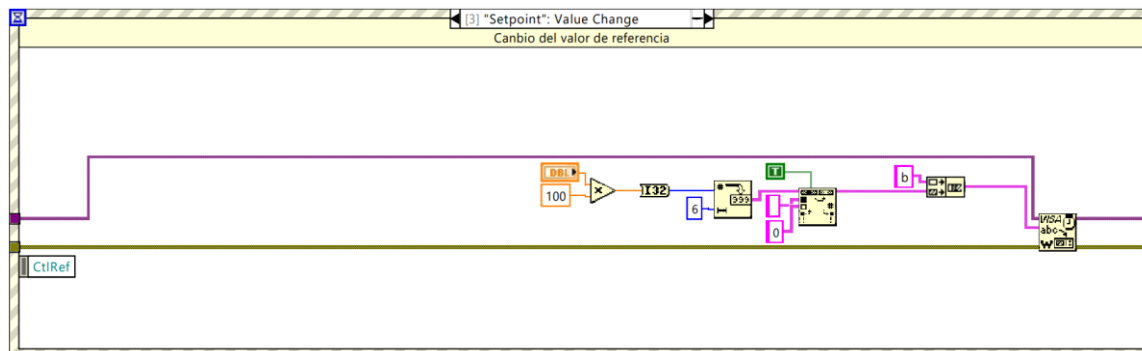
Los parámetros PID se codifican y se envían a través de un ciclo for, en el cual cada parámetro se envía junto con una letra que indica su tipo. Para el parámetro proporcional se utiliza la letra 'c', para el parámetro integral se utiliza la letra 'd' y para el parámetro derivativo se utiliza la letra 'e'. Durante este ciclo, todos los parámetros se envían simultáneamente a través del puerto serial. Cada vez que se realiza una modificación en alguno de los parámetros, se envía inmediatamente para asegurar que los cambios se apliquen en tiempo real en el sistema de control.

En la siguiente etapa del proceso, se envía el parámetro Setpoint a través del puerto serial utilizando una codificación similar a la de los parámetros PID. El envío comienza con la letra 'b' para indicar que es el valor de la consigna, seguido del valor actualizado del Setpoint en el mismo formato de envío de datos utilizado para los parámetros PID. La Figura 32 ilustra este proceso,

mostrando la codificación y el envío del valor correspondiente a través del puerto serial. Este mecanismo permite ajustar dinámicamente el Setpoint y enviarlo al sistema de control para su actualización en tiempo real.

Figura 32

Configuración del setpoint



En la etapa final del proceso, se habilita el envío del parámetro Alpha, el cual permite ajustar el nivel de filtrado utilizado por el filtro de media exponencial implementado en Arduino.

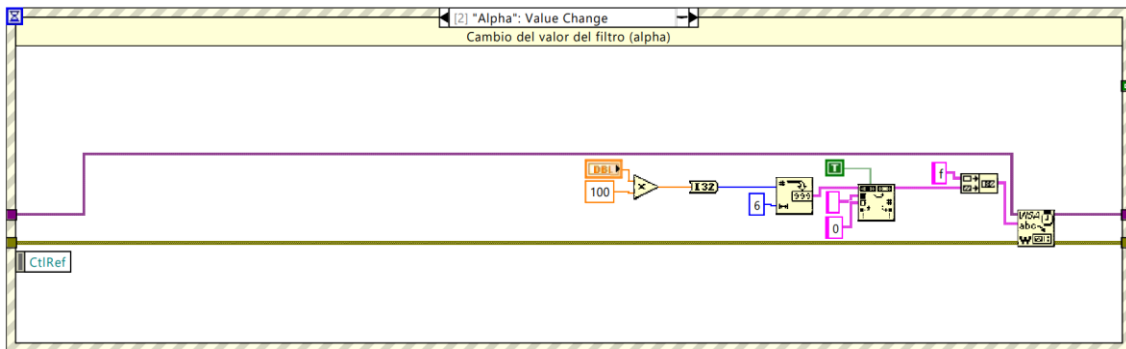
El parámetro Alpha determina la ponderación relativa de los valores previos y actuales en el cálculo del filtro de media exponencial. Un valor más alto de Alpha otorga mayor importancia a los datos actuales, lo que resulta en una respuesta más rápida del filtro ante cambios en la señal. Por otro lado, un valor más bajo de Alpha otorga mayor importancia a los datos previos, suavizando así la respuesta del filtro y reduciendo el efecto de variaciones rápidas en la señal.

El envío del parámetro Alpha se realiza a través del puerto serial utilizando una codificación similar a la empleada para los parámetros anteriores. Se inicia con la letra 'f', seguida del valor actualizado de Alpha, tal como se muestra en la Figura 33.

Cabe mencionar que el código puede detenerse mediante el botón STOP ubicado en la interfaz principal, lo que permite interrumpir la ejecución del programa de manera controlada.

Figura 33

Configuración de parámetros Alpha



2.6.3. Configuración Arduino

El módulo Arduino desempeña un papel central y actúa como el núcleo principal del sistema de control esfera-barra. Su principal función consiste en adquirir las señales de los sensores, procesar la función PID y controlar los dispositivos responsables del movimiento del sistema. Esta integración de funciones permite la creación de un sistema de control retroalimentado en el que se toman decisiones en tiempo real para mantener la estabilidad y un control preciso del sistema.

Al comenzar el código, es imprescindible configurar todas las variables necesarias para asegurar el correcto funcionamiento del sistema. Estas variables incluyen los parámetros PID, el Setpoint y Alpha, los cuales serán modificados posteriormente a través de la interfaz del programa de control. A continuación, se mostrará las funciones que permiten el funcionamiento correcto del sistema de control.

2.6.4. Función PID:

Esta función desempeña un papel fundamental al calcular la salida del controlador Proporcional Integral Derivativo (PID) para ajustar la posición del servomotor. El proceso de cálculo se lleva a cabo con los siguientes procedimientos:

- a) $\text{error} = \text{dist_sharp} - \text{variables} [1]$

Calcula el error como la diferencia entre la lectura actual del sensor de distancia (`dist_sharp`) y el valor objetivo o setpoint almacenado en variables [1].

b) `integral += error`

Acumula el error actual (`error`) en la variable `integral`. Esto permite tener en cuenta los errores pasados y aplicar una corrección a largo plazo.

c) `integral = constrain(integral, -20, 20)`

Limita el valor acumulado de `integral` dentro del rango de -20 a 20. Esto evita que el término `integral` se vuelva demasiado grande y afecte negativamente el comportamiento del controlador.

d) `derivativo = error - last_error`

Calcula la derivada del error actual restando el error actual (`error`) del error anterior (`last_error`). La derivada se utiliza para medir la tasa de cambio del error y ayudar a anticipar las futuras tendencias, ayudando a estabilizar el control y mejorar las respuestas del sistema.

e) `output = 81 + variables [2] * error + variables [3] * integral / 100 + variables [4] * derivativo`

Calcula la salida del controlador PID utilizando los valores de error, `integral` y `derivativo`, así como los parámetros de configuración almacenados en variables. La fórmula utilizada es una combinación de los términos proporcionales, integrales y derivativos multiplicados por sus respectivos coeficientes (`variables [2]`, `variables [3]` y `variables [4]`). Además, se suma 81 para ajustar el punto de referencia del servo motor.

f) `last_error = error`

Guarda el valor actual de error en `last_error` para su uso en la siguiente iteración. Esto asegura que siempre se tenga acceso al valor del error anterior para calcular la derivada en la próxima iteración.

2.6.5. *Función distancia_sharp*

Esta función se encarga de leer y procesar la señal proveniente de un sensor de distancia Sharp conectado al pin analógico A0 aplicando un suavizado exponencial para reducir el ruido en la lectura, realiza un cálculo para linealizar la lectura y aplica un ajuste adicional para obtener la distancia real medida por el sensor. El resultado final se almacena en la variable `dist_sharp`, que se utiliza posteriormente en el cálculo del controlador PID. A continuación, se describe el proceso paso a paso:

```
dist_sharp = analogRead(A0)
```

Lee el valor analógico presente en el pin A0 utilizando la función `analogRead()`. Este valor representa la lectura actual del sensor de distancia Sharp.

```
alpha = variables [5]
```

Asigna el valor almacenado en `variables [5]` a la variable `alpha`. El parámetro `alpha` se utiliza en el cálculo de suavizado exponencial.

```
dist_sharp_sm = alpha * dist_sharp + (1 - alpha) * dist_sharp_sm
```

Realiza un suavizado exponencial utilizando la técnica de media ponderada exponencial. Se multiplica la lectura actual del sensor (`dist_sharp`) por el valor de `alpha` y se suma al producto del valor anterior de `dist_sharp_sm` por $(1 - \alpha)$. El resultado se asigna a `dist_sharp_sm`, que representa la lectura suavizada del sensor.

```
float d_sharp = 5061.3 / dist_sharp_sm - 10.144
```

Realiza un cálculo para linealizar la lectura del sensor de distancia. La fórmula utilizada es específica para el sensor Sharp en uso. El resultado se almacena en la variable `d_sharp`.

```
dist_sharp = d_sharp - 2.5
```


Aplica un ajuste adicional a la lectura linealizada para obtener la distancia real medida por el sensor de distancia. El valor 2.5 representa un desplazamiento de calibración específico para el sensor utilizado.

2.6.6. Filtro de media exponencial

El filtro de media exponencial (EMA, por sus siglas en inglés) es una técnica utilizada para suavizar una serie de datos, dándole más peso a las observaciones más recientes. A diferencia del filtro de media aritmética, donde se asigna el mismo peso a todos los puntos de datos, el EMA permite una adaptación más rápida a los cambios en la tendencia de los datos siendo una herramienta útil para dispositivos de procesamiento limitados.

El EMA se calcula utilizando la siguiente fórmula recursiva:

$$\text{EMA}(t) = \alpha * X(t) + (1 - \alpha) * \text{EMA}(t-1)$$

Donde:

1. $\text{EMA}(t)$: Valor del EMA en el tiempo t .
2. $X(t)$: Valor actual de la serie de datos en el tiempo t .
3. $\text{EMA}(t-1)$: Valor del EMA en el tiempo $t-1$, muestra anterior
4. α : Factor de suavizado exponencial, varía en el rango de 0 a 1.

El factor de suavizado, α , determina la importancia relativa de los datos antiguos y nuevos en el cálculo del EMA. Un valor más cercano a 1 da mayor peso a los datos más recientes, lo que resulta en una adaptación más rápida a los cambios. Un valor más cercano a 0 da mayor peso a los datos antiguos, lo que produce un suavizado más gradual.

Al utilizar el EMA como filtro, cada nuevo punto de datos reemplaza parcialmente al valor anterior del EMA, lo que permite que la serie de datos suavizada siga los cambios en la tendencia de los datos de manera más ágil.

Función imprimir

Esta función envía datos a LabVIEW a través de la comunicación serial para su visualización en la interfaz. Da formato a la distancia y el ángulo en cadenas de caracteres y las envía, permitiendo la supervisión en tiempo real de estos valores.

1. `char buff[10]`

Declara un arreglo de caracteres llamado buff con capacidad para 10 elementos. Este arreglo se utiliza para almacenar los caracteres formateados antes de ser enviados a través de la comunicación serial.

2. `sprintf(buff, "a%06lu", (unsigned long)((dist_sharp + 10) * 100))`

Da formato a la distancia medida por el sensor de distancia (dist_sharp) en una cadena de caracteres y la almacena en buff. La cadena de formato utilizada es "a%06lu", lo que significa que se espera un número entero sin signo de 6 dígitos de longitud y se agrega una 'a' al comienzo de la cadena formateada.

3. `Serial.print(buff)`

Envía la cadena formateada almacenada en buff a través de la comunicación serial. Esto imprime la información de la distancia medida por el sensor de distancia.

4. `sprintf(buff, "b%06lu", (unsigned long)((angle)*100))`

Formatea el ángulo actual del servo motor (angle) en una cadena de caracteres y la almacena en buff. La cadena de formato utilizada es "b%06lu", lo que significa que se espera un número entero sin signo de 6 dígitos de longitud y se agrega una 'b' al comienzo de la cadena formateada.

5. `Serial.print(buff)`

Envía la cadena formateada almacenada en buff a través de la comunicación serial. Esto imprime la información del ángulo actual del servo motor.

Función serialEvent

Esta función es un evento especial en Arduino que se ejecuta automáticamente cuando hay datos disponibles en el puerto serie. Su principal objetivo es leer los caracteres recibidos y almacenarlos en un arreglo denominado `input`. Posteriormente, realiza la conversión de la cadena de caracteres en un valor numérico y lo asigna al parámetro correspondiente en el arreglo de variables. Gracias a este mecanismo, se habilita la posibilidad de configurar dinámicamente los parámetros del programa a través de la comunicación serial, brindando flexibilidad y permitiendo ajustar los valores de manera interactiva durante la ejecución del programa. A continuación, se proporciona una explicación detallada de esta función:

1. `float value`

Declara una variable `value` de tipo `float` que se utilizará para almacenar el valor numérico recibido a través de la comunicación serial.

2. `if (Serial.available())`

Verifica si hay datos disponibles en el puerto serie para su lectura.

3. `input[i] = (char)Serial.read()`

Lee el siguiente byte de datos disponible en el puerto serie y lo guarda en la posición `i` del arreglo `input`. El byte leído se trata como un carácter y se almacena en `input` como un carácter ASCII.

4. `int pos = input [0] - 'a'`

Calcula la posición en el arreglo variables donde se almacenará el valor recibido. Esto se hace restando el carácter 'a' del primer elemento de `input`, lo cual convierte el carácter en su correspondiente valor numérico.

5. `if (pos >= 0 && pos < 10) i++; else i = 0`

Verifica si la posición calculada está dentro del rango válido (0 a 9). Si es así, incrementa *i* en uno para avanzar a la siguiente posición en *input*. De lo contrario, reinicia *i* a cero para comenzar a recibir una nueva cadena de caracteres.

```
6.    if (i == 7)
```

Comprueba si se han leído los siete caracteres esperados para formar una cadena completa.

```
7.    int val = 0
```

Declara una variable *val* de tipo entero para almacenar el valor numérico convertido de la cadena de caracteres recibida.

```
8.    for (int j = 1; j < 7; j++) val = val * 10 + input[j] - '0'
```

Convierte la cadena de caracteres recibida en un número entero. El bucle *for* itera a través de los caracteres de la cadena de caracteres, multiplica *val* por 10 en cada iteración y le suma el valor numérico del carácter actual (*input[j] - '0'*) para construir el número entero completo.

```
9.    value = (float)val / 100
```

Divide el valor entero *val* por 100 para obtener el valor en formato *float* y lo asigna a *value*.

```
10.   variables[pos] = value
```

Almacena el valor convertido en la posición *pos* del arreglo *variables*, que corresponde al parámetro que se está configurando.

```
11.   i = 0
```

Reinicia *i* a cero para comenzar a recibir una nueva cadena de caracteres.

CAPÍTULO III: RESULTADOS Y DISCUSIÓN

Este capítulo se centra en presentar y analizar los resultados obtenidos de las simulaciones y experimentos realizados con el sistema Barra-esfera. A través de datos concretos, gráficos y tablas, se busca elucidar el comportamiento del sistema bajo distintas condiciones y entradas. Además, este capítulo no solo se limita a la exposición de los resultados, sino que también profundiza en su interpretación, estableciendo conexiones con las teorías y metodologías presentadas previamente en la tesis. La discusión detallada permite identificar puntos clave de éxito, áreas de mejora y proporciona una comprensión integral del comportamiento del sistema en el contexto de sistemas de control.

3.1 Metodología de Validación

En este punto se detallará el enfoque estructurado adoptado para validar el comportamiento y las características del sistema Barra-esfera en el contexto de sistemas de control. Desde las herramientas y software seleccionados hasta los criterios específicos de evaluación, este proceso meticuloso garantiza que los resultados y conclusiones derivados sean no solo precisos, sino también relevantes y aplicables a situaciones reales.

3.1.1. *Herramientas y software utilizados*

Para garantizar una validación efectiva del sistema Barra-esfera, es esencial elegir herramientas y software que no solo sean robustos, sino que también ofrezcan una interfaz eficaz entre el mundo digital y el físico.

MATLAB con Simulink: MATLAB, un entorno de programación de alto rendimiento para cálculos matemáticos, es complementado con Simulink, una plataforma de modelado y simulación que permite diseñar sistemas dinámicos de forma visual. En este estudio, Simulink se

utilizó para modelar y simular el comportamiento del sistema Barra-esfera bajo diferentes condiciones y entradas.

Toolbox "Simulink Arduino": Esta herramienta sirve como un puente entre Simulink y el hardware Arduino, permitiendo que el modelo de Simulink se comuniquen en tiempo real con el Arduino. A través de este toolbox, fue posible implementar el controlador diseñado en Simulink directamente en el Arduino, llevando el sistema Barra-esfera desde la simulación teórica a la aplicación práctica y tangible.

Con la combinación de estas herramientas, se estableció una metodología cohesiva que permitió validar de manera efectiva el sistema, contrastando resultados teóricos con observaciones prácticas.

3.1.1. Procedimiento de simulación

El proceso de simulación es esencial para obtener una comprensión inicial del comportamiento del sistema Barra-esfera antes de su implementación real. A continuación, se detalla el procedimiento adoptado:

Modelado en Simulink:

Inicialmente, el sistema Barra-esfera fue modelado en Simulink utilizando bloques específicos que representan cada componente del sistema, como actuadores, sensores y el propio controlador.

Se configuraron las propiedades y parámetros de cada bloque según las especificaciones y características técnicas conocidas del sistema real.

Configuración de la Simulación:

Se definieron las condiciones iniciales, como el estado inicial de la esfera, parámetros de control, entre otros.

Se estableció un tiempo de simulación adecuado para observar el comportamiento del sistema durante un ciclo completo o hasta que alcanzara un estado estacionario.

Selección de Entradas:

Se aplicaron diferentes entradas al sistema para evaluar su respuesta bajo distintas condiciones. Estas podrían incluir funciones escalón, impulso, sinusoidales o cualquier otra entrada relevante para la investigación.

Simulación y Registro de Datos:

Con todo configurado, se inició la simulación en Simulink. se monitorizaron y registraron variables clave, como la posición de la esfera, errores, señales de control, entre otros, utilizando las herramientas de registro de datos de Simulink.

Implementación en Arduino:

Una vez satisfechos con la simulación en Simulink, se utilizó el toolbox "Simulink Arduino" para cargar el controlador en la tarjeta Arduino.

El modelo de Simulink se conectó al hardware físico (Barra-esfera y Arduino) a través del toolbox, permitiendo la comunicación en tiempo real y el control directo del sistema.

Pruebas en Tiempo Real:

Con el controlador cargado en el Arduino, se realizó una serie de pruebas en tiempo real, observando el comportamiento del sistema Barra-esfera y comparando con los resultados de la simulación.

Iteraciones:

Basándose en las discrepancias o áreas de mejora observadas entre la simulación y las pruebas en tiempo real, se realizaron ajustes en el modelo de Simulink o en el controlador y se repitió el proceso hasta obtener resultados consistentes y satisfactorios.

3.1.2. *Parámetros y condiciones iniciales utilizadas.*

La precisión y relevancia de una simulación dependen en gran medida de los parámetros y condiciones iniciales seleccionados. Estas elecciones no sólo reflejan el estado inicial del sistema, sino que también determinan cómo se comportará durante la simulación. A continuación, se presentan los parámetros y condiciones iniciales empleados en nuestro estudio sobre el sistema Barra-esfera.

Parámetros del Sistema:

Coefficientes de Fricción: Estos valores reflejan cómo la resistencia afecta el movimiento de la esfera sobre la barra.

Constantes del Controlador: Ganancias, tiempos de respuesta, y otros parámetros relacionados al controlador.

Propiedades de la Barra y la Esfera: Esto incluye masa, longitud, material y otras propiedades físicas relevantes.

Actuadores y Sensores: Especificaciones técnicas y características que influyen en la dinámica del sistema, como la velocidad de respuesta o el rango operativo.

Condiciones Iniciales:

Posición de la Esfera: La ubicación inicial de la esfera en la barra al comienzo de la simulación.

Velocidad de la Esfera: La velocidad y dirección inicial (por ejemplo, si está en reposo o tiene una velocidad inicial dada).

Estado del Controlador: Si el controlador comienza en un estado particular, como en modo de espera o activo.

Condiciones Externas:

Entorno Operativo: Temperatura, humedad u otros factores ambientales que podrían afectar el comportamiento del sistema.

Perturbaciones: Si se simulan perturbaciones externas, como golpes o cambios bruscos en la inclinación de la barra.

3.1.3. Descripción de las entradas utilizadas para probar el sistema.***Entrada en Escalón:***

Descripción: Una entrada que cambia bruscamente de un valor constante a otro. Es útil para evaluar la respuesta transitoria del sistema.

Aplicación en el sistema Barra-esfera: Se representará un cambio repentino en la posición deseada de la esfera o un cambio súbito en la inclinación de la barra.

Entrada Sinusoidal:

Descripción: Una entrada que varía sinusoidalmente con el tiempo. Ayuda a evaluar cómo el sistema responde a entradas oscilatorias.

Aplicación en el sistema Barra-esfera: Simula condiciones donde la esfera o la barra están sujetas a oscilaciones, como vibraciones o movimientos cíclicos.

Entrada de Impulso:

Descripción: Una entrada breve y de alta magnitud. Es útil para entender la respuesta dinámica del sistema a perturbaciones rápidas.

Aplicación en el sistema Barra-esfera: Podría representar una perturbación súbita, como un golpe a la esfera o a la barra.

Entrada Rampa:

Descripción: Una entrada que aumenta linealmente con el tiempo. Sirve para evaluar cómo el sistema maneja cambios constantes en el tiempo.

Aplicación en el sistema Barra-esfera: Podría simular una situación donde se desea mover la esfera a lo largo de la barra a una velocidad constante.

Entradas Combinadas:

Descripción: Una mezcla de las entradas anteriores para simular escenarios más complejos.

Aplicación en el sistema Barra-esfera: Un ejemplo podría ser una entrada en escalón seguida de una entrada sinusoidal, simulando un cambio de posición y luego una vibración.

Entradas de Perturbación:

Descripción: Entradas diseñadas específicamente para perturbar el sistema y evaluar su robustez o capacidad de recuperación.

Aplicación en el sistema Barra-esfera: Podría ser un cambio brusco en la inclinación de la barra mientras se intenta mantener la esfera en una posición particular.

3.2. Resultados de Simulación

A continuación, se destaca los diferentes puntos validos que se tomaron en cuenta para los resultados.

3.2.1. Datos Obtenidos

Los datos obtenidos se seccionan de acuerdo al modelamiento matemático y a la simulación.

Modelo matemático

En la sección de modelamiento, se aborda la tarea fundamental de representar y comprender las dinámicas intrínsecas del sistema Barra-Esfera. Esta representación se bifurca en dos enfoques complementarios. Primero, se presenta un modelo experimental derivado directamente de la recolección de datos, donde las entradas y salidas observadas del sistema se utilizan en MATLAB para identificar una función de transferencia que capture la relación entrada-salida con el menor error posible. Esta metodología se apoya en herramientas avanzadas como el identificador de

funciones para una precisión optimizada. En paralelo, también se examina un modelo teórico basado en principios físicos fundamentales.

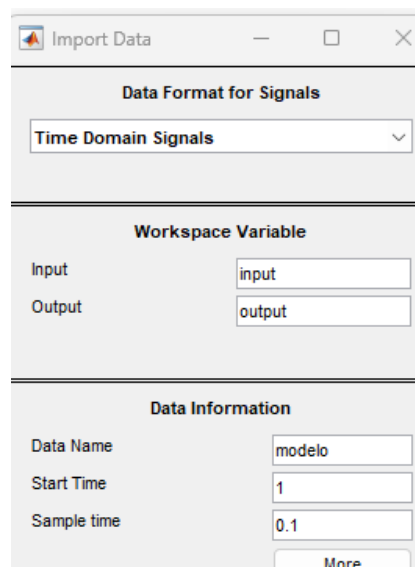
Modelo Experimental

Dentro del ámbito de la ingeniería de control, la capacidad de extraer modelos precisos directamente de datos experimentales es esencial, especialmente cuando los sistemas presentan complejidades o no linealidades que dificultan un enfoque puramente teórico. En esta sección, se aborda en el modelo experimental del sistema Barra-Esfera. A través de la recolección meticulosa de datos de entrada y salida del sistema en funcionamiento real, utilizando herramientas avanzadas en MATLAB, como el identificador de funciones, para deducir una función de transferencia que representa fielmente el comportamiento observado.

Esta aproximación empírica permite capturar dinámicas no anticipadas y ofrece una vista pragmática del sistema que complementa las perspectivas teóricas, Figura 35.

Figura 35

Configuración de la ventana de importación de modelos.

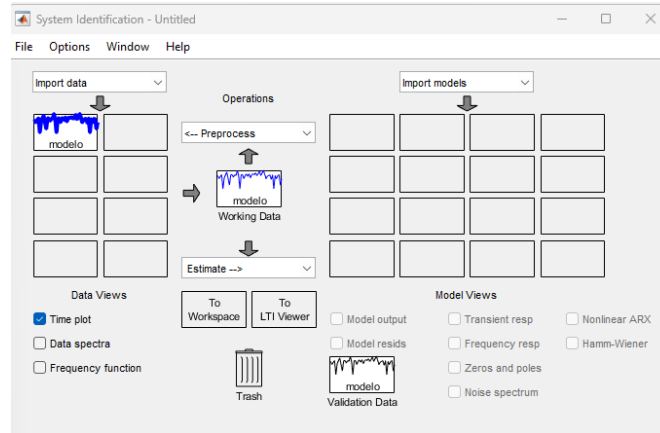


Una herramienta particularmente valiosa en nuestra exploración es el proceso de identificación del sistema. Esta técnica consiste en derivar modelos matemáticos a partir de datos

medidos, brindando la capacidad de capturar características dinámicas intrincadas que podrían ser difíciles de modelar de manera teórica, Figura 36.

Figura 36

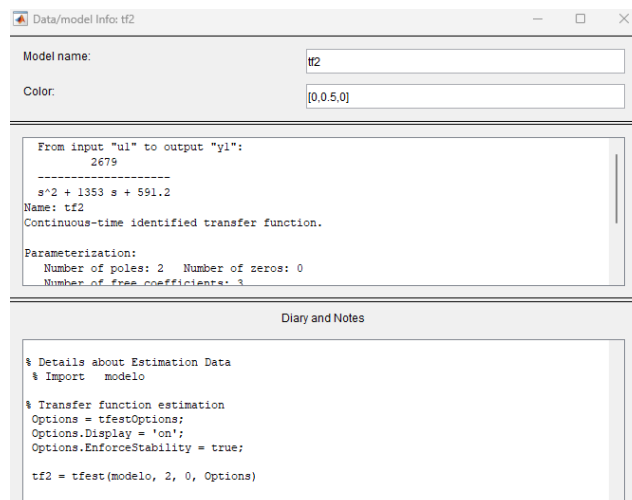
Configuración de la importación del modelo.



Utilizando la toolbox de identificación del sistema en MATLAB, se analiza y ajusta diferentes estructuras de modelos, comparando su precisión con respecto a los datos reales adquiridos. Esta adaptabilidad no solo permite una representación más realista del sistema Barra-Esfera, sino que también sienta las bases para un diseño de control más informado y robusto, Figura 37, al tener un modelo que se alinea estrechamente con el comportamiento real observado.

Figura 37

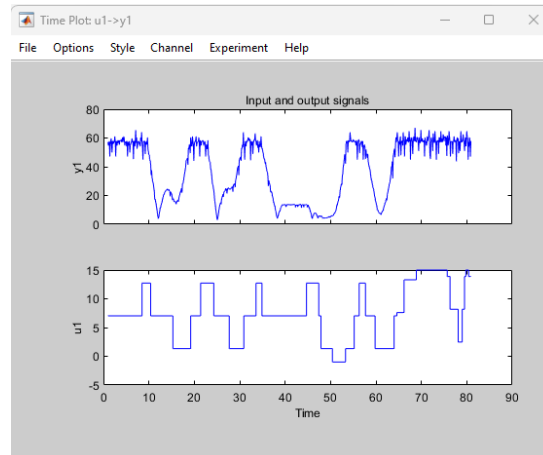
Parámetros de la función de transferencia del modelo importado.



De este procedimiento se obtuvo la siguiente función de transferencia de la Figura 38 con una respuesta dinámica del sistema como se ve en la, en la que se trabaja como modelo de experimentación.

Figura 38

Gráficas de la función de transferencia del modelo.



Modelo Teórico (Modelando la dinámica de la esfera en la barra inclinada):

La fuerza debido a la gravedad que actúa sobre la esfera es: $m * g * \sin \theta$, donde θ es el ángulo de inclinación de la barra. Si se considera que $\sin \theta \approx \theta$ para ángulos pequeños, la ecuación de movimiento para la esfera en la barra es:

$$m \frac{d^2x}{dt^2} = mg\theta - f \frac{dx}{dt}$$

Donde:

- m es la masa de la esfera.
- x(t) es la posición de la esfera en la barra.
- g es la aceleración debido a la gravedad (normalmente 9.81m/s²).
- f es el coeficiente de fricción entre la esfera y la barra.
- $\Theta(t)$ es el ángulo de inclinación de la barra.

Transformando esta ecuación diferencial al dominio de Laplace (y asumiendo condiciones iniciales cero):

$$ms^2X(s) = mg\theta(s) - fsX(s)$$

Donde:

- $X(s)$ es la transformada de Laplace de la posición
- $x(t)$ de la esfera.
- $\theta(s)$ es la transformada de Laplace del ángulo $\theta(t)$ de la barra.

Re arreglando y factorizando para $X(s)/\theta(s)$:

$$\frac{X(s)}{\theta(s)} = \frac{mg}{ms^2 + fs}$$

Esta función de transferencia describe la relación entre la posición de la esfera $X(s)$ y el ángulo de inclinación de la barra $\theta(s)$. Es una representación simplificada, basada en aproximaciones lineales y la omisión de otras posibles dinámicas externas.

El coeficiente de fricción entre una esfera de caucho y una superficie plástica lisa depende de varias condiciones, incluida la textura exacta del plástico, la presión aplicada por la esfera sobre la superficie (relacionada con su peso), y las condiciones ambientales (temperatura, humedad, etc.).

Dicho esto, para las esferas de caucho sobre superficies plásticas, el coeficiente de fricción estático (cuando la esfera no está en movimiento) suele estar en un rango de 0.5 a 1.0, mientras que el coeficiente de fricción cinético (mientras la esfera se está moviendo) suele ser ligeramente menor, en este caso de esta experimentación el coeficiente de fricción se lo tomo de 0,5.

Simulación

En la sección de simulación, se procede a exponer el ejercicio práctico que da vida a nuestra investigación tanto en la práctica de simulación real y la práctica de simulación de modelo.

Practica de simulación real

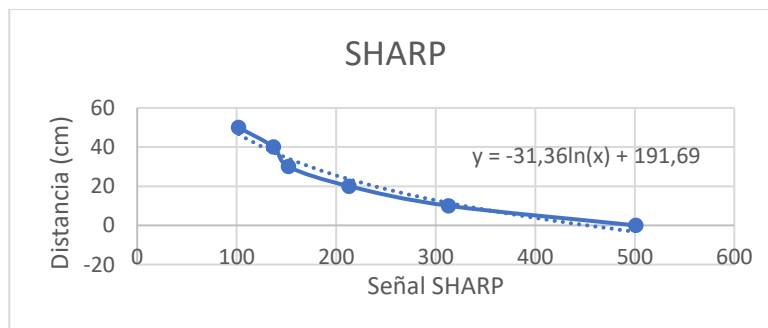
En esta práctica real se utilizó el toolbox arduino, en el cual se configuro la entrada y la salida de acuerdo con la linealización del sensor y a la relación del ángulo del servomotor con respecto al ángulo de la barra.

Una fase crítica de esta etapa fue la precisa alineación del sensor Sharp, asegurando que cada lectura fuese fidedigna y representativa de las condiciones reales del sistema, ya que la característica dinámica del sensor no es lineal, lo cual se procedió a obtener datos experimentales del sensor con cada 10 cm de distancia como se ve en anexo 1.

Del cual se obtuvo el siguiente resultado expuesto en la Figura 39, en el cual también se expresa su ecuación, la cual se representa en Simulink como la entrada.

Figura 39

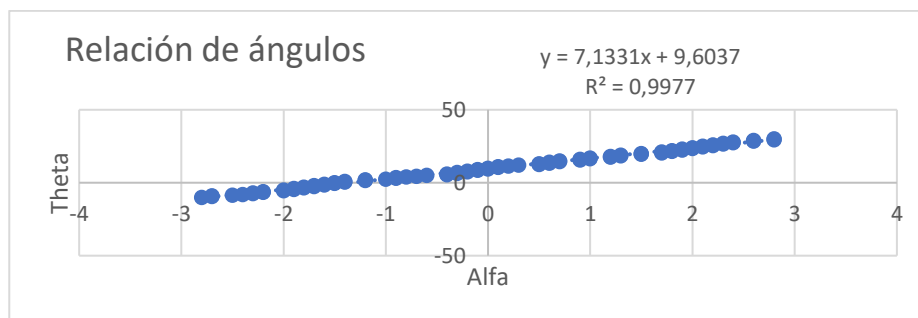
Gráfica y formula de la entrada (sensor).



A la par, se llevó a cabo el restablecimiento de las relaciones angulares, con un enfoque particular en la correlación entre el ángulo θ regulado por el servomotor y el ángulo α de la barra. Esta calibración meticulosa fue esencial para garantizar que las simulaciones fueran lo más cercanas a la realidad, de esta manera se tomaron los muestras desde -10 grados a 30 grados por experimentación como se observa en el anexo 1 obteniendo de resultado la siguiente Figura 40 acompañada con la ecuación que debía ser interpretada en Simulink como una salida.

Figura 40

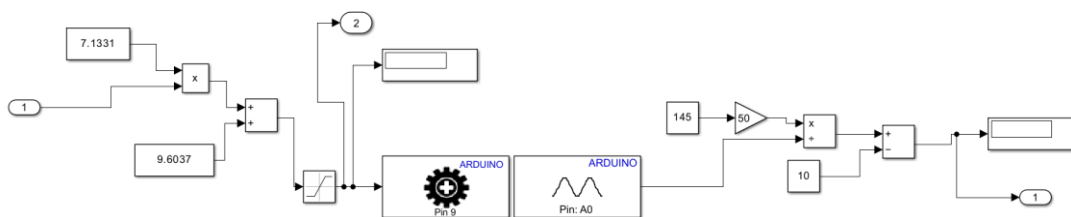
Gráfica y fórmula de relación de ángulos.



Teniendo en cuenta las ecuaciones descritas en Simulink se procede a su creación como se indica en la Figura 41.

Figura 41

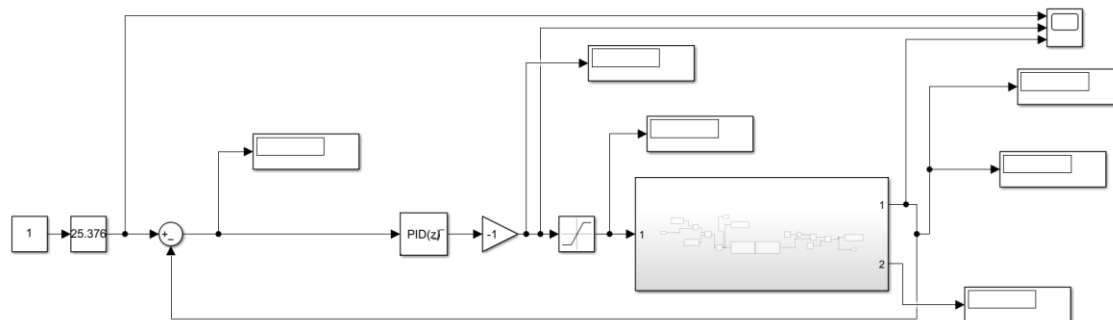
Estructuración de la planta.



Ya establecido las entradas y salidas de la planta se proporciona el controlador y la configuración lazo cerrado con el setpoint ubicado para la entrada y el PID, Figura 42.

Figura 42

Estructuración del sistema lazo cerrado.

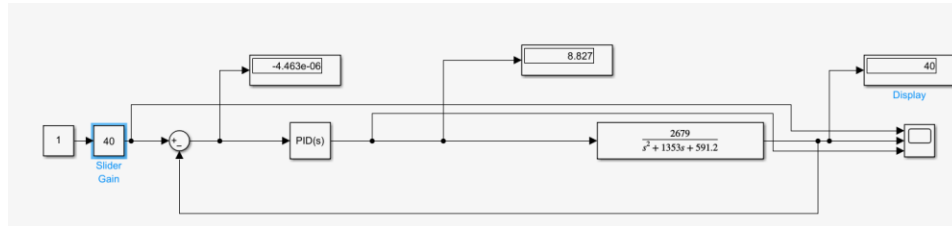


Practica de simulación de modelo

En el proceso de validación de un sistema de control, es esencial contraponer las predicciones teóricas con los resultados obtenidos en escenarios prácticos. Por un lado, el modelo experimental, forjado a partir de datos recopilados (Anexo 2) por experimentación, nos ofrece una ventana directa a la realidad y comportamiento real del sistema, obteniendo la función de transferencia expuesta en la Figura 43.

Figura 43

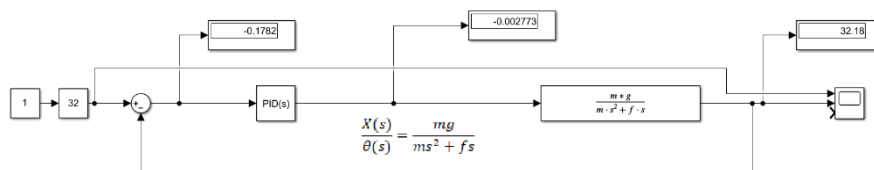
Simulación del sistema de control mediante pruebas.



Por otro lado, el modelo teórico, construido meticulosamente a través de ecuaciones y principios fundamentales, nos brinda el marco conceptual y matemático que subyace a nuestro diseño. Al simular ambos modelos, se busca identificar las coincidencias y divergencias entre la teoría y la práctica, permitiéndonos refinar nuestro enfoque y fortalecer la robustez de nuestro sistema, Figura 44.

Figura 44

Simulación del sistema de control mediante ecuaciones.



Adicionalmente, se realizó una simulación exhaustiva comparando el modelo experimental, basado en datos empíricos, con el modelo teórico previamente establecido. Estas comparaciones

no solo validan nuestro enfoque, sino que también proporcionan una visión detallada de cómo las teorías y las prácticas convergen en el ámbito de sistemas de control esto se detalla en el análisis de resultados.

3.2.2. Descripción Resultados del controlador

Una vez implementados y ajustados los parámetros de nuestro sistema de control, es imperativo analizar su rendimiento y eficacia en diversas condiciones. En esta sección, se desglosa los resultados obtenidos a partir del controlador, iluminando tanto sus fortalezas como las áreas con potencial de mejora. La descripción detallada de estos resultados no sólo valida la solidez del diseño del controlador, sino que también proporciona resultados valiosos sobre su comportamiento en situaciones reales. Es a través de esta introspección que comprende plenamente las capacidades de nuestro sistema y trazar el camino a seguir para futuras optimizaciones.

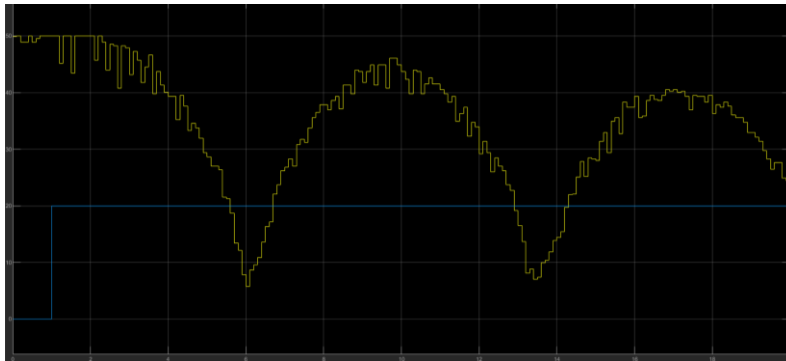
Cálculos referenciales al PID

Dentro del vasto ámbito de los sistemas de control, el controlador PID se destaca por su versatilidad y adaptabilidad a una amplia variedad de sistemas. Su eficacia se basa en la interacción adecuada de sus tres componentes: proporcional, integral y derivativo. Sin embargo, la eficiencia de un PID no se da por sentada; es esencial calibrar correctamente sus parámetros para garantizar un desempeño óptimo. En este inciso, se establece los cálculos referenciales vinculados al PID, ofreciendo un panorama claro sobre cómo se determinan estos valores y su relevancia en la conformación del comportamiento general del controlador. Esta exposición no solo sirve como guía para la sintonización del PID, sino también como una herramienta para entender las razones detrás de cada decisión tomada en el proceso.

En este caso según la respuesta dinámica del sistema que se tiene en la planta física, el cual es de tipo Oscilatorio observado en la Figura 45, por ende, se aplica el método de lazo abierto de Ziegler Nichols.

Figura 45

Respuesta dinámica de la planta física.



Método de Ziegler-Nichols

Al abordar la dinámica de un sistema de control, el comportamiento oscilatorio inherente presenta un desafío significativo para la estabilización y el desempeño deseado del sistema. En tales casos, el uso de lazo cerrado se vuelve imperativo, ya que permite un control más robusto y una respuesta más afinada ante perturbaciones y cambios dinámicos. Al trabajar con lazo cerrado, el sistema tiene la capacidad de ajustarse continuamente en función de las discrepancias entre la salida deseada y la salida real. El método de Ziegler-Nichols en lazo cerrado surge como una solución pragmática para sintonizar controladores PID en contextos donde la planta muestra tendencias oscilatorias. En las siguientes secciones, se detallará este enfoque, subrayando su utilidad y aplicabilidad en nuestro sistema específico.

Preparación:

Asegurando que el sistema está operando en lazo cerrado con un controlador proporcional (solo K_p , sin K_i ni K_d).

Se establece inicialmente K_p en un valor pequeño, y configura K_i y K_d en cero.

Aumenta la Ganancia Proporcional:

Se incrementó gradualmente K_p hasta que la salida del sistema oscile constantemente con una amplitud constante. Esta condición se conoce como el límite de estabilidad.

Se registró este valor crítico de la ganancia como K_{pc} (ganancia proporcional crítica) que dio el valor de 1.

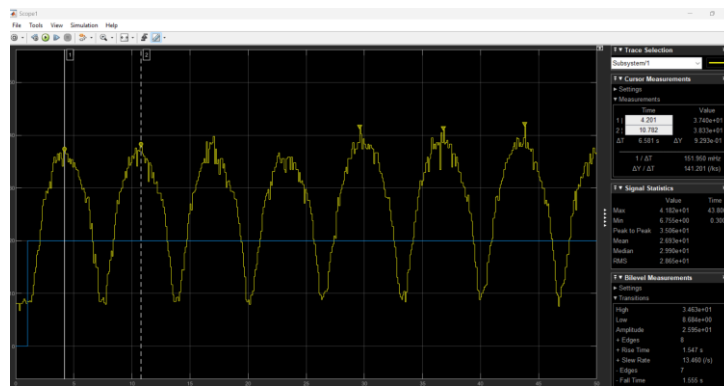
Determina el Periodo de Oscilación Crítico:

Este se mide el periodo de las oscilaciones cuando el sistema está en el límite de estabilidad.

Este periodo es T_c (periodo crítico), Figura 46.

Figura 46

Oscilaciones del sistema.



En este caso se tiene:

$$T_c = 10.782 - 4.201 = 6.581$$

En una Ganancia critica de $K_{pc} = 1$

Utiliza las Tablas de Ziegler-Nichols:

Con K_{pc} y T_c determinados, se sigue las recomendaciones de Ziegler-Nichols para establecer K_p , K_i , y K_d de tu controlador PID:

Para un controlador PID se tiene:

$$K_p = 0.6 * K_{pc} = 0.6$$

$$K_i = \frac{2 * K_p}{T_c} = 0.1825$$

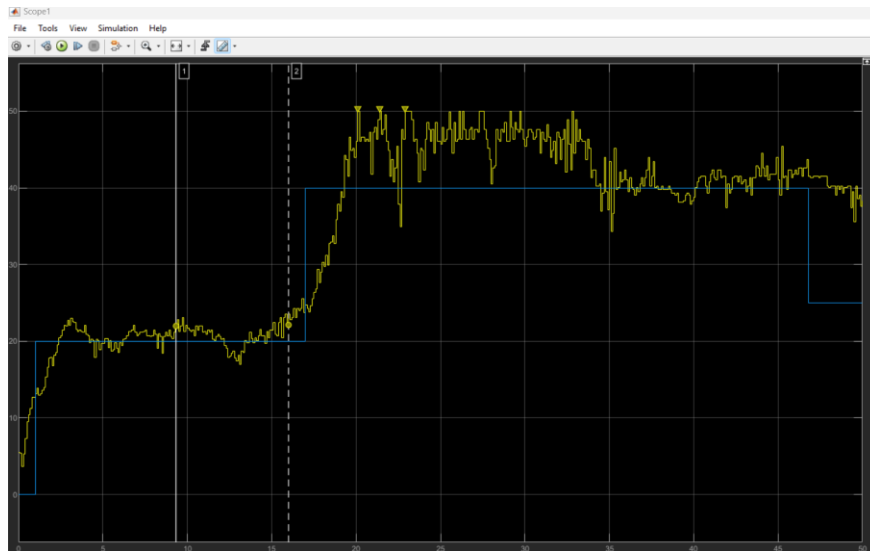
$$K_d = \frac{K_p * T_c}{8} = 0.49995$$

Prueba y Ajuste:

Con los valores iniciales determinados en el paso 4, se probó el sistema. En este caso el comportamiento no es el deseado, ya que existe control, pero hay cierto tipo de perturbaciones como se observa en la Figura 47.

Figura 47

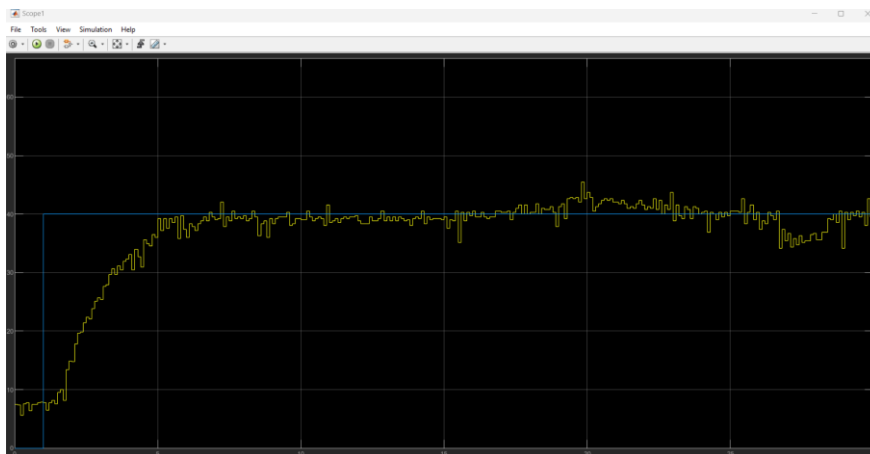
Gráfica del sistema con perturbaciones presentes.



Observando el comportamiento se realizó ajustes manuales en los parámetros hasta obtener una respuesta satisfactoria, la cual se observa a continuación, Figura 48.

Figura 48

Gráfica del sistema con correcciones.



3.3. Análisis de Resultados.

En el campo de los sistemas de control, es vital contrastar el desempeño real de un sistema con las expectativas teóricas establecidas en sus especificaciones iniciales. Estas especificaciones son derivadas de modelos matemáticos y análisis teóricos que anticipan a comportarse el sistema bajo ciertas condiciones ideales. En esta sección, se abordará una comparación detallada entre los resultados experimentales obtenidos a través de pruebas en el sistema y las predicciones teóricas. Esta comparación nos permitirá evaluar la precisión y relevancia del modelo teórico, identificar posibles desviaciones y comprender mejor cualquier discrepancia. Una alineación cercana entre los resultados experimentales y teóricos es indicativa de un modelo robusto y preciso, mientras que las discrepancias significativas señalan la necesidad de revisiones o adaptaciones en el modelo o en el sistema real.

3.3.1. *Contraste de los resultados de la simulación con las predicciones o especificaciones teóricas*

El proceso de modelado y simulación es esencial para anticipar el comportamiento de un sistema en diversas circunstancias y configuraciones. Sin embargo, es crucial contrastar estos resultados simulados con predicciones teóricas o especificaciones predefinidas para validar la precisión y fiabilidad del modelo. Este contraste no solo valida la eficacia del modelo, sino que también revela posibles áreas de mejora o adaptación.

Para una comprensión más clara y concisa de las diferencias entre el modelo teórico, el modelo experimental y las mediciones de la planta real, se presenta a continuación una tabla comparativa. Esta tabla ilustra las distancias estimadas por ambos modelos frente a las distancias reales observadas en la planta para diferentes puntos de referencia. A través de esta representación, será más sencillo visualizar las discrepancias y concordancias entre las predicciones y la realidad, Tabla 1.

Tabla 1. Comparativa de medidas de distancia con el sensor sharp.

Distancia (cm)	Planta real
10	8.5
20	18.5
30	27.98
40	36.9
50	48.5

Observaciones Generales:

En todos los puntos de referencia, el modelo teórico y el modelo experimental tienden a sobreestimar la distancia en comparación con la planta real.

El modelo teórico parece ser consistentemente más preciso que el modelo experimental, con la excepción del punto de 30 cm, donde el modelo experimental es ligeramente más cercano al valor de la planta real.

A medida que la distancia de referencia aumenta, la discrepancia entre la planta real y los modelos parece incrementarse. Es decir, la diferencia entre los modelos y la planta real es menor en distancias cortas y mayor en distancias largas.

Análisis Cuantitativo:

Para 10 cm: La diferencia entre el modelo teórico y la planta real es de 1.36 cm.

Para 20 cm: La diferencia entre el modelo teórico y la planta real es de 2.07 cm.

Para 30 cm: La diferencia entre el modelo teórico y la planta real es de 2.04 cm.

Para 40 cm: La diferencia entre el modelo teórico y la planta real es de 3.66 cm.

Para 50 cm: La diferencia entre el modelo teórico y la planta real es de 1.6 cm.

Al analizar la tabla comparativa de distancias entre los modelos teórico, experimental y la planta real, se evidencia una tendencia consistente: ambos modelos se aproximan con relativa precisión a las mediciones reales, aunque con ciertas desviaciones. Específicamente, el modelo

teórico tiende a ofrecer una ligera sobreestimación, mientras que el modelo experimental se acerca más a los valores reales, con ligeras variaciones. Sin embargo, es importante destacar que, en todas las mediciones, la planta real presenta valores menores que los predichos por ambos modelos, sugiriendo la presencia de factores no considerados en las simulaciones o diferencias en las condiciones de operación.

Es importante recordar que ningún modelo es perfecto, pero un modelo útil es aquel que proporciona una aproximación razonable de la realidad y ofrece datos significativos. Es posible que se requieran ajustes o refinamientos adicionales en los modelos para mejorar la precisión en futuras aplicaciones.

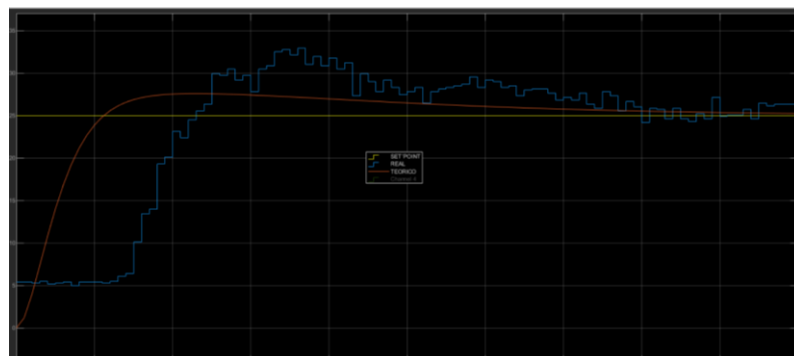
3.3.2. Gráficos Comparativos

Después de presentar los datos en forma tabular, se da un gráfico comparativo que ilustra visualmente (Figura 49) las diferencias entre los tres escenarios a una distancia ya establecida como los 25 cm.

En este gráfico se observa que existe una diferencia en el modelo Físico como en el Real, en el cual el gráfico real contiene diferentes tipos de perturbaciones que no fueron tomadas en cuenta en el modelo teórico, pero el comportamiento dinámico es el mismo producido por el controlador.

Figura 49

Gráfica comparativa de los sistemas.



3.3.3. Cálculo y Análisis de Errores

En esta sección, se calcula los errores absolutos y relativos para cada punto de datos. Los errores absolutos se calculan como la diferencia entre el valor observado y el valor real. Los errores relativos se calculan como el error absoluto dividido por el valor real, generalmente expresado como un porcentaje.

Para calcular estos errores, puedes utilizar las siguientes fórmulas:

$$\text{Error absoluto} = |\text{Valor real} - \text{Valor observado}|$$

$$\text{Error relativo} = \left(\frac{\text{Error absoluto}}{\text{Valor Real}} \right) \times 100\%$$

Con lo siguiente se elabora la siguiente Tabla 2 en la que consta los diferentes errores con las diferentes medidas probadas.

Tabla 2. Comparativa de errores en la medición.

Distancia (cm)	Planta real	Error Absoluto	Error Relativo (%)
10	8.5	1.5	15.00
20	18.5	1.5	7.50
30	27.98	2.02	6.73
40	36.9	3.1	7.75
50	48.5	1.5	3.00

Análisis de errores:

Error a los 10 cm:

Error Absoluto: 1.5 cm

Error Relativo: 15.00%

Este punto tiene el error relativo más alto, indicando que la diferencia entre el valor medido y el verdadero es bastante significativa proporcionalmente.

Error a los 20 cm:

Error Absoluto: 1.5 cm

Error Relativo: 7.50%

Aunque el error absoluto se mantuvo igual en comparación con el punto de 10 cm, el error relativo se redujo a la mitad, mostrando una mejor aproximación al valor verdadero.

Error a los 30 cm:

Error Absoluto: 2.02 cm

Error Relativo: 6.73%

Aquí, el error absoluto ha aumentado en comparación con los puntos de 10 y 20 cm, pero el error relativo es menor que en el punto de 20 cm, indicando una mejor precisión relativa.

Error a los 40 cm:

Error Absoluto: 3.1 cm

Error Relativo: 7.75%

En este punto, tanto el error absoluto como el relativo han aumentado, lo que indica una mayor desviación del valor verdadero.

Error a los 50 cm:

Error Absoluto: 1.5 cm

Error Relativo: 3.00%

Aquí, el error relativo es el más bajo de toda la serie, lo que indica la mejor aproximación al valor verdadero en toda la serie.

Una vez calculados los errores, se procede a discutir los resultados. En esta sección, interpreta los errores calculados, identificando cualquier tendencia notable y proponiendo posibles explicaciones para las diferencias de las medidas observadas.

3.3.4. *Discusión de los Resultados*

Durante el proceso de análisis y experimentación del sistema de control que se ha diseñado, ha sido posible observar varios aspectos destacados que subrayan el potencial y la eficacia del enfoque utilizado.

Concordancia entre los Modelos Teórico y Experimental

A pesar de las diferencias notables que surgieron al comparar los modelos teórico y experimental con los datos de la planta real, es fundamental destacar la relativa concordancia que se encontró. Ambos modelos pudieron trazar una ruta que sigue la tendencia general del comportamiento de la planta real, señalando que las bases teóricas y los datos experimentales utilizados para su desarrollo fueron sólidos y bien fundados.

Los errores relativos, aunque presentes, se mantuvieron en un rango manejable, ilustrando que el sistema tiene una base sólida y que, con ajustes incrementales, la precisión y la eficacia pueden ser mejoradas significativamente.

Estabilidad Conseguida Mediante la Ajuste de Parámetros

A través de las series de pruebas, se observa que la estabilidad del sistema podía alcanzarse ajustando los parámetros del controlador PID, particularmente omitiendo el componente derivativo (K_d). Este hallazgo es crucial ya que indica que la estabilidad no es un problema intrínseco y que el sistema es optimizado para operar de manera estable y eficiente.

Valiosa Información Obtenida del Método de Ziegler-Nichols

Aunque se enfrentan desafíos con los parámetros iniciales proporcionados por el método de Ziegler-Nichols, el método mismo demostró ser una herramienta valiosa, guiándonos hacia una comprensión inicial de las dinámicas del sistema y proporcionando un punto de partida desde el cual se puede iterar.

En vista de los resultados obtenidos, se afirma un camino positivo hacia el establecimiento de un sistema de control robusto y preciso. La correlación observable entre los datos teóricos, experimentales y reales enfatiza que el fundamento teórico y el enfoque metodológico adoptado han sido apropiados y que el modelo tiene un alto potencial de ser perfeccionado a un estado de alta precisión y eficacia.

Mientras se reconoce que existen áreas para mejoras incrementales, la progresión hasta ahora indica una trayectoria ascendente y prometedora hacia la realización de un sistema de control que no solo puede simular con precisión la dinámica de la planta real, sino también manipular la posición de la esfera de manera efectiva y controlada.

Al continuar refinando los parámetros y ajustando el controlador basándonos en observaciones prácticas directas, se tiene todas las razones para anticipar mejoras sustanciales en la funcionalidad y el rendimiento del sistema.

3.3.5. Propuesta para prácticas de laboratorio

En este contexto, las prácticas de laboratorio emergen como una herramienta educativa y de investigación vital. Estas no sólo facilitan la comprensión profunda de los principios teóricos involucrados, sino que también promueven el desarrollo de habilidades prácticas esenciales en el campo de la ingeniería de control.

Con la infraestructura existente, centrada en nuestra planta constituida por una barra inclinada controlada por un servomotor y monitoreada por un sensor Sharp para medir la distancia, se presenta una oportunidad significativa para crear un entorno de aprendizaje práctico y dinámico. El sistema en sí, que implica una mecánica compleja y una interacción significativa entre sus componentes, ofrece un campo fértil para la exploración práctica de una variedad de conceptos centrales en la ingeniería de control.

Dicho esto, el presente segmento se dedica a proponer una serie de prácticas de laboratorio diseñadas para proporcionar una experiencia práctica valiosa, fomentando así un aprendizaje inmersivo y aplicado. Estas prácticas estarán orientadas hacia la consecución de una serie de objetivos educativos claramente definidos, permitiendo así a los estudiantes o investigadores construir una comprensión sólida y holística del funcionamiento y control de sistemas dinámicos.

Práctica de Laboratorio 1: Calibración de Sensores y Actuadores

- **Objetivo**

Aprender y aplicar técnicas de calibración para el sensor sharp y el servomotor.

- **Materiales**

- Planta física (barra inclinada con sensor sharp y servomotor)
- Esfera de caucho
- Regla o cinta métrica
- Software MATLAB y Simulink

- **Procedimiento**

1. Configure el sistema asegurándose de que todos los componentes estén conectados correctamente.
2. Realice una serie de mediciones para calibrar el sensor sharp utilizando una regla o cinta métrica para verificar las lecturas del sensor.
3. Establezca una relación lineal entre el ángulo de rotación del servomotor y el ángulo de inclinación de la barra.
4. Utilizando el software MATLAB y Simulink, realice pruebas para verificar la precisión de la calibración.

- **Preguntas de Reflexión**

- ¿Cuál es la importancia de calibrar correctamente los sensores y actuadores en un sistema de control?
- ¿Cómo afectan los errores de calibración al rendimiento del sistema?

Práctica de Laboratorio 2: Diseño y Ajuste de un Controlador PID

- **Objetivo**

Diseñar y ajustar un controlador PID para controlar la posición de la esfera en la barra inclinada.

- **Materiales**

- Planta física (barra inclinada con sensor sharp y servomotor)
- Esfera de caucho
- Software MATLAB y Simulink

- **Procedimiento**

1. Establezca un modelo matemático del sistema utilizando las ecuaciones de movimiento.
2. Diseñe un controlador PID inicial utilizando el método de Ziegler-Nichols.
3. Implemente el controlador en Simulink y realice simulaciones para validar su diseño.
4. Ajuste los parámetros del controlador PID basándose en las respuestas del sistema obtenidas durante las simulaciones y las pruebas experimentales.

- **Preguntas de Reflexión**

- ¿Cómo afecta cada parámetro (K_p , K_i , K_d) al rendimiento del controlador?
- ¿Qué técnicas adicionales podrían utilizarse para sintonizar el controlador PID de forma más efectiva?

Práctica de Laboratorio 3: Validación Experimental

- **Objetivo**

Validar el modelo del sistema y el controlador PID diseñado mediante experimentación práctica.

- **Materiales**

- Planta física (barra inclinada con sensor sharp y servomotor)
- Esfera de caucho
- Software MATLAB y Simulink

- **Procedimiento**

Realice experimentos prácticos utilizando la planta física para validar el modelo del sistema y el controlador PID diseñado.

Compare los resultados experimentales con las predicciones teóricas y los resultados de simulación.

Identifique y analice cualquier discrepancia entre los resultados experimentales y las predicciones/simulaciones.

- **Preguntas de Reflexión**

- ¿Los resultados experimentales validaron efectivamente el modelo y el controlador?
- ¿Qué factores podrían haber contribuido a cualquier discrepancia observada entre los resultados experimentales y las predicciones/simulaciones?

Estas prácticas de laboratorio están diseñadas para brindar una comprensión práctica y profunda de los conceptos involucrados en el control de sistemas dinámicos.

CAPITULO IV: CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- Se analizaron los parámetros en la construcción del sistema como la estructura por acero inoxidable para la planta, ya que ofrecía una mayor resistencia y durabilidad en comparación con los materiales de acrílico. También, se definieron condiciones de diseño para las barras conectadas al servomotor, garantizando que la relación entre el ángulo del servomotor y la inclinación de la barra fueran en los rangos de inclinación determinadas de -10 grados a 15 grados.
- De los experimentos realizados, se evidenció que el sensor de distancia posee una relación no lineal entre el voltaje y su respuesta. Esta característica resaltó la importancia de una adecuada interpretación y procesamiento de los datos. Además, debido a la sensibilidad del sensor a interferencias luminosas externas, se tomó precaución adicional al interpretar las mediciones. Para abordar la no linealidad, se recopilaron datos experimentales y, a partir de estos, se utilizó una función logarítmica en Matlab. La misma que proporciona una representación linealizada de la relación entre el voltaje y la distancia, permitiendo interpretaciones más directas y precisas. Adicionalmente, se aplicaron filtro de media exponencial para minimizar el impacto de las interferencias luminosas, asegurando así la confiabilidad de las lecturas del sensor en el sistema de control.
- Al comparar el modelo real con respecto al teórico, se observaron desviaciones en las distancias entre la esfera y el punto de sensado. El modelo real mostró variaciones en un rango de 1.36 cm y 3.66 cm con respecto al teórico, los valores

registrados están en un rango cercano lo que demuestran que el comportamiento del sistema está acorde a las expectativas.

- Dada la respuesta dinámica oscilatoria del sistema, se optó por el método de Ziegler-Nichols para la sintonización del PID. Este enfoque, centrado en la caracterización de la respuesta del sistema ante perturbaciones, facilitó la determinación de los parámetros del controlador.
- Se contrastaron las respuestas del modelo teórico con las mediciones reales en diversas distancias. Se identificó un error relativo promedio de 8% al analizar el comportamiento en todo el rango de distancias. Se observaron desviaciones en ciertas distancias específicas, el error general demuestra una coherencia del sistema con el modelo teórico.

4.2.Recomendaciones

- La ejecución de este proyecto ha brindado una valiosa comprensión de los principios detrás del control de sistemas dinámicos. Sería beneficioso mantener esta inercia y aprovechar el aprendizaje adquirido para futuros proyectos que involucren conceptos similares.
- Aunque se notaron algunas diferencias en las mediciones entre el modelo teórico, el modelo experimental y la planta real, los resultados están bastante alineados. Se recomienda realizar ajustes finos en el controlador PID para alinear aún más los resultados de la planta real con los modelos teóricos y experimentales.
- Considerando que se ha logrado un significativo progreso hasta ahora, sería provechoso seguir explorando cursos y talleres especializados para aumentar aún más el nivel de pericia en sistemas de control y MATLAB.

- Para los estudiantes que interactúan con la interfaz de este sistema, se recomienda dedicar tiempo a familiarizarse con la documentación asociados antes de iniciar cualquier experimento o prueba. Es esencial entender las limitaciones y capacidades del sistema para garantizar resultados precisos y evitar posibles fallos o daños. Se sugiere también que los estudiantes adopten un enfoque sistemático al realizar cambios en los parámetros o al configurar nuevas pruebas, documentando cuidadosamente cada paso y observación.

BIBLIOGRAFÍA

- AALBORG. (2021). *LabVIEW Integration for XFM instruments*. Manufacturer of High Quality Flow Instrumentation. https://www.aalborg.com/aalborg-labview-integration-for-xfm-instruments_pr/17
- Acrome. (2015, June 9). *Ball and Beam*. <https://acrome.net/product/ball-and-beam>
- Amazon. (2022a). *Paquete de 20 rodamientos de esferas 623ZZ 0.118 in x 0.394 in x 0.157 in*. <https://www.amazon.com/Camoo-ZZ-Bearings/dp/B092Z2QB6X?th=1>
- Amazon. (2021b). *Servomotor*. <https://www.amazon.com/ZOSKAY-Coreless-Digital-Stainless-arduino/dp/B07S9XZYN2?th=1>
- Amazon. (2023). *Paquete de 2 conectores de dirección de acoplamiento de eje universal*. <https://www.amazon.com/Universal-Coupler-Coupling-Steering-Connector/dp/B0782T3M68>
- Autycom. (2023, June 10). *¿Qué es un sistema de control? - AUTYCOM*. <https://www.autycom.com/que-es-un-sistema-de-control/>
- Buza, G., Milton, J., Bencsik, L., & Insperger, T. (2020). Establishing metrics and control laws for the learning process: ball and beam balancing. *Biological Cybernetics*, 114. <https://doi.org/10.1007/s00422-020-00815-z>
- Carrasco, L. (2020). *Diseño e implementación de un control PID IMC para planta piloto de refrigeración*. Universidad de Piura.
- Carrillo, A. (2011). *Sistemas Automáticos de Control Fundamentos básicos de análisis y modelado* (Vol. 2).

- Contreras, M. (2015). Control por eventos para la estabilización del péndulo lineal. Valencia: Escuela Técnica Superior de Ingeniería del Diseño.
- Dale, S., Thomas, E., & Duncan, M. (2004). *Process dynamics and control: Vol. segunda*.
- Fidias, G. (2016). *El proyecto de investigación Introducción a la metodología científica* (McGraw-Hill., Ed.; sexta).
- Guilcamaigua, D., & Villacis, M. (2022). *Análisis del control PID de un sistema didáctico de tanques*.
- Ionos. (2019, November 3). *El modelo en cascada en el desarrollo de software - IONOS*.
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>
- Mogrovejo, D. (2017). *Diseño e implantación de un sistema de cuatro tanques interconectados con control PID robusto multivariable*.
- Morales, R., & Ramírez, R. (2013). *Sistema de control moderno: Vol. primera edición*.
www.ebookstec.com
- Muzo, J. (2018). *Diseño de una casa inteligente utilizando plataformas Android y Arduino*.
- Ortiz, F. (2004). *Modelado y control PD-difuso en tiempo real para el sistema barra-esfera*.
- Ogata, K. (2010). *Ingeniería de control moderna*. Pearson Educación.
- Quanser. (2023, June 9). *Ball and Beam - Quanser*. <https://www.quanser.com/products/ball-and-beam/#overview>
- Romero, P. (2021). *Desarrollo de un sistema de control y monitoreo a distancia de una línea de proceso industrial con Labview e internet*.
- Ruiz, C. (2016, November 23). *Librerías en Arduino – Open Lanuza*.
<https://openlanuza.com/librerias-en-arduino/>
- Sira, H., Márquez, R., Rivas, F., & Orestes, S. (2004). *Control de Sistemas No Lineales*.

SKU. (2023, June 18). *Tiendamia del mundo a tu puerta - tiendamia.com.*

[https://tiendamia.com/ec/producto?amz=B089NFCZ9F&pName=Filamento%20PETG%20para%20impresora%203D,%20bobina%20de%20filamento%20SUNLU%20Super%20Neat,%20filamento%20PETG%20resistente%20de%201,75%20mm,%20precisi%C3%B3n%20dimensional%20+/-%200,02%20mm,%20bobina%20de%201%20kg%20\(2,2%20libras\),%20320%20metros,%20PETG%20azul/&gclid=Cj0KCQjw1rqkBhCTARIsAAHz7K0WLP4KMH9yfmPJJT97FEV1EwioIrgIPV9afGqceU6WcKSGFNpG5UaAsjBEALw_wcB](https://tiendamia.com/ec/producto?amz=B089NFCZ9F&pName=Filamento%20PETG%20para%20impresora%203D,%20bobina%20de%20filamento%20SUNLU%20Super%20Neat,%20filamento%20PETG%20resistente%20de%201,75%20mm,%20precisi%C3%B3n%20dimensional%20+/-%200,02%20mm,%20bobina%20de%201%20kg%20(2,2%20libras),%20320%20metros,%20PETG%20azul/&gclid=Cj0KCQjw1rqkBhCTARIsAAHz7K0WLP4KMH9yfmPJJT97FEV1EwioIrgIPV9afGqceU6WcKSGFNpG5UaAsjBEALw_wcB)

Sosa, I. (2018). *Diseño de in programador lógico programable usando microcontrolador ATMEGA y lenguaje LADDER para aplicaciones de laboratorio.*

Tecquipment. (2021). *BALL AND BEAM APPARATUS.* <https://www.tecquipment.com/ball-and-beam-apparatus>

Ubuy. (2023, June 18). *ARRIS 8mm 6mm x 8mm x 500mm 3K Roll Wrapped 100% | Ubuy Morocco.*

https://www.ubuy.ma/en/product/4UPFRJK-arris-8mm-6mm-x-8mm-x-500mm-3k-roll-wrapped-100-carbon-fiber-tube-matt-surface-2-pcs?campaign_source=google&campaign_medium=&campaign_name=&campaignid=20283255816&gclid=Cj0KCQjw1rqkBhCTARIsAAHz7K2pq55m7ujDQLe3MxC9ndRPS6lsGPD43lx4JPE8S3lu3n8sEQ7QXEaAglkEALw_wcB

Xataka. (2022, September 23). *Qué es Arduino, cómo funciona y qué puedes hacer con uno.*

<https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>

ANEXOS

Anexo 1: Evidencia fotográfica

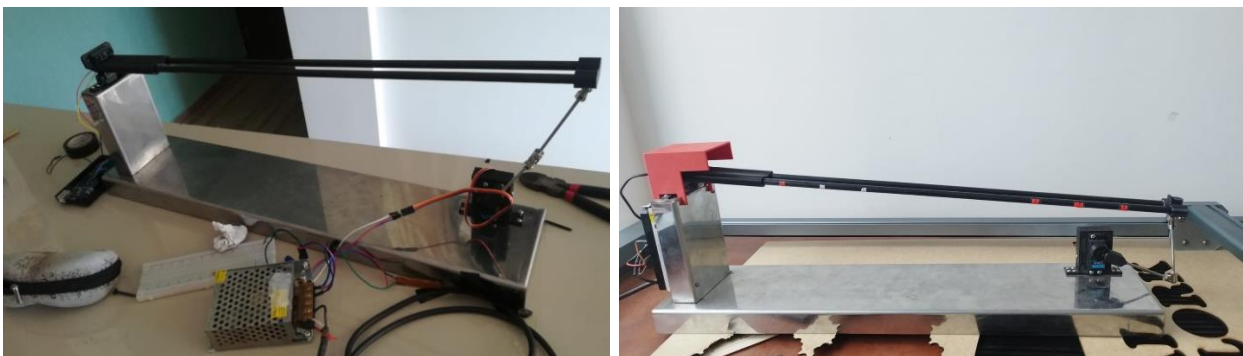
- 1er prototipo hecho en PLA



- Corte de acero inoxidable, para el segundo prototipo.



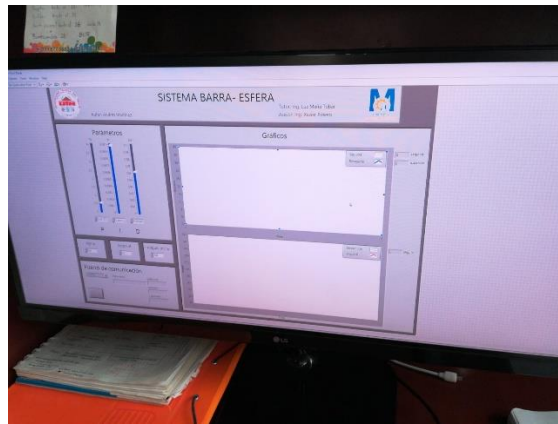
- Ensamble del segundo prototipo



- Toma de medidas de la planta con el software

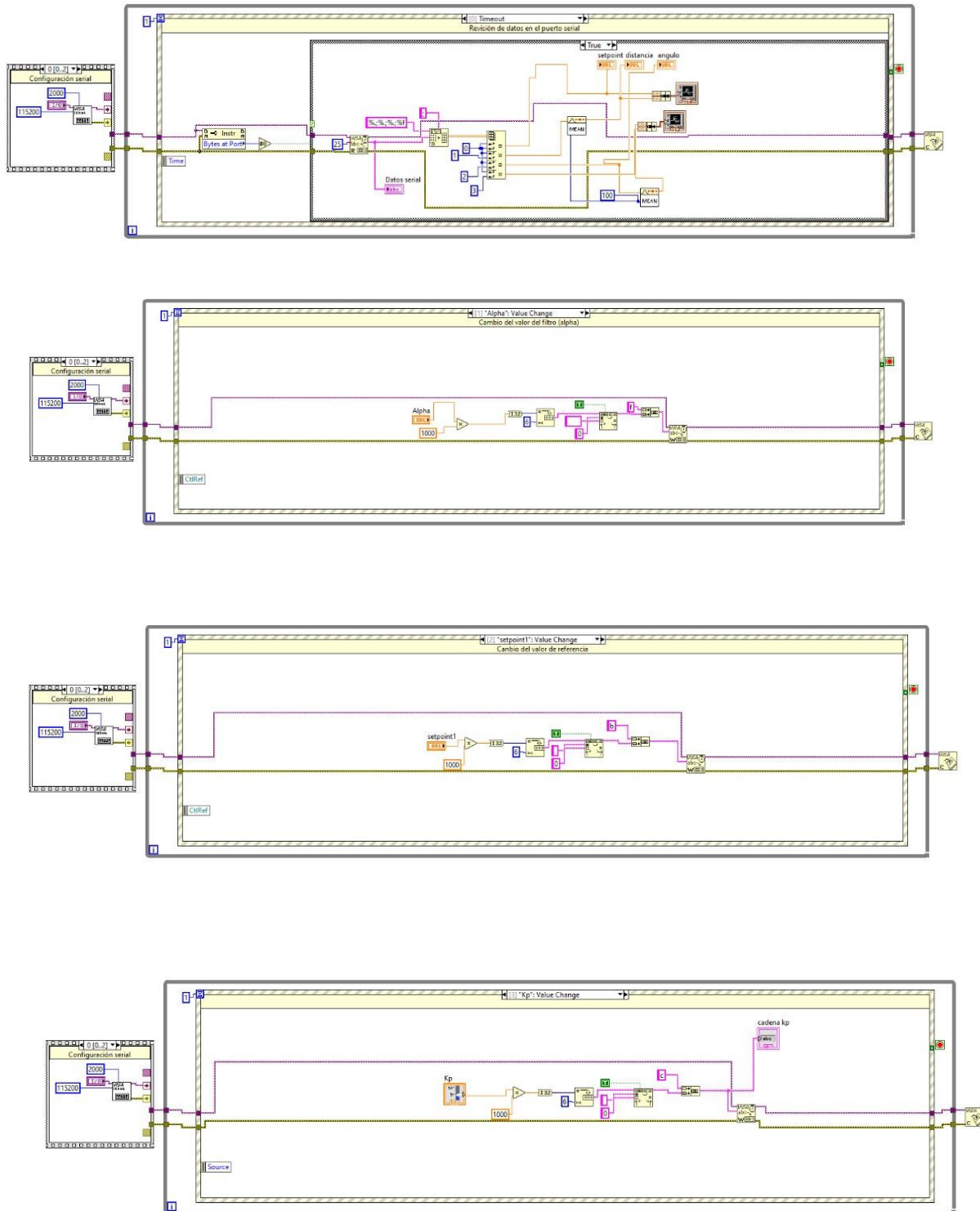


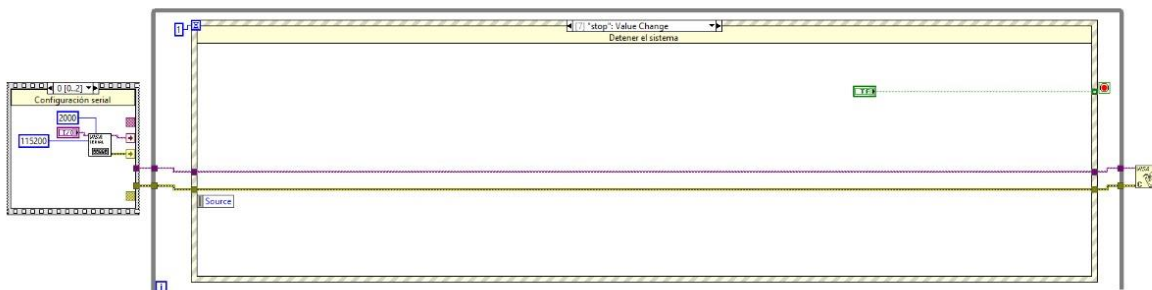
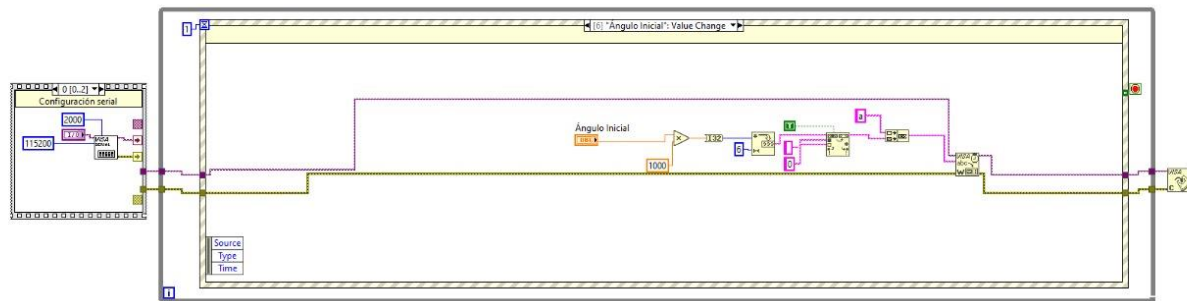
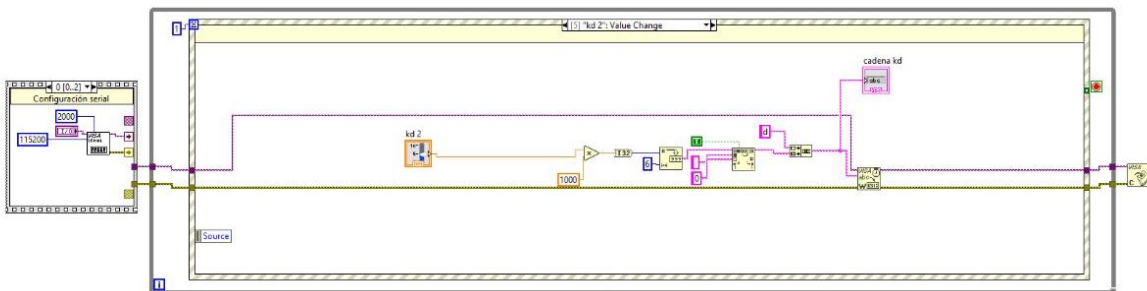
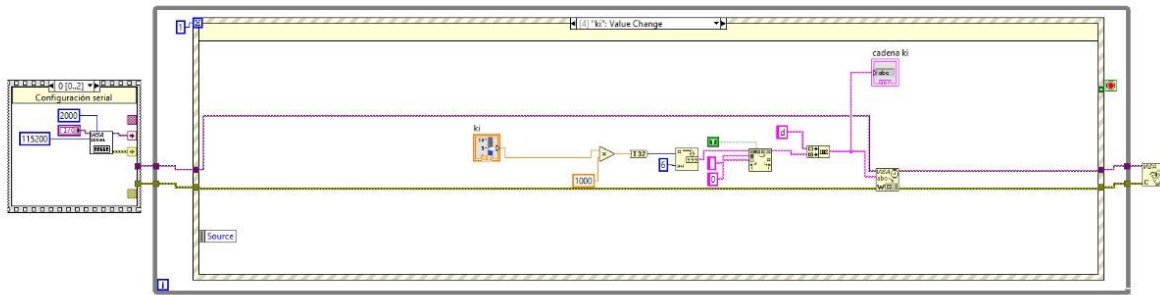
- Interfaz de practicas



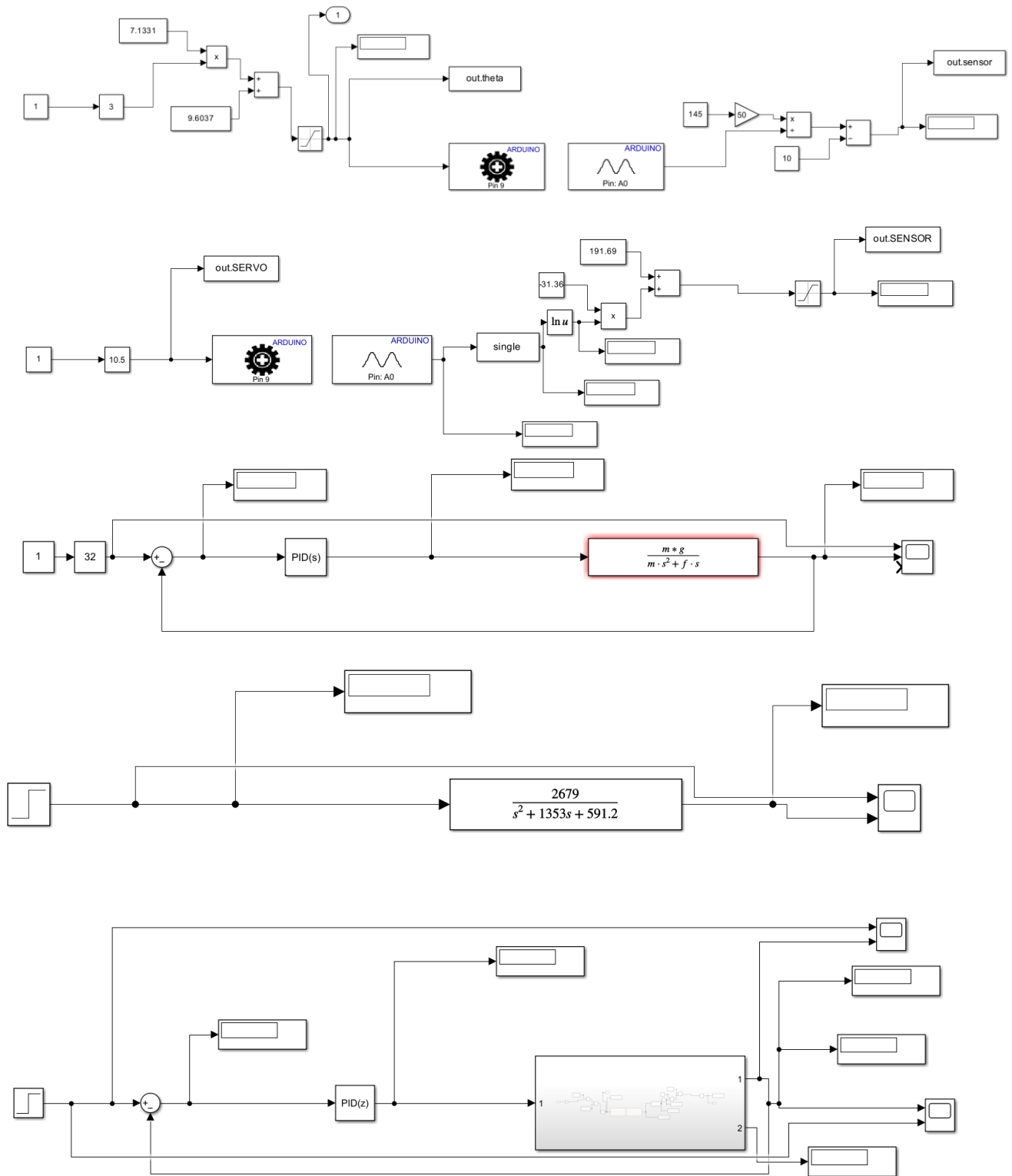
Anexo 2: Algoritmos

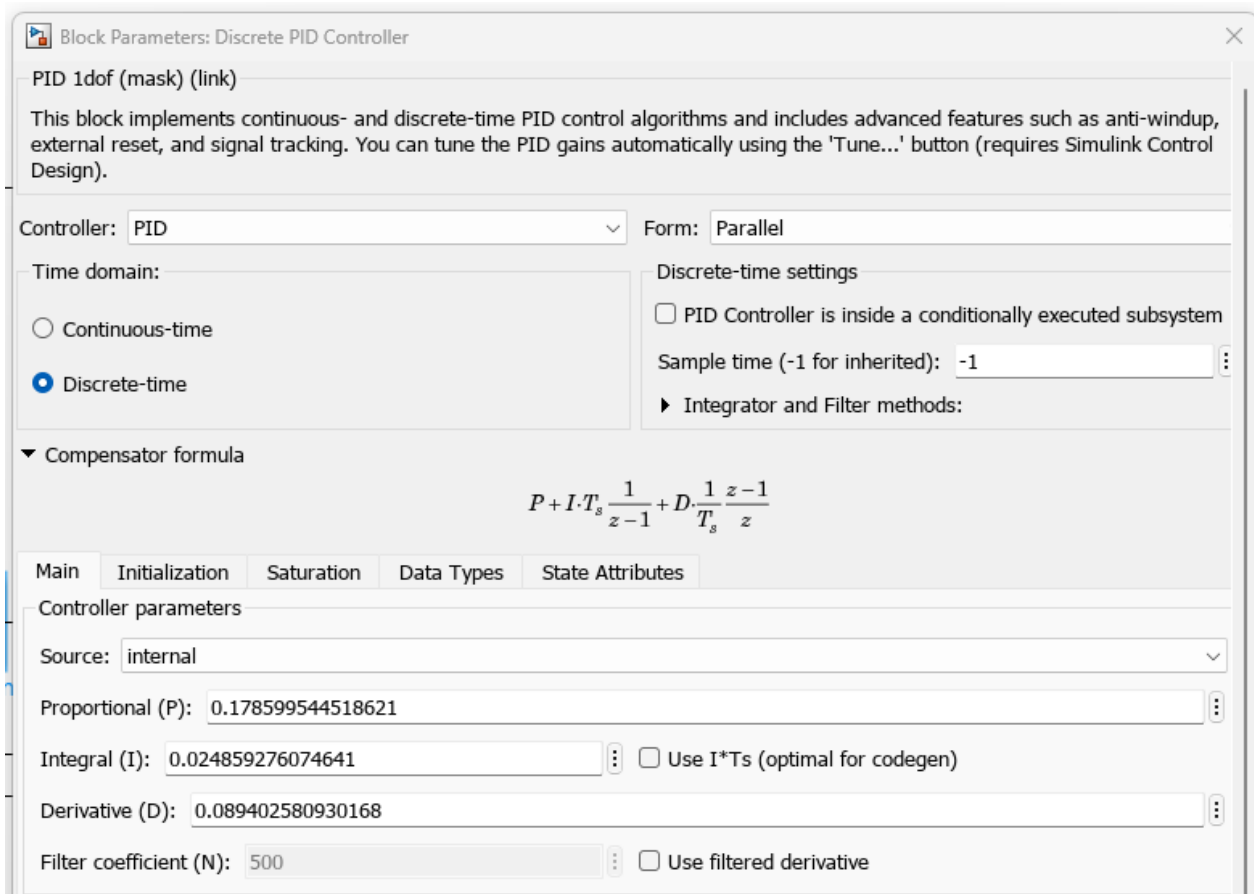
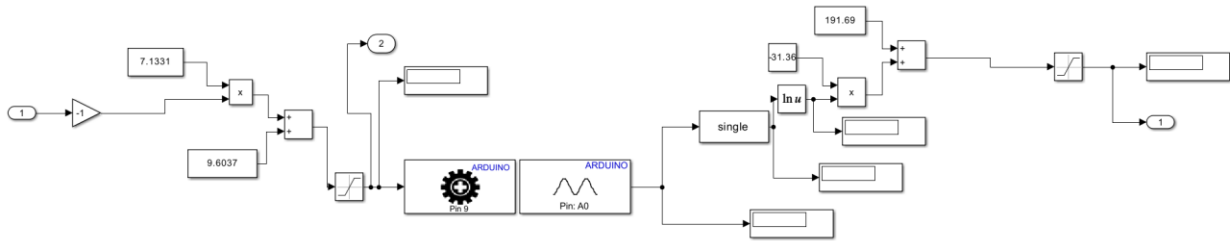
- LabVIEW





• Matlab SIMULINK





- Arduino

```
#include <Servo.h>
```

```
#include <math.h>
```

```
float variables[10]; // Aquí se almacenan los parámetros de configuración
```

```
byte i = 0;
```

```
char input[6];
```

```
float output = 0;

float dist_sharp;

float dist_sharp_sm;

int dist_sharp_int;

/*Filtro de media exponencial*/

float alpha = 0.4;

/*Variables de control PID*/

float error = 0;

float last_error = 0;

float integral = 0;

float derivativo = 0;

float a = 102; //ángulo inicial

float Kp = 0.5;

float Ki = 0.0000; //presicion

float Kd = 0.0000; // velocidad ante cambios

int angle = variables[0];

Servo myServo; // Crear un objeto de Servo

void setup() {

    Serial.begin(115200);

    Serial.flush();

    myServo.attach(9);

    variables[0] = a;

    variables[1] = 25;

    variables[2] = Kp;
```

```

variables[3] = Ki;

variables[4] = Kd;

variables[5] = alpha;

}

void loop() {

  distancia_sharp();

  PID();

  angle = (int)constrain(output, 60, 142);

  myServo.write(angle);

  imprimir();

  delay(0);

}

void PID() {

  // Calcula la salida del controlador PID

  error = dist_sharp - variables[1];

  integral += error;           // Acumula el error para el término integral

  derivativo = error - last_error; // Diferencia el error actual con el anterior para el término
derivativo

  output = variables[0] + variables[2] * error + variables[3] * integral + variables[4] *
derivativo; // Calcula la salida del controlador PID

  last_error = error;         // Guarda el error actual para la próxima iteración

}

void distancia_sharp() {

  dist_sharp = analogRead(A0);

```

```

alpha = variables[5];

dist_sharp_sm = alpha * dist_sharp + (1 - alpha) * dist_sharp_sm;

//linealización del sensor

//dist_sharp =103.79*exp(-0.01*dist_sharp_sm); //esfera acero
dist_sharp =38635 * pow(dist_sharp_sm, -1.444); // esfera verde
}

/*

void imprimir() {

  Serial.print(0);

  Serial.print(',');

  Serial.print(40); //45

  Serial.print(',');

  Serial.print(variables[1]);

  Serial.print(',');

  Serial.println(dist_sharp);

}

/*Evento serial para recepción de datos*/

void imprimir() {

  Serial.print(variables[1]); //setpoint

  Serial.print(',');

  Serial.print(dist_sharp);

  Serial.print(',');

  Serial.print(angle);

  Serial.print(',');

```

```
Serial.println(variables[0]); //angulo de referencia

}

void serialEvent() {

    float value;

    if (Serial.available()) {

        input[i] = (char)Serial.read();

        int pos = input[0] - 'a';

        if (pos >= 0 && pos < 10)

            i++;

        else

            i = 0;

        if (i == 6) {

            // Convertir caracteres en un número entero

            int val = 0;

            for (int j = 1; j < 6; j++)

                val = val * 10 + input[j] - '0';

            value = (float)val / 100;

            // Asignar los parámetros leídos basados en el orden alfabético 'a'=0, 'b'=1...

            variables[pos] = value;

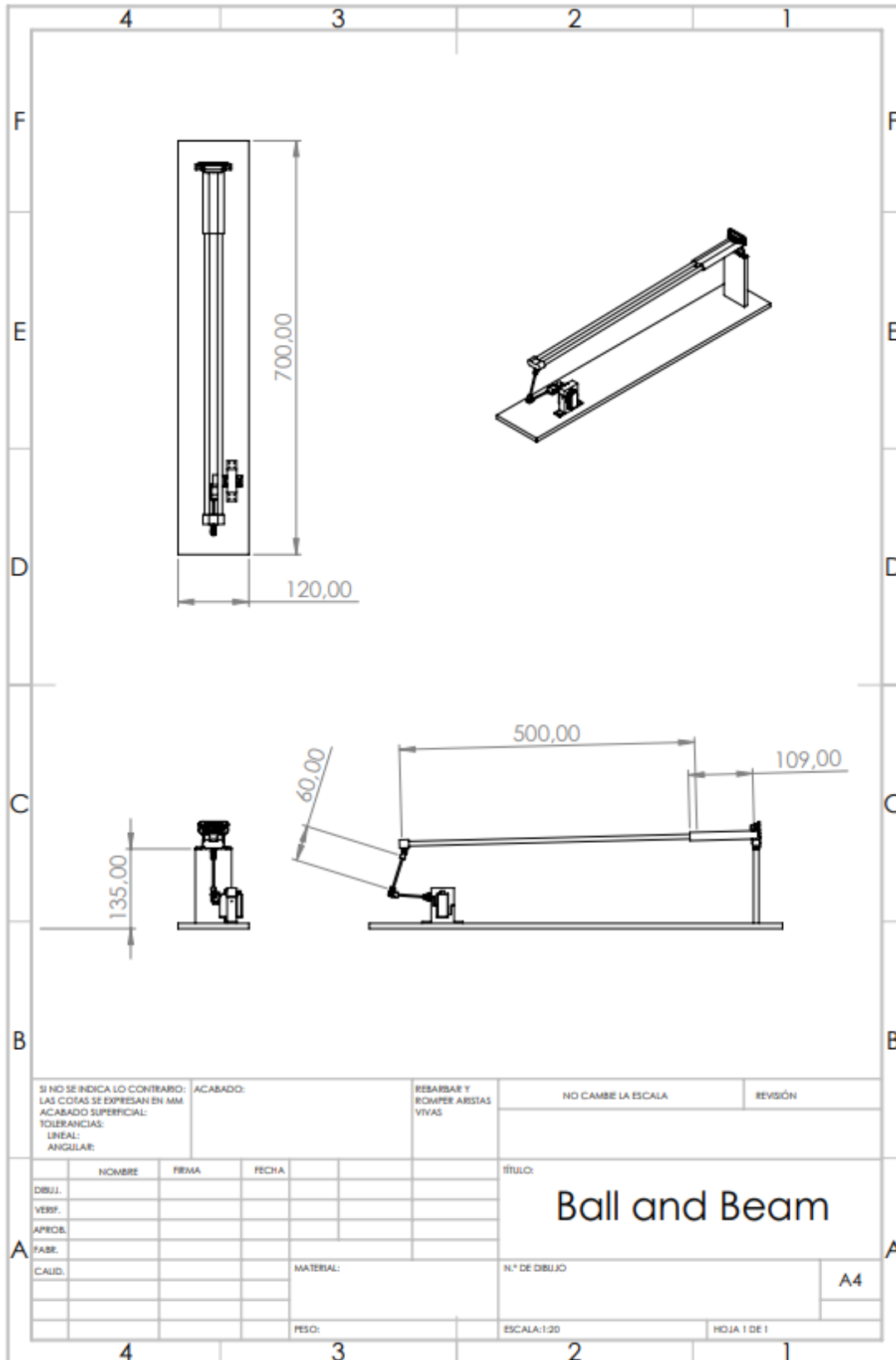
            i = 0;

        }

    }

}
```

Anexo 3: Planos de la planta



Anexo 4: Datasheet

- Arduino mega (<https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf>)



Arduino® MEGA 2560 Rev3

Product Reference Manual

SKU: A000067



Description

Arduino® Mega 2560 is an exemplary development board dedicated for building extensive applications as compared to other maker boards by Arduino. The board accommodates the ATmega2560 microcontroller, which operates at a frequency of 16 MHz. The board contains 54 digital input/output pins, 16 analog inputs, 4 UARTs (hardware serial ports), a USB connection, a power jack, an ICSP header, and a reset button.

Target Areas

3D Printing, Robotics, Maker

- Sharp GP2Y0a21f98

(https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf)

SHARP

GP2Y0A21YK0F

GP2Y0A21YK0F

Distance Measuring Sensor Unit
Measuring distance: 10 to 80 cm
Analog output type



■ Description

GP2Y0A21YK0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector), IRED (infrared emitting diode) and signal processing circuit.

The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method.

This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

■ Features

1. Distance measuring range : 10 to 80 cm
2. Analog output type
3. Package size : 29.5×13×13.5 mm
4. Consumption current : Typ. 30 mA

■ Agency approvals/Compliance

1. Compliant with RoHS directive (2011/65/EU)

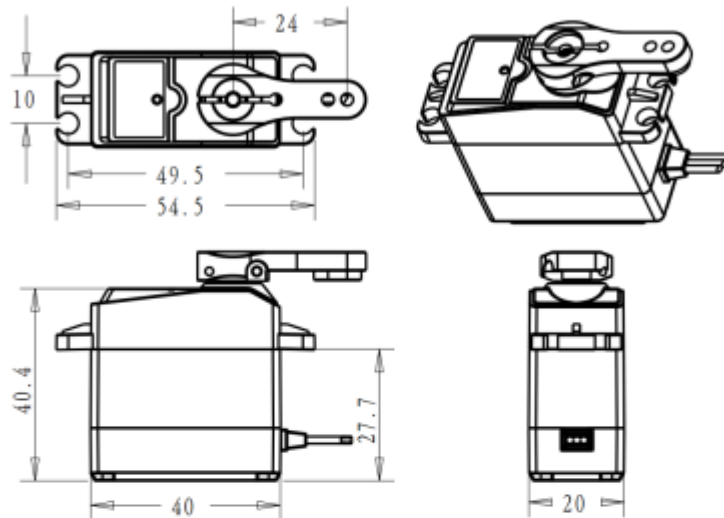
■ Applications

1. Touch-less switch
(Sanitary equipment, Control of illumination, etc.)
2. Robot cleaner
3. Sensor for energy saving
(ATM, Copier, Vending machine)
4. Amusement equipment
(Robot, Arcade game machine)

- Servomotor DS3235SG (https://github.com/microrobotics/DS3235-270/blob/master/DS3235-270_datasheet.pdf)



1. Dimensions



2. Specification

No.	Item	Specification
1-1	Storage Temperature Range	-30°C ~ 80°C
1-2	Operating Temperature Range	-15°C ~ 70°C
1-3	Operating Voltage Range	5-7.4V

Manual de Usuario - Sistema de Control Ball and Beam con LabVIEW

Introducción

Bienvenido al Manual de Usuario del Sistema de Control Ball and Beam utilizando LabVIEW. Este manual le guiará a través de las funciones y características del programa, así como su interacción con la planta a través de LabVIEW y un Arduino. Este sistema permite controlar la posición de una esfera en una superficie inclinada (ball and beam) mediante el ajuste de los parámetros de control.

Requisitos del Sistema

Antes de comenzar, asegúrese de que tu ordenador tenga las siguientes características:

- Sistema Operativo:
 - Windows 10/8.1/7 SP1
 - Windows Server 2012 R2, 2016, 2019
 - LabVIEW también tiene versiones para Linux y macOS, pero la mayoría de los módulos y toolkits están optimizados para Windows.
- Procesador:
 - Intel Core i5 o equivalente (se recomienda Core i7 o superior)
- Memoria RAM:
 - Mínimo 4 GB de RAM (se recomiendan 8 GB o más)
- Espacio en Disco Duro
 - 5 GB de espacio libre para la instalación completa. Sin embargo, este número puede variar según qué módulos o toolkits adicionales.
- Tarjeta de Video:
 - Se recomienda que sea compatible con DirectX 11 para un rendimiento óptimo, especialmente si se trabaja con interfaz de usuario avanzada.
- Resolución de Pantalla:
 - Se recomienda una resolución de 1024 x 768 pixels o superior.
- Software Adicional:
 - Algunas versiones de LabVIEW pueden requerir software adicional como .NET Framework o ciertas librerías.

Por consiguiente, e instalado LabVIEW, el sistema debe cumplir con los siguientes requisitos:

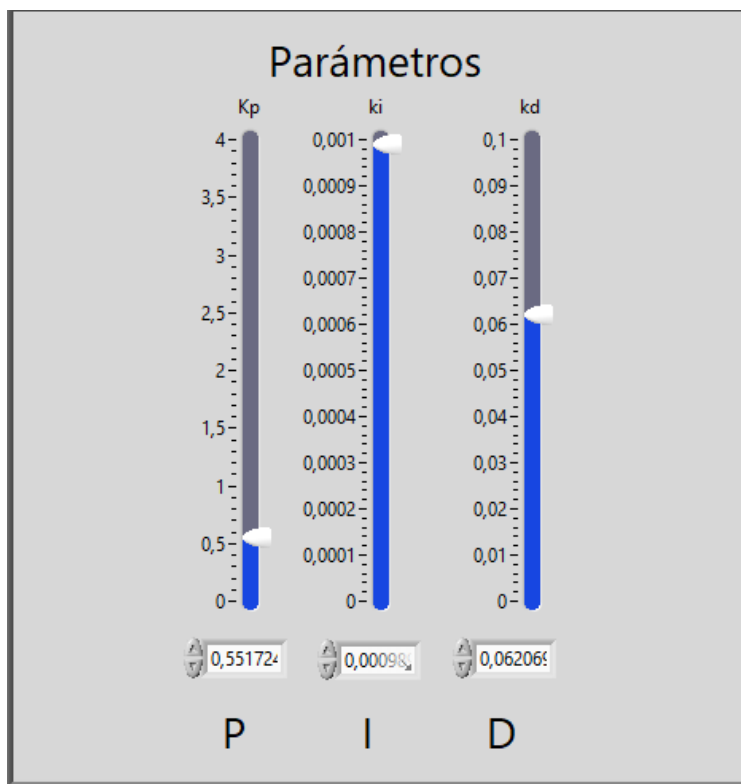
- Conexión al puerto COM al que está conectado el Arduino de la planta.
- Planta Ball and Beam configurada y conectada al Arduino.

Interfaz de Usuario

La interfaz de LabVIEW consta de varias secciones que le permiten controlar y monitorear el sistema de Ball and Beam:

Sección de Control de Parámetros PID

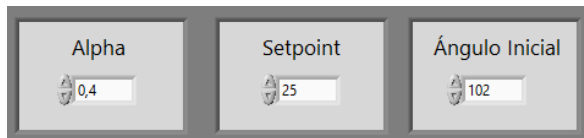
En esta sección, puede ajustar los valores de los parámetros del controlador PID utilizando tres deslizadores o el teclado numérico:



- **Kp (Ganancia Proporcional):** Controla la influencia proporcional en la respuesta del sistema.
- **Ki (Ganancia Integral):** Controla la influencia de la acumulación de errores a lo largo del tiempo.
- **Kd (Ganancia Derivativa):** Controla la influencia de la velocidad de cambio de error.

Sección de Configuración de Planta

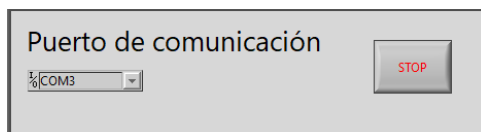
Aquí, puede realizar configuraciones relacionadas con la planta y la medición:



1. Alpha (Reducción de Ruido): Ajuste este valor para reducir el ruido en la medición. Valores más cercanos a 1 reducen el ruido, pero ralentizan la respuesta del sensor.
2. Set Point (Punto de Referencia): Establezca el valor al que desea que la esfera se mueva.
3. Ángulo Inicial: Configure el ángulo inicial de la superficie donde se encuentra la planta.

Sección de Configuración de Puerto COM

Configure el puerto COM al que está conectada la planta. Asegúrese de que coincida con el puerto al que está conectado el Arduino.



Sección de Gráficos

En esta sección, se muestran gráficos en tiempo real para monitorear el sistema:



1. Gráfico de Posición (eje Y vs. Tiempo): Muestra la posición de la esfera en el eje Y en función del tiempo.
2. Gráfico de Ángulo (eje Y vs. Tiempo): Muestra el ángulo de inclinación en función del tiempo.

Uso del Programa

1. Asegúrese de que la planta Ball and Beam esté correctamente conectada al Arduino y que este esté conectado a la computadora a través del puerto COM configurado en la sección correspondiente.
2. Realice experimentos junto con la rúbrica de las prácticas de laboratorio, para obtener los valores adecuados de K_p , K_i y K_d según los objetivos de su práctica o proyecto.
3. Anote los valores calculados de K_p , K_i y K_d que desea ingresar en el sistema.
4. En la interfaz de LabVIEW, configure los deslizadores o utilice el teclado numérico para ingresar los valores obtenidos de K_p , K_i y K_d en la sección de control de Parámetros PID.
5. Configure otros parámetros según sea necesario, como Alpha, Set Point y Ángulo inicial.
6. Haga clic en el botón "Iniciar Control" para aplicar los valores de K_p , K_i y K_d que ha ingresado.
7. Monitoree la respuesta del sistema y realice análisis en función de los valores de los parámetros PID que ha seleccionado.
8. Puede detener el control en cualquier momento haciendo clic en el botón "STOP".

Este enfoque le permite ajustar y experimentar con diferentes configuraciones de controlador PID para lograr los resultados deseados en sus prácticas de sistemas de control. Recuerde que los cálculos y experimentos para determinar los valores de K_p , K_i y K_d se realiza por separado y luego ingresarlos a través de la interfaz de LabVIEW para su implementación en el sistema.

Prácticas de Laboratorio

Práctica de Laboratorio 1: Calibración de Sensores y Actuadores

- **Objetivo**

Aprender y aplicar técnicas de calibración para el sensor sharp y el servomotor.

- **Materiales**

- Planta física (barra inclinada con sensor sharp y servomotor)

- Esfera de caucho
 - Regla o cinta métrica
 - Software MATLAB y Simulink
 - **Procedimiento**
5. Configure el sistema asegurándose de que todos los componentes estén conectados correctamente.
 6. Realice una serie de mediciones para calibrar el sensor sharp utilizando una regla o cinta métrica para verificar las lecturas del sensor.
 7. Establezca una relación lineal entre el ángulo de rotación del servomotor y el ángulo de inclinación de la barra.
 8. Utilizando el software MATLAB y Simulink, realice pruebas para verificar la precisión de la calibración.
 - **Preguntas de Reflexión**
 - ¿Cuál es la importancia de calibrar correctamente los sensores y actuadores en un sistema de control?
 - ¿Cómo afectan los errores de calibración al rendimiento del sistema?

Práctica de Laboratorio 2: Diseño y Ajuste de un Controlador PID

- **Objetivo**
- Diseñar y ajustar un controlador PID para controlar la posición de la esfera en la barra inclinada.
- **Materiales**
 - Planta física (barra inclinada con sensor sharp y servomotor)
 - Esfera de caucho
 - Software MATLAB y Simulink
 - **Procedimiento**
5. Establezca un modelo matemático del sistema utilizando las ecuaciones de movimiento.
 6. Diseñe un controlador PID inicial utilizando el método de Ziegler-Nichols.
 7. Implemente el controlador en Simulink y realice simulaciones para validar su diseño.
 8. Ajuste los parámetros del controlador PID basándose en las respuestas del sistema obtenidas durante las simulaciones y las pruebas experimentales.
 - **Preguntas de Reflexión**
 - ¿Cómo afecta cada parámetro (K_p , K_i , K_d) al rendimiento del controlador?
 - ¿Qué técnicas adicionales podrían utilizarse para sintonizar el controlador PID de forma más efectiva?

Práctica de Laboratorio 3: Validación Experimental

- **Objetivo**
- Validar el modelo del sistema y el controlador PID diseñado mediante experimentación práctica.
- **Materiales**
 - Planta física (barra inclinada con sensor sharp y servomotor)
 - Esfera de caucho

- Software MATLAB y Simulink

- **Procedimiento**

Realice experimentos prácticos utilizando la planta física para validar el modelo del sistema y el controlador PID diseñado.

Compare los resultados experimentales con las predicciones teóricas y los resultados de simulación.

Identifique y analice cualquier discrepancia entre los resultados experimentales y las predicciones/simulaciones.

- **Preguntas de Reflexión**

- ¿Los resultados experimentales validaron efectivamente el modelo y el controlador?
- ¿Qué factores podrían haber contribuido a cualquier discrepancia observada entre los resultados experimentales y las predicciones/simulaciones?

Estas prácticas de laboratorio están diseñadas para brindar una comprensión práctica y profunda de los conceptos involucrados en el control de sistemas dinámicos.

Problemas Comunes y Soluciones

A continuación, se enumeran algunos problemas comunes que pueden surgir durante el uso del Sistema de Control Ball and Beam con LabVIEW, junto con posibles soluciones:

Problema 1: Conexión al Puerto COM*

- **Problema:** La comunicación con la planta Ball and Beam a través del puerto COM no se establece correctamente.
- **Solución:**
 - Asegúrese de que el Arduino esté correctamente conectado y alimentado.
 - Verifique que el puerto COM configurado en LabVIEW coincida con el puerto al que está conectado el Arduino.
 - Revise los cables y conexiones físicas para asegurarse de que estén en buen estado.

Problema 2: Respuesta Inestable del Sistema

- **Problema:** El sistema muestra una respuesta inestable o oscilante.
- **Solución:**
 - Ajuste los valores de los parámetros PID (K_p , K_i y K_d) en la sección de Control de Parámetros PID hasta que obtenga una respuesta estable.
 - Verifique si los valores de ganancia son demasiado altos o bajos y ajústelos en consecuencia.

- Puede ser útil reducir el valor de Alpha en la sección de Configuración de Planta para reducir el ruido en la medición.

Problema 3: No se Alcanza el Set Point

- **Problema:** La esfera no alcanza el valor deseado definido en el Set Point.
- **Solución:**
 - Ajuste los valores de los parámetros PID para mejorar la respuesta del sistema y permitir que alcance el Set Point.
 - Asegúrese de que el Set Point sea alcanzable dado el rango físico de la planta y sus limitaciones.
 - Verifique si el Ángulo Inicial está configurado correctamente y puede afectar la capacidad del sistema para alcanzar el Set Point.

Problema 4: Problemas de Comunicación con el Arduino

- **Problema:** La comunicación con el Arduino se interrumpe o se produce una pérdida de datos.
- **Solución:**
 - Asegúrese de que el Arduino esté correctamente programado y configurado para la comunicación con LabVIEW.
 - Revise si hay interferencias eléctricas o problemas de conexión en el cableado.
 - Verifique la velocidad de comunicación (baud rate) en LabVIEW y asegúrese de que coincida con la configuración del Arduino.

Problema 5: Respuesta Lenta del Sistema

- **Problema:** El sistema responde lentamente a los cambios en el Set Point o perturbaciones.
- **Solución:**
 - Ajuste los valores de K_p , K_i y K_d para mejorar la respuesta del sistema y reducir la sobresaturación o la integral acumulada.
 - Aumente el valor de Alpha en la sección de Configuración de Planta para reducir el ruido en la medición, pero tenga en cuenta que esto puede ralentizar la respuesta.

Si encuentra otros problemas o desafíos al utilizar el Sistema de Control Ball and Beam con LabVIEW, consulte la documentación de LabVIEW, busque asistencia técnica o recurra a sus instructores o supervisores para obtener orientación adicional. La solución a los problemas puede variar según las circunstancias específicas de su configuración y experimentos.